

# **Image Data Augmentation from Small Training Datasets Using Generative Adversarial Networks (GANs)**

A thesis submitted for the degree of  
*Doctor of Philosophy*

**Shih-Kai Hung**

School of Computer Science and Electronic Engineering  
University of Essex

June 2023

# Abstract

The scarcity of labelled data is a serious problem since deep models generally require a large amount of training data to achieve desired performance. Data augmentation is widely adopted to enhance the diversity of original datasets and further improve the performance of deep learning models. Learning-based methods, compared to traditional techniques, are specialized in feature extraction, which enhances the effectiveness of data augmentation.

Generative adversarial networks (GANs), one of the learning-based generative models, have made remarkable advances in data synthesis. However, GANs still face many challenges in generating high-quality augmented images from small datasets because learning-based generative methods are difficult to create reliable outcomes without sufficient training data. This difficulty deteriorates the data augmentation applications using learning-based methods. In this thesis, to tackle the problem of labelled data scarcity and the training difficulty of augmenting image data from small datasets, three novel GAN models suitable for training with a small number of training samples have been proposed based on three different mapping relationships between the input and output images, including one-to-many mapping, one-to-one mapping, and many-to-many mapping. The proposed GANs employ limited training data, such as a small number of images and limited conditional features, and the synthetic images generated by the proposed GANs are expected to generate images of not only high generative quality but also desirable data diversity.

To evaluate the effectiveness of the augmented images generated by the proposed models, inception distances and human perception methods are adopted. Additionally, different image classification tasks were carried out and accuracies from using the original datasets and the augmented datasets were compared. Experimental results illustrate the image classification performance based on convolutional neural networks, *i.e.*, AlexNet, GoogLeNet, ResNet and VGGNet, is comprehensively enhanced, and the scale of improvement is significant when a small number of training samples are involved.

*Keywords: scarcity of labelled data, deep learning, generative adversarial networks (GANs), data augmentation, image classification.*

# Acknowledgement

In the first place, I want to express my gratitude to my family. Due to their emotional countenance and continuous financial support throughout my life, I had a chance to shape myself into the person I am. I hardly repay for all their care, efforts and endless sacrifices.

Secondly, I would like to express my deepest gratitude and appreciation to my supervisor Professor John Gan, for his outstanding supervision and support during my study. He not only led me to the challenging field of computer vision, artificial intelligence and deep learning but also provided a comfortable research environment where I always had sufficient freedom and space to build up my research profile. Moreover, he shaped me as an independent researcher and motivated continuous improvements based on my shortages. When I had research problems or knowledge gaps, he always guided me on how to develop appropriate capabilities for resolving the faced obstacles. Thanks to his enthusiasm and encouragement, I had an opportunity to learn from his rigorous academic attitude as well as visionary thought. Without his instructions on this project, the journey would never be that fruitful.

Last but not least, I would like to thank my colleagues and friends in the School of Computer Science and Electronic Engineering at Essex University. I learned a lot from these outstanding and smart people, who always supported me and shared good times to make my life at Essex memorable and colourful.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgement</b>	<b>iv</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xvii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Problem Statement	1
1.2 Background	1
1.3 Motivation	4
1.4 Thesis Structure	5
<b>2. Literature Review on Deep Learning and Image Data Augmentation</b>	<b>7</b>
2.1 Convolutional Neural Networks (CNNs)	7
2.1.1 Deep Learning	7
2.1.1.1 Supervised Learning	8
2.1.1.2 Unsupervised Learning	9
2.1.1.3 Reinforcement Learning	9
2.1.2 Basic Components of CNNs	10
2.1.2.1 Convolutional Layer	10
2.1.2.2 Regularisation Layer	11
2.1.2.2.1 Pooling	12
2.1.2.2.2 Dropout	13
2.1.2.2.3 Batch Normalisation	14
2.1.2.3 Fully-connected Layer	15
2.1.2.4 Activation Functions	16
2.1.2.4.1 Sigmoid	17
2.1.2.4.2 Hyperbolic Tangent	17
2.1.2.4.3 Rectified Linear Unit	17
2.1.2.4.4 Leaky Rectified Linear Unit (Leaky ReLU)	18
2.1.2.4.5 Softmax	19
2.1.3 CNN Architecture	19
2.1.3.1 AlexNet	19

2.1.3.2 VGGNet.....	20
2.1.3.3 GoogLeNet .....	21
2.1.3.4 Residual Network .....	22
2.2 Techniques for Learning from Small Training Data.....	23
2.2.1 Scarcity of Labelled Data and Class Imbalance .....	24
2.2.1.1 Scarcity of Labelled Data .....	24
2.2.1.2 Class Imbalance .....	25
2.2.2 Techniques for Learning from Small Training Data .....	26
2.2.2.1 Transfer Learning .....	26
2.2.2.2 Semi-supervised Learning .....	27
2.2.2.3 One-shot Learning and Few-shot Learning .....	27
2.2.2.4 Data Synthesis .....	28
2.2.2.5 Data Augmentation .....	29
2.3 Image Data Augmentation .....	30
2.3.1 Traditional Augmentation Techniques.....	31
2.3.1.1 Geometric Transformations .....	31
2.3.1.1.1 Flipping .....	32
2.3.1.1.2 Rotation .....	32
2.3.1.1.3 Translation.....	33
2.3.1.1.4 Cropping.....	34
2.3.1.2 Photometric Transformations .....	34
2.3.1.2.1 Noise Adding .....	34
2.3.1.2.2 Colour Space Shifting .....	35
2.3.1.2.3 Kernel Filter .....	36
2.3.1.2.4 Random Erasing .....	37
2.3.2 Image Data Augmentations Based on Deep Learning Methods .....	38
2.3.2.1 Meta-metric Learning .....	38
2.3.2.2 Feature Space Augmentation.....	38
2.3.2.3 Augmentation Using Generative Adversarial Networks .....	39
2.4 Conclusion .....	39
<b>3. Literature Review on Generative Adversarial Networks .....</b>	<b>41</b>
3.1 Introduction.....	41

3.2 A Fundamental Framework of GANs .....	42
3.2.1 Typical GANs .....	42
3.2.2 Convolution-based GAN .....	44
3.3 Structure Variants of GANs for Image Synthesis .....	45
3.3.1 Condition-based GAN .....	45
3.3.2 Auxiliary Classifier GAN .....	47
3.3.3 Autoencoder-based GAN .....	48
3.3.4 Attention-based GAN .....	51
3.4 Loss Function Variants of GANs .....	52
3.4.1 Wasserstein GAN .....	52
3.4.2 Wasserstein GAN with Gradient Penalty .....	53
3.4.3 Least Square GAN .....	54
3.5 Challenges in Training GANs .....	55
3.5.1 Mode Collapse .....	55
3.5.2 Gradient Vanishing .....	57
3.5.3 Non-convergence .....	58
3.5.4 Hyperparameter Optimisation .....	59
3.6 Evaluation Metrics .....	60
3.6.1 Likelihood Estimation .....	60
3.6.2 Inception Scores .....	61
3.6.3 Fréchet Inception Distance .....	62
3.6.4 Kernel Inception Distance .....	63
3.6.5 Classification Accuracy as an Evaluation Metric .....	63
3.7 Applications of GANs in Image Synthesis .....	64
3.7.1 Image Super-resolution .....	64
3.7.2 Image Repairing .....	65
3.7.3 Face Synthesis .....	65
3.7.4 Image Translation .....	66
3.7.4.1 Paired Image-to-image Translation .....	66
3.7.4.2 Unpaired Image-to-image Translation .....	68
3.7.5 Video Synthesis .....	70
3.8 Conclusion .....	71

<b>4. Small Training Data Augmentation Using GANs Based on One-to-many Image Mapping for Enhancing the Performance of Image Classification.....</b>	<b>72</b>
4.1 Introduction.....	72
4.2 Methods.....	75
4.2.1 Network Framework .....	76
4.2.2 Perturbation Mechanism.....	77
4.2.3 Model Building .....	77
4.2.3.1 Generator .....	79
4.2.3.2 Discriminator .....	80
4.2.4 Loss Functions.....	82
4.2.5 Training Process .....	82
4.3 Experiments with the Proposed GAN Framework .....	85
4.3.1 Data Preparation .....	85
4.3.2 Hardware & Software.....	85
4.3.3 Hyperparameters Setting .....	86
4.4 Evaluation of Training Data Augmentation for Image Classification .....	91
4.4.1 Performance Comparison of Image Classification with Augmented Image Data .....	92
4.4.1.1 Student's T-test.....	94
4.4.1.2 Evaluation by Confusion Matrix .....	94
4.4.1.3 Evaluation by t-SEN Plot .....	99
4.4.2 Comparison of Image Classification Accuracies with Traditional Image Augmentation Methods .....	100
4.5 Conclusion .....	102
<b>5. Facial Image Synthesis from Small Training Data and Sparse Edge Features Using a GAN Framework based on One-to-one Image Mapping .....</b>	<b>103</b>
5.1 Introduction.....	103
5.2 Methods.....	106
5.2.1 The Proposed Condition-based GAN Framework .....	109
5.2.2 Image Pre-processing and Refining .....	111
5.2.3 Edge Extraction .....	111
5.2.4 Adoption of Interim Domain.....	112
5.2.5 Model Training and Loss Functions.....	115
5.3 Experiments with the Proposed GAN Framework .....	116



5.3.1 Data Preparation .....	116
5.3.2 Implementation Details .....	116
5.4 Results and Performance Evaluation .....	120
5.4.1 Diversity in Facial Image Augmentation Using the Proposed Condition-based GAN .....	120
5.4.2 Qualitative Comparison.....	125
5.4.3 Quantitative Comparison.....	126
5.4.3.1 Evaluation of the Influence of Conditional Edges.....	127
5.4.3.2 Evaluation of the Usefulness of Interim Domain .....	129
5.4.3.3 Evaluation of the Impact of the Number of Training Samples.....	131
5.4.3.4 Evaluation by Human Perception .....	132
5.4.3.5 Evaluation by Balanced Image Classification .....	133
5.4.3.6 Evaluation by Imbalanced Image Classification .....	135
5.5 Conclusion .....	140
<b>6. Augmenting Small Facial Expression Training Dataset Using a Novel GAN Model Based on Many-to-many Image Mapping .....</b>	<b>141</b>
6.1 Introduction.....	141
6.2 Methods.....	143
6.2.1 Subnetworks .....	145
6.2.1.1 Generators.....	145
6.2.1.2 Discriminators.....	146
6.2.1.3 Feature Extractor .....	148
6.2.2 Feature Map Mechanism.....	149
6.2.3 Model Learning .....	151
6.2.3.1 Adversarial Loss .....	152
6.2.3.2 Cycle Loss .....	153
6.2.3.3 Perceptual Loss .....	153
6.2.3.4 Feature Loss.....	154
6.2.3.5 Overall Loss.....	155
6.3 Experiments with the Proposed GAN Framework .....	155
6.3.1 Datasets .....	155
6.3.1.1 Extended Cohn-Kanade Dataset .....	155
6.3.1.2 Karolinska Directed Emotional Faces Dataset .....	156

6.3.1.3 Taiwanese Facial Expression Image Dataset.....	156
6.3.2 Experimental Setup .....	157
6.3.3 Ablation Studies .....	157
6.3.3.1 Weighting Values in the Overall Loss Function.....	158
6.3.3.2 Adversarial Loss .....	160
6.3.3.3 Feature Map Mechanism .....	162
6.4 Performance Evaluation.....	163
6.4.1 Qualitative Comparison.....	163
6.4.1.1 Visual Analysis with Different Number of Training Images in the Target Domain .....	163
6.4.1.2 Visual Analysis with Different Expressional Classes .....	166
6.4.1.3 Comparison with the State-of-the-art .....	168
6.4.2 Quantitative Evaluation.....	171
6.4.2.1 Evaluation by FID and KID.....	171
6.4.2.2 Evaluation by Performance Enhancement in Image Classification...	172
6.4.2.3 Student’s T-test.....	175
6.5 Conclusion .....	176
<b>7. Conclusions and Future Work .....</b>	<b>177</b>
7.1 Summary of Contributions.....	177
7.2 Limitations and Future Work.....	178
<b>Bibliography</b>	<b>181</b>
<b>Appendix A</b>	<b>201</b>

# List of Figures

Figure 1.1	Illustration of image transforming from source domains (red dots) to target domains (blue dots) with different image-to-image mapping relationships. ....	3
Figure 2.1	An example of a convolutional layer. ....	11
Figure 2.2	An example of max pooling and average pooling. ....	13
Figure 2.3	An illustration of the dropout in a fully-connected layer. ....	14
Figure 2.4	An illustration of the fully-connected layer. ....	16
Figure 2.5	Activation function: (a) Sigmoid function, (b) Hyperbolic tangent function, (c) Rectified linear unit function, (d) Leaky rectified linear unit function. ....	18
Figure 2.6	Basic architecture of AlexNet. ....	20
Figure 2.7	Basic architecture of the inception block. ....	21
Figure 2.8	Basic architecture of the residual block. ....	23
Figure 2.9	Flipping technique, where (a) is the original image, (b) is vertical flipping, (c) is horizontal flipping, and (d) is vertical and horizontal flipping. ....	32
Figure 2.10	Samples of rotated images. ....	33
Figure 2.11	Samples of translation images. ....	33
Figure 2.12	Samples of cropping images. ....	34
Figure 2.13	Sample images of noise added by different percentages. (a) to (d) is the images with salt & pepper noise, and (e) to (h) add noise with Gaussian distribution. ....	35
Figure 2.14	Sample images of colour space shifting. ....	35
Figure 2.15	Sample images using different kernel filters. ....	36
Figure 2.16	Sample images of random erasing. ....	37
Figure 3.1	Conceptual idea of the GAN structure. ....	43
Figure 3.2	The structure of a deep convolutional generative adversarial network. ....	44
Figure 3.3	The basic structure of a condition-based GAN. ....	46
Figure 3.4	The basic structure of an auxiliary classifier GAN. ....	47
Figure 3.5	The basic structure of an adversarial autoencoder. ....	48
Figure 3.6	The basic structure of BiGAN. ....	49
Figure 3.7	The basic structure of an adversarial generator-encoder network (AGE). ....	50

Figure 3.8	The images generated by GANs using the CelebA dataset. The left column shows more diverse generative results while the right column presents a mode collapse when only a few modes of facial data are generated. ....	56
Figure 3.9	The basic structure of the pix2pix model.....	67
Figure 3.10	The basic structure of the CycleGAN.....	69
Figure 4.1	Comparison of perfect training samples in human vision (left column) and good training samples in deep learning models (right column). ....	73
Figure 4.2	Overview of the proposed GAN framework for data augmentation from a single original image. ....	76
Figure 4.3	The model components of discriminator (left column) and generator (right column), where the repeated components indicate the upsampling or downsampling number of convolutional layers in the proposed GAN model. ....	78
Figure 4.4	The original image and generated images with a bad setup of generating images from a single original image using the proposed GAN framework. ....	87
Figure 4.5	Original images (left column) for training and the generated images (right column) using the proposed model with different transformation matrices $M'$ on the MNIST dataset. ....	89
Figure 4.6	Original images and the generated images with small-scale rotations implemented by matrix $M'$ and larger-scale rotations implemented by matrix $M$ . ....	90
Figure 4.7	Comparison of validation accuracy of CNNs on the MNIST dataset. ....	93
Figure 4.8	Comparison of validation accuracy of CNNs on the RPS dataset. ....	93
Figure 4.9	Confusion matrix for testing data on the RPS dataset. The AlexNet is trained without using augmented data. ....	95
Figure 4.10	Confusion Matrix for testing data on the RPS dataset. The AlexNet is trained with the original and augmented data. ....	95
Figure 4.11	Confusion Matrix for testing data on the MNIST dataset. The AlexNet is trained without using augmented data. ....	96
Figure 4.12	Confusion Matrix for testing data on the MNIST dataset. The AlexNet is trained with the original and augmented data. ....	96
Figure 4.13	A comparison of the sensitivity, specificity, accuracy and precision with augmented data and original small data to train the AlexNet. ....	98
Figure 4.14	Two-dimension t-SEN plot of the RPS dataset. (a) The validation data distribution with a validation accuracy = 78.08% when the AlexNet is trained by the original small training data. (b) The validation data	

	distribution with a validation accuracy = 87.8% when the AlexNet is trained by the original small training data and augmented data. ....	99
Figure 4.15	Two-dimension t-SEN plot of the MNIST dataset. (a) The validation data distribution with a validation accuracy = 71.59% when the AlexNet is trained by the original small training data. (b) The validation data distribution with a validation accuracy = 86.37% when the AlexNet is trained by the original small training data and augmented data. ....	100
Figure 5.1	Examples of training results with a different number of training images and different conditional edge inputs by using the same parameter setting for the one-to-one translation method. ....	104
Figure 5.2	Examples of inference results with a different number of training images and different conditional edge inputs by using the same parameter setting for the one-to-one translation method. ....	105
Figure 5.3	The proposed translation method by defining a refined domain based on a small training dataset. The GANs and image pre-processing are adopted to enrich the mapping relationships from the source domain to target domain. ....	107
Figure 5.4	Overview of the proposed model for translating edges to photorealistic images using two U-nets. ....	110
Figure 5.5	Corresponding mapping relationships among the conditional inputs, refined image and ground truth. ....	112
Figure 5.6	Inference results in translating sparse edges to labelled segmentation masks with 50 random training images. (a) The outputs can roughly resume the missing facial components from incomplete layouts when given abstract inputs. (b) The red boxes indicate the corresponding indefinite contours in the original inputs and generative masks. ....	113
Figure 5.7	Inference results in translating sparse edges to binary regional images with 50 random training images. (a) The outputs integrate discontinued contours when given sparse inputs. (b) The outputs get rid of ‘bogus’ edges when given very dense inputs. ....	114
Figure 5.8	Comparison of different edge detectors: (a) results of Canny. (b) results of Sobel. (c) results of Laplace. (d) results of Gradient. ....	117
Figure 5.9	Inference results for refined images and final outputs. The red boxes represent blending areas in the refined region, which can be reflected by the brightness in the generated image outputs. ....	118
Figure 5.10	Synthesis results of exchanging conditional facial edges to generate diverse styles of facial images. ....	119
Figure 5.11	Inference results in the source, interim, and target domains respectively. The various density levels in the conditional inputs are not in the training phase except the one in the red box generated by the Canny edge	

	detector with the threshold value of 0.4. The results are from GANs trained using 50 images only. ....	121
Figure 5.12	Examples of facial image augmentation results using 50 training images, with parts of input edges modified for introducing diversity to augment each training image with desirable facial features. ....	122
Figure 5.13	Examples of facial image augmentation results using 50 training images, with face components and hairstyles in different training images swapped in the edges as conditional inputs to generate diverse facial images. ....	123
Figure 5.14	Inference results shown by images in the source, refined, and target domains respectively. The conditional inputs are hand-drawn sketches showing different facial expressions. The proposed GAN was trained using 50 training images. ....	124
Figure 5.15	Inference results generated with sparse edge inputs (the first row), in comparison with those obtained from the state-of-the-art condition-based GANs. The images were generated respectively by the three GANs for comparison, trained using the same small dataset of 50 training images. ....	126
Figure 5.16	Samples of different threshold values using the Canny edge extractor. ....	127
Figure 5.17	FID and KID scores with different levels of input edge density using the proposed model. One input type (high threshold = 0.4) and three input types (high threshold = 0.2, 0.4, 0.6) in the source domain were used during training with a small training dataset of 50 images. The FID and KID scores were evaluated based on the same 1,000 inference images for each edge density level from 0.01 to 0.9. ....	128
Figure 5.18	FID and KID scores of double U-nets with refined domain and single U-net with one input type (high threshold = 0.4) in the source domain, where a small training dataset of 50 images was used during training. The FID and KID scores were calculated based on the same 1,000 inference images at different edge density levels. ....	130
Figure 5.19	FID and KID scores of double U-nets with refined domain and single U-net with three input types (high threshold = 0.2, 0.4, 0.6) in the source domain, where a small training dataset of 50 images was used during training. The FID and KID scores were calculated based on the same 1,000 inference images at different edge density levels. ....	130
Figure 5.20	Changes in FID scores (first row) and KID scores (second row) with a different number of training images. Comparison among three edge-to-image translation methods with sparse and dense edge inputs respectively: pix2pix, pix2pixHD and ours. ....	131

Figure 5.21	Samples of inference results (second and third row) generated from the classes of male and female separately (bottom row), where the results are generated with 15 training images by inputting the same sparse edges (top row). .....	134
Figure 5.22	Samples of inference results (bottom row) from the proposed GAN, trained by 20 images in the abnormal class, where the input edges (second row) are extracted from the normal images (top row).....	137
Figure 5.23	Comparison of confusion matrices of the GoogLeNet trained with and without using augmented images: The left column shows the results without using augmented images and the right column shows the results using augmented images. ....	138
Figure 5.24	Comparison of the accuracy, precision, recall and F1 of GoogLeNet trained with and without using augmented images.....	139
Figure 6.1	An overview of the proposed GAN model, which contains five subnetworks, including two generators with the encoder and decoder structure, a feature extractor and two discriminators. The proposed feature map mechanism is involved in both the encoder and feature extractor. ....	144
Figure 6.2	The residual block used in the proposed GAN model. ....	146
Figure 6.3	The feature map mechanism in the proposed model. ....	151
Figure 6.4	Effect of different weight values in the overall loss function on the generated surprise images with the KDEF dataset. ....	159
Figure 6.5	Effect of different adversarial loss functions on the generated surprise images with the KDEF dataset.....	160
Figure 6.6	Effect of different adversarial loss functions on the generated surprise images with the CK+ dataset. ....	161
Figure 6.7	Comparison of facial expression images generated by the proposed GAN model with and without the feature map mechanism. The real images are from the KDEF dataset without being involved in the training, during which fear images were used as target samples and neutral images were used as inputs. ....	162
Figure 6.8	Images generated by the proposed GAN model for augmenting the surprise class of the KDEF dataset. All the synthetic results were obtained by using 40 neutral images in the source domain and up to 40 surprise images in the target domain. The real images shown in the figure were not involved in the training.....	164
Figure 6.9	Images generated by the proposed GAN model for augmenting the surprise class of the TFEID dataset. All the synthetic results were obtained using 40 neutral images in the source domain and up to 40	

	surprise images in the target domain. The real images shown in the figure were not involved in the training.....	165
Figure 6.10	Images generated by the proposed GAN model by augmenting the TFEID dataset, where 40 neutral images and 20 images of each emotional class were used to train the proposed GAN model for transferring neutral images to images with various facial expressions.	166
Figure 6.11	Images generated by the proposed GAN by augmenting the CK+ dataset, where 50 neutral images and 20 images of each emotional class were used to train the proposed GAN model for transferring neutral images to images with various facial expressions.....	167
Figure 6.12	Images generated by the proposed GAN by augmenting the KDEF dataset, where 70 neutral images and 20 images of each emotional class were used to train the proposed GAN model for transferring neutral images to images with various facial expressions. ....	167
Figure 6.13	Comparisons of facial expression transfer from neutral face images (first column) to surprise expression by CycleGAN, MUNIT, UNIT, AttentionGAN.v2 and the proposed GAN model respectively: (a) CK+ dataset. (b) KDEF dataset. (c) TFEID dataset. ....	170



## List of Tables

Table 4.1	The generator network and related parameters. ....	80
Table 4.2	The Discriminator network and related parameters. ....	81
Table 4.3	Hardware environment. ....	86
Table 4.4	Software version. ....	86
Table 4.5	The setting of parameter values. ....	87
Table 4.6	Augmented images from the original image using different transformation matrices $M'$ . ....	88
Table 4.7	Significance test results (p-values) for comparing validation accuracy of CNNs trained with vs without augmented training data. ....	94
Table 4.8	Validation accuracy of 20 sample images per class for our method and traditional data augmentation on the MNIST dataset. ....	101
Table 4.9	Validation accuracy of 20 sample images per class for our method and traditional data augmentation on the RPS dataset. ....	101
Table 5.1	Results from user preference study. The percentage indicates the users who favour the results of our proposed method over the competing method. ....	133
Table 5.2	Validation accuracies of CNNs trained with different numbers of original images and augmented images per class. ....	135
Table 5.3	Validation accuracies of CNNs trained with different numbers of normal images in the imbalanced training dataset. ....	136
Table 5.4	Comparison of validation accuracies of CNNs trained with and without using augmented images. ....	138
Table 6.1	The generator network and related parameters. ....	147
Table 6.2	The discriminator network and related parameters. ....	148
Table 6.3	The feature extractor network and related parameters. ....	149
Table 6.4	Description of facial expression datasets. ....	156
Table 6.5	Number of images used as the small datasets. ....	157
Table 6.6	Reality scores estimated by FID and KID metrics. Lower FID and KID values indicate higher visual similarity between real and generated images. ....	172
Table 6.7	Comparison of validation accuracies of four CNNs trained with 20 training samples from each class and augmented facial expression images generated by the proposed GAN and CycleGAN on the CK+ dataset. ....	173

Table 6.8	Comparison of validation accuracies of four CNNs trained with 20 training samples from each class and augmented facial expression images generated by the proposed GAN on the CK+, KDEF and TFEID facial expression datasets respectively.....	174
Table 6.9	The p-values of the student's t-test for comparing the performances of four CNNs trained with and without using augmented facial expression images respectively. ....	175

# Chapter 1

## Introduction

### 1.1 Problem Statement

With the advances in artificial intelligence, machine learning has become a very popular research field in recent years [1]. Machine learning is a subfield of artificial intelligence that enables computers to learn from data to make predictions or decisions. Since data quality and quantity are two foundational elements in training deep learning models, acquiring high-quality training data is a prerequisite to achieving the expected performance, and training with a large amount of diverse data is one of the necessary factors for the desired results [2]. In some practical applications, collecting a large amount of training data is impractical because data collection generally takes plenty of time and needs to be labelled or post-processed by experts. Furthermore, public data might have confidentiality and privacy concerns, and most public data contain label limitations or quality restrictions, which are hard to be freely used for various applications.

Many remarkable deep learning algorithms and models have been developed in recent years and demonstrated their powerful capacities to achieve great performance in some specific applications. However, with the increased requirements of training data quantity and quality, data availability is a primary factor affecting the performance of deep learning, and data scarcity has become a common but serious problem in the development of machine learning and deep learning methods [3]. A small number of labelled data samples hardly provide enough information for a deep model to learn well, and insufficient training data generally leads to negative impacts on the final performance of the trained deep models [4]. Consequently, the quality and quantity of training data are fundamental to ensure a deep learning model able to achieve the desired capabilities.

### 1.2 Background

Data augmentation is a technique used to increase data types with methods of data editions or data synthesis for enlarging data diversity and quantity. Data augmentation is a common resolution to mitigate the unfavourable impacts of labelled data scarcity

and class imbalance in machine learning [5]. Due to the demanding requirement of a large amount of high-quality training data in various machine learning applications, developing synthetic techniques for training data augmentation has become a popular research area [6]. In many studies, data augmentation has been proven to play a critical role in efficiently promoting the performance of deep learning applications [7], [8]. Additionally, data synthesis is a beneficial means to augment rare datasets, when sufficient or meaningful data are difficult to be collected in some research schemes (*e.g.*, data on rare diseases, space images, remote sensing data, *etc.*).

With the developments of optimisation techniques and computing hardware, modern computers make deep learning applications achievable, where numerous parameters in deep learning models need to be fine-tuned [9]. However, it is fairly hard to attain convincing results without collecting sufficient training data, although deep learning models, such as convolutional neural networks (CNNs), have achieved extraordinary successes in a wide range of computer vision applications (*i.e.*, image classification, object recognition, *etc.*). Even with the latest techniques, it is still a challenging task to automatically synthesise high-quality data from a small number of training samples with limited feature information [10]. Particularly, image synthesis using learning-based methods needs a relatively large amount of training data to achieve photorealistic results and mitigate synthetic problems such as distortion and overfitting [11]. In this thesis, learning-based synthetic methods are developed to explore novel generative models, which aim to reduce the negative effects, *i.e.*, overfitting, gradient vanishing, non-converge, mode collapse, hyperparameter optimisation *etc.*, caused by training with a small number of sample images. These negative effects normally result in generative uncertainty and easily bring about blurring, distortion, and less diversity in the output results.

In terms of the image mapping relationships between input and output data, the existing generative models used for image synthesis can be approximately categorised into four groups, one-to-one, one-to-many, many-to-one and many-to-many [12], [13]. Figure 1.1 shows the difference among these four mapping relationships by transforming images from source domains to target domains, where the red dots represent the input data in the source domain and blue dots the output data in the target main, the solid lines correspondingly indicate the mapping relationships between two domains and the dashed lines the internal relationships in one defined domain.

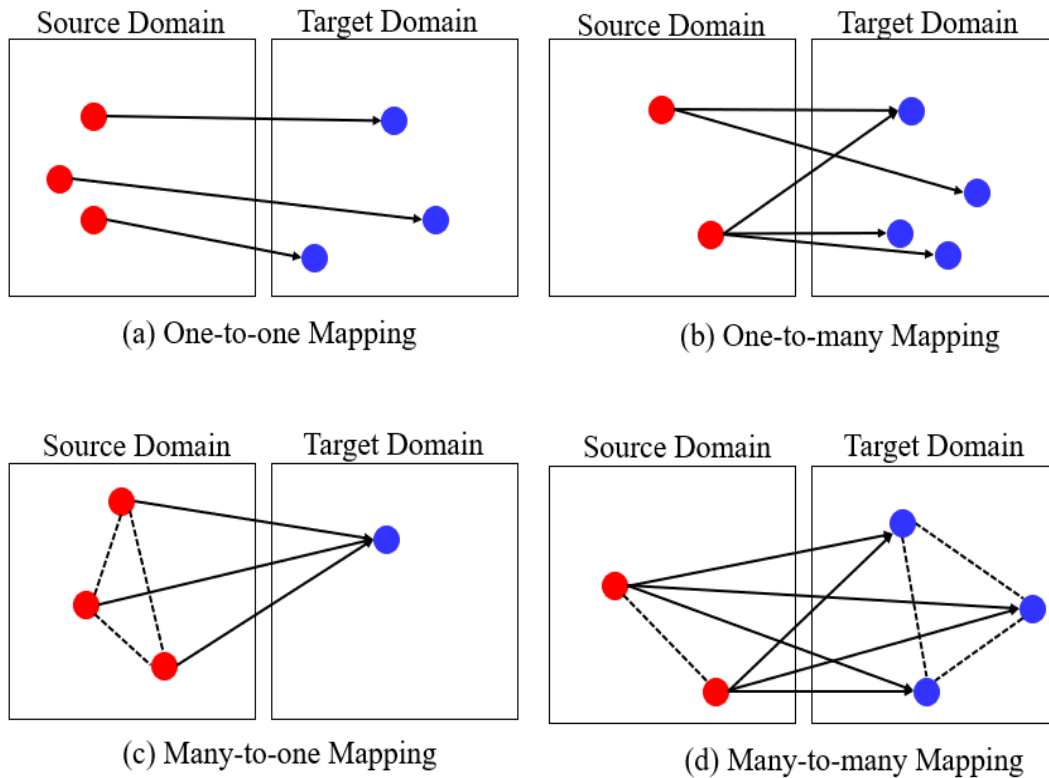


Figure 1.1: Illustration of image transforming from source domains (red dots) to target domains (blue dots) with different image-to-image mapping relationships.

Different mapping relationships will affect the design of network structures, components, algorithms, loss functions and so on. A brief of each mapping relationship is described as follows: 1) Firstly, one-to-one mapping methods transform images from one style to another, such as the translation from image to sketch, from low resolution to high resolution, or from optical diagram to infrared spectrum. The one-to-one mapping generally needs paired images for models to learn the mapping relationships by specific objective functions. 2) Secondly, many-to-one mapping methods translate images from many related inputs to one united outcome. For instance, a frontal view of human faces can be transferred by multi-views of different facial positions and profiles. However, many-to-one mapping methods commonly need many input data to generate a few specific outcomes, which may not be suitable for regular data augmentation cases to create a large amount of additional and diverse data from small datasets. 3) Thirdly, one-to-many mapping methods synthesise many output results from one single image. Many data augmentation methods with generative models are based on one-to-many mapping relationships, which create many diverse results from a few training samples. 4) Finally, many-to-many mapping methods control models to learn the relationships among many related images between different domains, and the many-to-many mapping generally does not need paired training images, compared to one-to-one

mapping. Hence, many-to-many mapping applications can be regarded as translation tasks between two data domains. It is notable that although many-to-many mapping methods eliminate the requirements on paired images, they still require the interpretation or labelled information between two domains, such as the image transformation cases of male to female, sketch to photo, real face to cartoon face, day scenery to night scenery and so on. Consequently, all the above-mentioned mapping relationships still have to rely on labelled data between two different domains to generate new images.

Generative adversarial networks (GANs) are one of the learning-based generative models associated with deep neural networks to synthesise data. GANs are powerful generative models and have been applied in many image synthesis applications, such as facial reconstruction, image generation, style transformation, image repairing, data augmentation, video synthesis and so on [14]. GAN structures have been proven by many studies to be able to generate remarkable image samples, and the advancement of GANs accelerates the applications of image synthesis [15]. However, with limited feature information, it is challenging to generate high-quality synthetic data from small datasets by using learning-based generative models, and training GANs with insufficient data easily results in many negative impacts on the generated data, such as poor diversity, low reality, large distortion and so on [16]. To solve the problems in training GANs with small training data, novel GAN structures suitable for training using small datasets are developed in this thesis to not only effectively mitigate the synthetic problems caused by training with a small number of training samples but also comprehensively promote the performance in various image classification applications.

## 1.3 Motivation

The capabilities of GANs to synthesise image data have been proven by many studies. However, typical GAN models for image synthesis are composed of deep convolutional layers and are difficult to synthesise high-quality images from limited training information, such as very sparse conditional features and a small number of training images [17]. To generate photorealistic and unblurring images, the two neural networks of GANs, *i.e.*, the generator and discriminator, need a large amount of training data to fine-tune their free parameters for generating good forgeries. In short, GANs are one of the most widespread generative models used for image data synthesis, but the drawback of GAN structures is obvious, that is, they need a large number of training images to reach high-quality synthetic results in image synthesis.

In this thesis, the research motivation is to propose novel GAN frameworks or

improve the existing GAN structures that hardly generate promising results based on small training datasets and limited training features. For mitigating the synthetic disadvantages caused by training with limited samples, this study aims to develop novel GAN models, which can not only enlarge the data quantity but also increase the data diversity based on different image mapping relationships. Compared to traditional learning-based methods difficult to obtain good results from small datasets, the synthetic data using the proposed GANs are designed to synthesise representative data to promote deep learning performance, such as enhancing accuracy in image classification by CNNs. Hence, the following two issues are to be investigated in this thesis: 1) How to resolve the non-convergence and overfitting problems in training GANs with small training datasets and reduce unexpected distortions. 2) How to generate photorealistic results with learning-based generative models using limited information in training data. Developing enhanced GAN models to deal with the problems caused by a small number of training samples is critical for extending deep learning applications. Consequently, the main aim of this thesis is to propose new GAN models to generate photorealistic images of desired diversity from a small number of training samples.

## 1.4 Thesis Structure

The structure and content for each chapter of this thesis are as follows.

**Chapter 2 Literature Review – Part 1:** In this chapter, three major topics related to this thesis, including convolutional neural networks, data insufficiency and methods of image data augmentation, are comprehensively reviewed.

**Chapter 3 Literature Review – Part 2:** GANs for image synthesis and their basic backgrounds are introduced in this chapter with six primary GAN schemes, including basic theory, structure variants, loss function variants, training challenges, evaluation metrics, and applications.

**Chapter 4 Small Training Data Augmentation Using GANs Based on One-to-many Image Mapping for Enhancing the Performance of Image Classification:** A novel method for data augmentation is proposed to solve the problem of machine learning with small training datasets. The proposed method can synthesise similar images with rich diversity from only a single original training sample to increase the number of training data by using GANs. It is expected that the synthesised images possess class-informative features, which may be in the validation or testing data but not in the training data because the training dataset is small, and thus they can be effective as augmented training data to improve the classification accuracy of CNNs.

**Chapter 5 Facial Image Synthesis from Small Training Data and Sparse Edge Features Using a GAN Framework based on One-to-one Image Mapping:**

A conditional GAN framework is proposed for facial image augmentation using a very small training dataset and incomplete or modified edge features as conditional input for diversity. The proposed method defines a new domain or space for refining interim images to prevent overfitting caused by using a very small training dataset and enhance the tolerance of distortions caused by incomplete edge features, which effectively improves the quality of facial image augmentation with diversity. Experimental results have shown that the proposed method can generate high-quality images of good diversity when the GANs are trained using very sparse edges and a small number of training samples. Compared to the state-of-the-art edge-to-image translation methods that directly convert sparse edges to images, when small training datasets were used, the proposed conditional GAN framework can generate facial images with desirable diversity and acceptable distortions for dataset augmentation and significantly outperform the existing methods in terms of the quality and quantity of synthesised images, evaluated by Fréchet inception distance (FID), kernel inception distance (KID) scores, student's t-test, human perception and image classification.

**Chapter 6 Augmenting Small Facial Expression Training Dataset Using a Novel GAN Model Based on Many-to-many Image Mapping:**

A new GAN model is proposed to transfer neutral face images to images with diverse facial expressions to deal with the problem of expressional data scarcity in deep learning for facial expression recognition (FER) based on a small set of training samples. To mitigate distortions and overfitting that often happen in training GANs with small training datasets, a novel GAN structure is proposed, which consists of a generator with two encoders and two decoders, two discriminators, and a feature extractor. Specifically, a feature map mechanism is proposed to discover regional feature differences between images in the source domain and target domain, which makes the proposed GAN structure able to not only generate desirable facial expression images but also maintain the original characters in the input neutral face images. Experimental results show that, by using the proposed GAN to augment a training dataset of images with up to 7 facial expressions, the FER accuracy of several deep neural networks tested in the experiments can be significantly improved by over 10%.

**Chapter 7 Conclusions and Future Work:** Conclusions are drawn and discussed in this chapter to summarise the findings of the research work. In addition, the limitations and future work for each proposed GAN model are presented in this chapter.



## Chapter 2

# Literature Review on Deep Learning and Image Data Augmentation

In this chapter, the state-of-the-art theory and approaches related to deep learning and data augmentation are reviewed. Firstly, convolutional neural networks (CNNs) are introduced, including the concept of deep learning, basic components and structures of CNNs. Secondly, since the advantages of deep learning are based on analysing a large amount of training data, labelled data scarcity and class imbalance become two serious concerns in deep learning applications. Therefore, techniques for learning from small training data are reviewed. Finally, image data augmentation methods with both traditional techniques and approaches based on deep learning methods are separately presented.

## 2.1 Convolutional Neural Networks (CNNs)

### 2.1.1 Deep Learning

Deep learning, also called representation learning or feature learning, is defined as a manner to extract features in a hierarchical structure [18]. In recent years, with the growth of data availability and advance in computer technologies, deep learning has become one of the most prevalent research areas and found widespread applications in information retrieval, image classification, decision recommendation, social network analysis, data mining and so on [19]. Methods in deep learning utilise technologies to develop multi-layer learning models depending on traditional frameworks of neural networks, and many latest deep learning techniques have demonstrated remarkable results in various application areas, *i.e.*, natural language processing, brain-computer interface, autopilot system and many other well-known applications [20]. Deep learning imitates the learning process of human neurons to create interconnected structures developed from cognition and information theory. Although deep learning is extensively applied to various types of real applications, it is impossible by far to develop one single model or network universally suitable for all requirements in reality [21].

Deep learning approaches are not new technologies, but they could not be

implemented due to the limitations of computing facilities in the past [22]. Fortunately, with the rapid development of computing capacities as well as advanced hardware in recent years, it has become possible to process large collections of data and parameters with deep learning algorithms. This improvement brings about an enormous evolution to enlarge realistic applications of deep learning.

To evaluate the deep learning performance, it adopts techniques to iteratively improve the training process and extract the representations from trained models [23]. In a learning process, a volume of data should be split into three parts, the training set, validation set and test set. A deep learning algorithm will learn representations from the given training set, which could be approximative functions to find feature distributions or decisions based on a training set. On the other side, the validation set will be used during training as a method to validate the effectiveness of the training process, and the validation results are evidence to tune learning parameters for improving the final performance. Finally, a test set, which is never involved in the training process, is provided to determine the final accuracy of the trained model using accuracy scores or other effectiveness metrics. In terms of inference, the test set can be regarded as a process of inputting data into a trained model and obtaining inferred outputs. The efficiency of deep learning approaches strongly relies on the quality and quantity of training data [24]. Training with limited data or representations easily leads to unexpected results, which is the main reason why data scientists and engineers are always concerned about the quality and quantity of training data for each learning case.

Deep learning is machine learning using deep neural networks. By definition, primary machine learning mechanisms include supervised learning, unsupervised learning, and reinforcement learning [25].

### **2.1.1.1 Supervised Learning**

Supervised learning needs to rely on labelled training data, which makes desired outputs available for supervising the learning process. Generally, supervised learning is a prediction mechanism, and two primary learning tasks, classification and regression, are usually conducted in supervised learning [26].

For classification, the output results of the learning tasks should be a specific set of classes, which can be a form of binary classification of two classes (*e.g.*, 0 or 1, right or wrong, true or false, *etc.*) or multiple classes. Multi-class classification can be treated as a binary classification among every class, which directly compares the classification results of every class using the binary method [27]. In contrast to classification, the output of regression learning is continuous values of a possibility rather than binary values. Regression learning is a statistical technique used to discover a relationship

between variables for predicting the outcome of unseen input data. By estimating how one variable affects the others, regression learning has been broadly applied in various areas to fill or forecast gaps of missing data, such as risk management, price prediction, and so on. Regression learning is a common method for supervised learning, which requires labelled data.

### **2.1.1.2 Unsupervised Learning**

Unsupervised learning, contrasted with supervised learning, is related to learning on unlabelled data [28]. In other words, the algorithms of unsupervised learning do not need human interventions or information about desired output to discover useful information for data learning purposes [29]. For instance, clustering algorithms may cluster data into groups without appropriate visual representations, which would be difficult to identify whether the clustering is appropriate or not. Therefore, the representation accuracies may still need to be further tested to determine an appropriate implementation.

Unsupervised learning is commonly employed for clustering, density estimation and dimensionality reduction [30], [31]: Clustering is based on statistical algorithms and occurs toward an alternative selection of centroids and clusters [32]. Density estimation is a statistical approximation to discover a data distribution. The density extraction of subgroup data is an instance of density estimation for evaluating correlations or the approximations of data distribution in a whole view. Dimensionality reduction is mainly for data compression and data simplification. For the implementation of dimensionality reduction, autoencoders are normally used for deep learning to transform input data into reduced and encoded outputs. The transformation process can also be regarded as representation learning or feature learning, which allows a deep model to automatically find out the representations and features useful for detection or classification. It is worth mentioning that manifold is one type of non-linear dimensionality reduction techniques, in which data with low dimensions can be easily plotted to show the structure of the analysed data when high-dimensional data are usually difficult to be visualized [33].

### **2.1.1.3 Reinforcement Learning**

Reinforcement learning is a critical area in deep learning, which trains an agent to take action in an environment by achieving a maximum reward. Different from supervised learning based on clear labels to learn, the agent decides the actions to discover the best possible behaviours in a specific situation, which does not merely

depend on the input data but the actions taken by the agent [34]. Reinforcement learning is regarded as an intermediate method between supervised and unsupervised learning, and actions are taken as a reward (or a punishment) if the data in the learning environment do not contain explicit labels. Typically, an agent learns from an unknown environment through a try-and-error way, which is a similar means for a child to observe new worlds. A reinforcement learning structure interacts with the environment to return a specific reward toward a changing environment. The purpose of reinforcement learning is to learn an optimal policy of maximising the rewards or other user-provided values as immediate rewards to take the best actions at every environmental transition. Reinforcement learning can infinitely occur to acquire the maximal rewards from the feedback of each section, and the feedback can directly come from the environment or can be offered by the calculation results. Two main methods of reinforcement learning are the policy search and value function: Policy search seeks the optimal values in a policy space; value functions are to estimate expected values in a given state and further attempt to select an optimal policy with maximal expected values. The policy needs to be evaluated and updated by function values. Furthermore, the quality function, also called the state-action value function, is the source of Q-learning [35], which learns the value of an action in a particular state. In terms of machine learning applications, deep neural networks can be used for policy optimization and value function approximation.

## **2.1.2 Basic Components of CNNs**

### **2.1.2.1 Convolutional Layer**

CNNs are remarkable models to solve image-related problems due to their immense effectiveness and high performance. In recent years, CNNs have been widely used in image recognition because of their extraordinary capabilities to capture accurate as well as abstract patterns from images. The convolutional layer is the main structure block of CNNs, which uses mathematical operations to merge two sets of information and serve as feature extractors to learn the feature representations from input images [36]. In general, the convolution layer is applied by kernels (or filters) to extract critical feature information. The filters aim to detect the key features from the input images, and the value and size of the kernels are not fixed, which can be designed according to the training requirements. In a convolutional layer, if an input image  $x$  of height  $H$  and width  $W$ , each element  $y_{i,j}$  in the output matrix  $y$  can be obtained by computing  $x$  with  $m \times m$  kernel  $K$ . The equation is formulated as follows [37]:

$$y_{i,j} = \sum_{k=1}^m \sum_{l=1}^m (K * x_{i:i+m-1,j:j+m-1}) \quad \begin{cases} 1 \leq i \leq H - m + 1 \\ 1 \leq j \leq W - m + 1 \end{cases} \quad (2.1)$$

where  $*$  represents the element-wise product, in which the process is similar to the concept of one-dimensional convolution [38]. The outputs of convolutional layers form feature maps, representing edges, textures, colour patterns, etc. depending on the kernels that are usually determined by learning.

Figure 2.1 is a simple example of a convolutional layer. The output is computed by element-wise product and summation. If the same size output is desired, a zero-padding method can keep the output size unchanged. If a colour image having 3 channels convolves with a  $3 \times 3$  kernel. The actual kernel size is  $3 \times 3 \times 3$ , where the first dimension is the number of input channels.

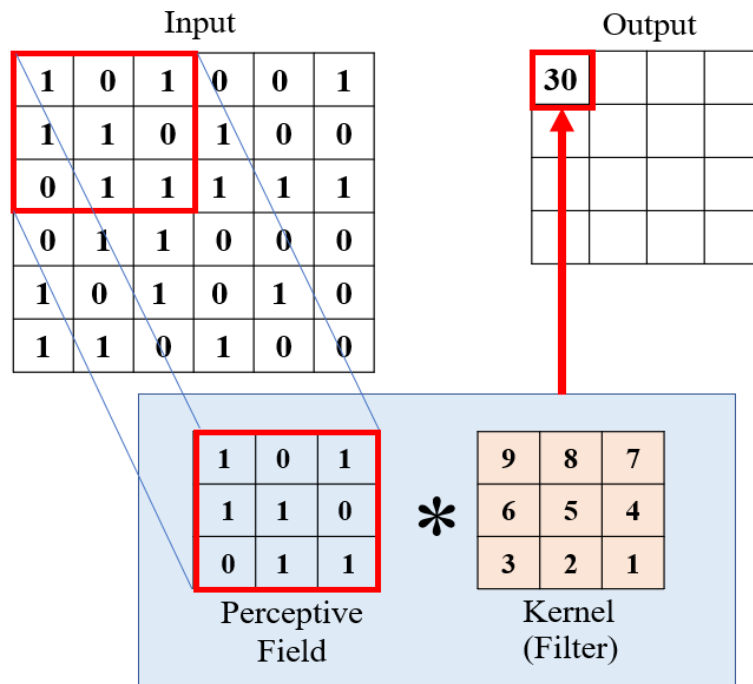


Figure 2.1: An example of a convolutional layer.

### 2.1.2.2 Regularisation Layer

Deep learning models are capable to learn complicated mapping relationships between input data and output data. However, overfitting, which leads to a good mapping in the training set rather than the test set, easily happens with limited training data [39]. The regularisation layer aims to mitigate the overfitting problem when a large

amount of data is impossible to be acquired. Many regularisation methods are proposed, including  $L_1$  and  $L_2$  normalisation, early stop, batch normalisation, pooling, dropout and so on. These regularisation methods bring about extensive improvements and inspired numerous utilisations of CNNs; the following subsections list the common methods using regularisation layers.

### 2.1.2.2.1 Pooling

Pooling layers are one of the structure blocks of convolutional neural networks. In contrast to convolutional layers responsible to extract features from images, pooling layers consolidate the features learned by CNNs. Pooling, identified as downsampling, is responsible to reduce the spatial size, variances, dimensions or computation complexity by combining the outputs of neuron clusters from one layer into the next layer [40]. The purpose of pooling is to reduce the spatial resolution of feature maps, and the adoption of pooling helps to extract the combination of feature maps. Different types of pooling formulations are used, such as the maximum, average,  $L_2$ , overlapping, spatial pyramid pooling and so on [41]. In general, the common operations are average pooling and max pooling: An average pooling layer propagates the average value of a small neighbourhood from a feature map to the next layer whilst a max pooling calculates the maximum values among a receptive field and then forwards them to the next layers. These two pooling methods provide translational invariance in image processing and preserve the detected features in small representations by discarding less important data. Max pooling and average pooling are respectively formulated as follows:

$$y_{k(p,q)} = \underset{(i,j) \in \mathfrak{R}_{pq}}{\text{Max}} x_{k(i,j)} \quad (2.2)$$

$$y_{k(p,q)} = \underset{(i,j) \in \mathfrak{R}_{pq}}{\text{Avg}} x_{k(i,j)} \quad (2.3)$$

where  $y_{k(p,q)}$  is the output of the pooling operation, which is based on the  $k^{\text{th}}$  feature map.  $x_{k(i,j)}$  is the element at location  $(i, j)$  by the pooling region of  $\mathfrak{R}_{pq}$ , where a receptive field is among the position  $(p, q)$ .

Figure 2.2 shows the output difference between max pooling and average pooling values, where the input size of the feature map is  $4 \times 4$  and the filter size is  $2 \times 2$  with a stride value of 2. Max pooling outputs the maximum values of each pooling region of  $2 \times 2$ , whilst average pooling produces the average rounded integer value.

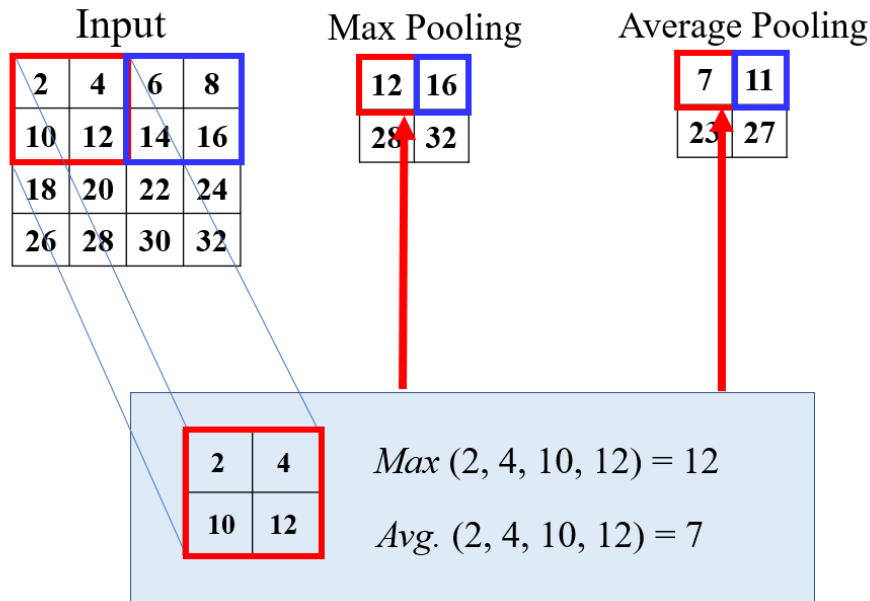


Figure 2.2: An example of max pooling and average pooling.

#### 2.1.2.2.2 Dropout

Dropout is a regularisation method, which ultimately improves regularisation by randomly skipping part of units connected with a certain probability [42]. The complicated connections may be multiply adapted by learning a non-linear relation in a deep network, which easily results in overfitting. A random dropping of some connections or units is beneficial to mitigate overfitting with the deep networks because the number of parameters can be reduced in a deep structure. Consequently, the primary advantage of dropout is the proven capability of significantly reducing overfitting and preventing feature coadaptation.

Dropout can be applied by fully-connected layers in a deep neural network. The feature selection is distributed equally across the whole neurons in fully-connected layers, and dropout forces a network to learn from other independent features. During a training process, the dropped neurons will not involve in the back-propagation and forward-propagation processes. The dropout outputs among the layers are simply expressed in Figure 2.3 and described as the following formula.

$$y = m * a(Wx) \tag{2.4}$$

where  $*$  denotes the element-wise product between a binary mask vector  $m$  and an activation vector,  $x$  is the input vector,  $W$  is the weight, and  $a$  is the nonlinear activation function.

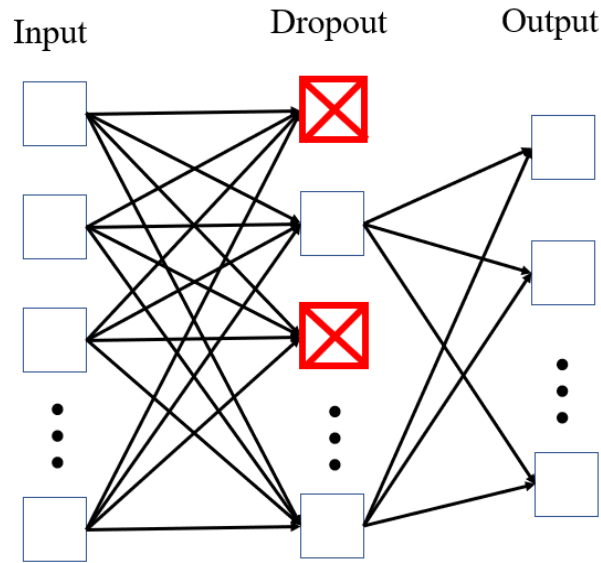


Figure 2.3: An illustration of the dropout in a fully-connected layer.

Another common dropout method is weight dropping, which is highly similar to the dropout in fully-connected layers [43]. In contrast to the dropout in fully-connected layers, the weights, which connect neurons between layers, are randomly dropped as the dropping target rather than the neurons.

### 2.1.2.2.3 Batch Normalisation

Batch normalisation uses methods of subtracting the mean values and dividing the standard deviation to normalise the outputs at each batch normalisation layer [44]. An adoption of batch normalisation allows the data distribution of input feature information normalised to a Gaussian distribution for enhancing output performance with activation functions. Batch normalisation is often adopted to reduce the situation of internal covariance shift in activation layers. The internal covariance shift is often caused by input data changed from previous layers, and the output data distribution will be correspondently shifted to the next layer. Due to the internal covariance defined by the activation functions, the internal covariance shift may become very high, and the deep networks have to take extra effort on convergence as the weights are continuously updated during training. Consequently, since batch normalisation is an efficient way to transform data distributions between layers, an operation of batch normalisation is employed to solve the problem of covariance shift and can be regarded as a standard data processing layer.

Ideally, batch normalisation is restrained by each mini-batch in a training process



[45], and the mean and variance values to each minimal batch are respectively denoted as follows:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (2.5)$$

$$\sigma_B = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (2.6)$$

where  $B$  is the number in the minimal batch,  $m$  is the number of samples of the entire training set, and  $x_i$  is the  $i^{\text{th}}$  input data. If a  $d$ -dimensional input  $x = (x^{(1)}, x^{(2)}, \dots, x^{(d)})$  is located in a layer of the network, each dimension of its input is normalised as:

$$x'_i{}^{(k)} = \frac{x_i^{(k)} - \mu_B^{(k)}}{\sqrt{\sigma_B^{(k)2} + \epsilon}} \quad (2.7)$$

where  $i$  is the  $i^{\text{th}}$  input sample in the entire training set,  $k$  is the  $k^{\text{th}}$  dimension, and  $\epsilon$  is an arbitrarily small constant added in the denominator for numerical stability. The results of the normalised activation  $x'_i{}^{(k)}$  have zero mean and unit variance. The above operation is a transform implementation of batch normalisation, and the transformation step to the next layer can be formulated as follows.

$$BN_{a^{(k)}, b^{(k)}}(x_i^{(k)}) = y_i^{(k)} = a^{(k)} x'_i{}^{(k)} + b^{(k)} \quad (2.8)$$

where  $a$  and  $b$  are learnable parameters,  $y_i$  is the output of the second transformation of the batch normalisation from the first transformation value of  $x'_i$ , and  $y_i$  will be propagated to the next layer.

### 2.1.2.3 Fully-connected Layer

Fully-connected layer indicates that neurons in each layer must have weight values associated with the connection of every neuron in the adjacent layers. Fully-connected layers are used as a probability distribution as well as dimension reduction, and they commonly occur as the last part of a deep network [46]. In classification problems, a fully-connected layer generally interprets the feature representations using the standard operation of softmax. The primary purpose of fully-connected layers is to connect the

output features by uniting layers as a final output layer [47]. It is common to use more than two fully-connected layers, and the number of full-connected layers can be decided by different requirements. However, the computation load and memory load would correspondingly increase with the number of fully-connected layers. Figure 2.4 illustrates the concept of fully-connected layers. The output of each neuron  $y_m$  in a fully-connected layer can be formulated as:

$$y_m = a \left( \sum_{n=1}^N x_n w_{mn} + b_m \right) \quad (2.9)$$

where  $N$  is the number of inputs,  $x_n$  is the output of the previous layer,  $b_m$  is the bias term, and  $a$  is the activation function.

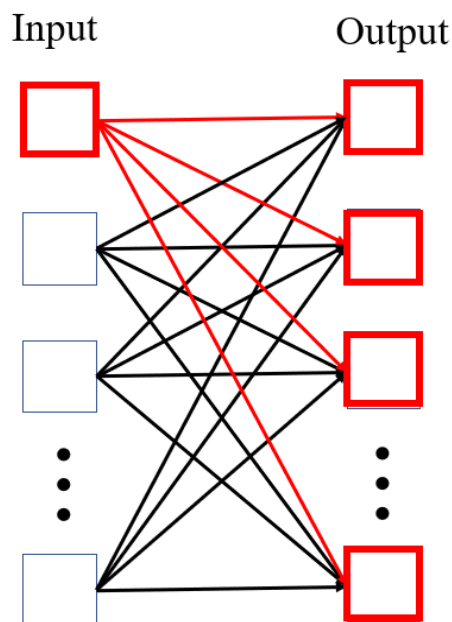


Figure 2.4: An illustration of the fully-connected layer.

#### 2.1.2.4 Activation Functions

Activation functions are one of the important components for convolutional neural networks to introduce complex mapping functions between inputs and response variables, which often take non-linear properties between layers [48], [49]. The activation function is a significant node put at the end of CNN layers to decide whether

the neurons forward to the next layer or not. The commonly used activation functions are described in the following section to compare the mathematical properties of different activation functions. Every activation function takes specific non-linearity performance to a mathematical operation.

#### 2.1.2.4.1 Sigmoid

In deep neural networks, sigmoid functions map the input data to intervals of 0 and 1. Figure 2.5 (a) shows the characteristic of the sigmoid function. The sigmoid function is used in many different applications, such as sound event detection and image processing. Furthermore, the sigmoid function confines the output probability range from 0 to 1 and can be expressed as follows [50]:

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (2.10)$$

#### 2.1.2.4.2 Hyperbolic Tangent

The hyperbolic tangent (tanh) function is defined below and shown in Figure 2.5 (b). In contrast to sigmoid functions, tanh functions map the input data from -1 to 1 [49].

$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.11)$$

#### 2.1.2.4.3 Rectified Linear Unit

The rectified linear unit (ReLU) [51] function provides neurons with nonlinearity while reducing the computation load in a gradient descent process. The output of ReLU function is positive or zero. The ReLU function is used as the activation function after the convolutional layers in the encoding stage of the noise reduction part. ReLU function is defined as follows and shown in Figure 2.5 (c).

$$\text{ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (2.12)$$

#### 2.1.2.4.4 Leaky Rectified Linear Unit (Leaky ReLU)

Leaky ReLU function is an improved version of the ReLU activation function. In contrast to the ReLU function, the output values of Leaky ReLU [52] can be either positive or negative and have a small slope of negative values rather than a flat slope. The critical advantages of Leaky ReLU are that Leaky ReLU not only solves the problem of ReLU returning zero-slope in negative input but also speeds up the training process while a balanced value between positive and negative inputs can make it learn faster. The Leaky ReLU function is defined by the following formula and shown in Figure 2.5 (d).

$$\text{Leaky ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{otherwise} \end{cases} \quad (2.13)$$

where  $\alpha$  is a constant to control the angle of the negative slope, which is available for negative input values and normally set with a small number, such as 0.01.

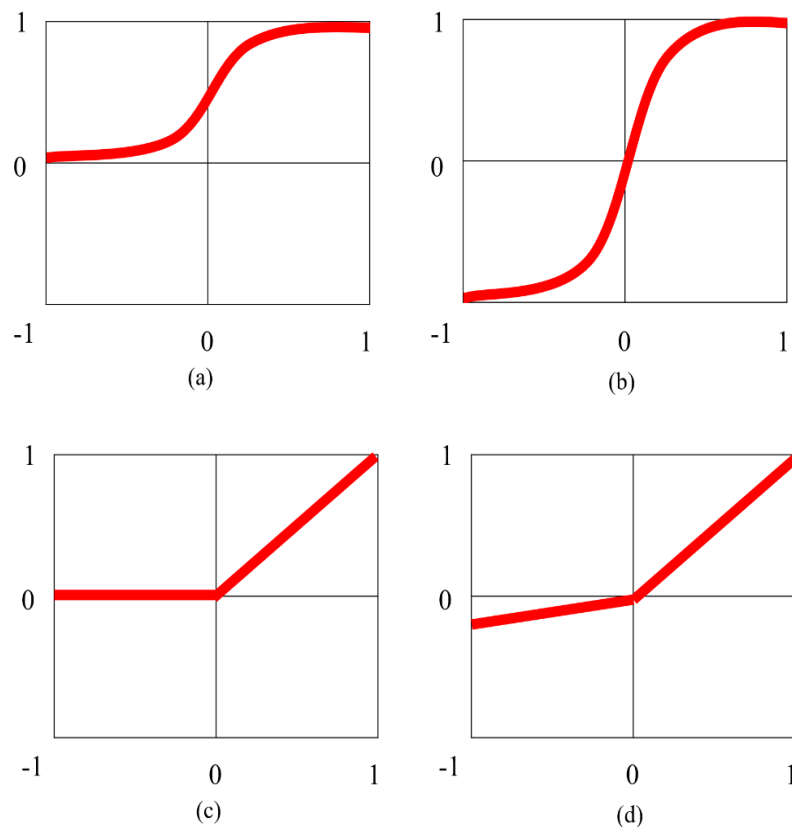


Figure 2.5: Activation function: (a) Sigmoid function, (b) Hyperbolic tangent function, (c) Rectified linear unit function, (d) Leaky rectified linear unit function.

#### 2.1.2.4.5 Softmax

The softmax function, also known as the normalised exponential function, normalises the logistic function to multiple dimensions [53]. The softmax function is employed to convert a weighted sum of inputs into probabilities summing to one, which is usually applied as the last activation function to normalise the predicted class with a probability distribution in a neural network. The probability of  $x$  belonging to the  $i$ th class is formulated as follows:

$$P(y = i | x) = \frac{e^{x^T w_i}}{\sum_{k=1}^K e^{x^T w_k}} \quad (2.14)$$

where  $x$  is a sample vector,  $w$  is a weight value,  $x^T w$  represents the inner product of  $x$  and  $w$ , and  $y$  is the softmax output of the  $i^{\text{th}}$  class. The  $P$  transforms the dimensional data of original inputs into vectors of a  $K$ -dimensional space.

### 2.1.3 CNN Architecture

The CNN architecture is composed of numerous basic components to extract different representations from training data. Many distinguished CNNs composed of these basic components have been developed and become the standard architectures with larger or deeper layers. Several important architectures commonly used as a benchmark to build up extended networks are described in the following sections. The listed architectures contain applicable advantages and have been proven through many studies or contests that they can achieve remarkable performance and have noticeable benefits in particular applications.

#### 2.1.3.1 AlexNet

AlexNet is considered a pioneering CNN architecture, which had shown ground-breaking results for image classification and recognition tasks. AlexNet was proposed by Krizhevsky *et al.* in 2013 [54] and its learning capability was enhanced by making the CNNs deeper as well as applying many parameter optimisation strategies. The basic architectural design of AlexNet is shown in Figure 2.6. A typical architecture of AlexNet encompasses more than 60 million parameters, 650,000 neurons and 630 million connections, which contains five convolutional layers, max pooling layers at three convolution layers, and three fully-connected layers. ReLU activation function is

applied at the end of every convolution layer.

In AlexNet, feature extraction is extended by adopting more convolutional layers to make them applicable to diverse categories. Despite the depth improving the generalisation for dealing with different image features, the main drawback of increasing network depth is overfitting. To address the overfitting challenge, the algorithm randomly skips some transformational units during the training phase to enforce the model learning features with robustness [55]. Compared with other proposed networks, additional adjustment of large-size filters is used at the initial layers, and ReLU is also employed as an activation function to promote the convergence rate and alleviate the problem of vanishing gradient [56]. Additionally, overlapping subsampling and local response normalisation are applied in AlexNet for improving generalisation. Based on the efficiency of AlexNet, it started a new generation in developing advanced architecture of CNNs.

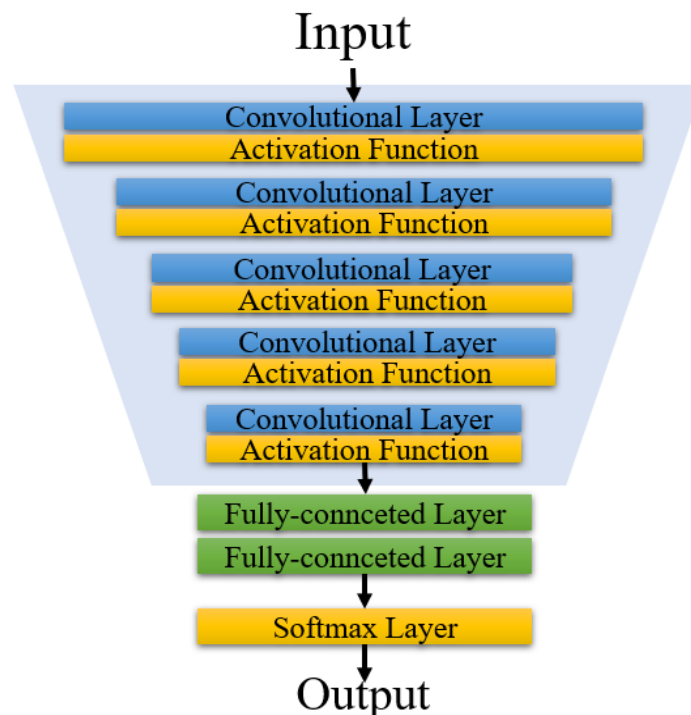


Figure 2.6: Basic architecture of AlexNet [54].

### 2.1.3.2 VGGNet

With the success of CNNs for image processing, Simonyan *et al.* proposed a simple and effective principle to design CNN architectures [57], and their architecture was named visual geometry group network (abbreviated as VGGNet). The main

contribution of VGGNet is that the depth of CNNs becomes a significant factor to achieve remarkable recognition performance and classification accuracy. Compared to AlexNet, VGGNet is made of deeper layers to simulate the relationships with a deeper representational capacity in the network [58].

The VGGNet architecture consists of convolutional layers, max pooling layers and fully-connected layers. The ReLU activation function is used, and the final layer is a softmax layer for classification purposes. VGGNet regulates the complexity of networks by placing  $1 \times 1$  convolution between two convolutional layers, which learn a linear combination of the feature maps. The max pooling layer is placed after the convolutional layers, and padding is performed to maintain the spatial resolution [59]. VGGNet changes the filter size to  $3 \times 3$  with a stride of 2 and experimentally demonstrates that  $3 \times 3$  filters can obtain better performance than the large filter size of  $5 \times 5$  and  $7 \times 7$ . Moreover, the use of small-size filters provides the additional benefit of low computational complexity by reducing the number of parameters. Consequently, these findings set a new research trend to work with smaller filter sizes in CNN developments.

VGGNet showed good results for both image classification and object localisation. There are many networks extended based on the original VGGNet, such as VGG-11, VGG-16, VGG-19 and so on [60], in which the number indicates the total number of layers in VGGNets. The main limitation associated with the VGGNet is its high computational cost. Even with the adoption of small-size filters, VGGNets still suffer from high computational loads to train more than 100 million parameters.

### 2.1.3.3 GoogLeNet

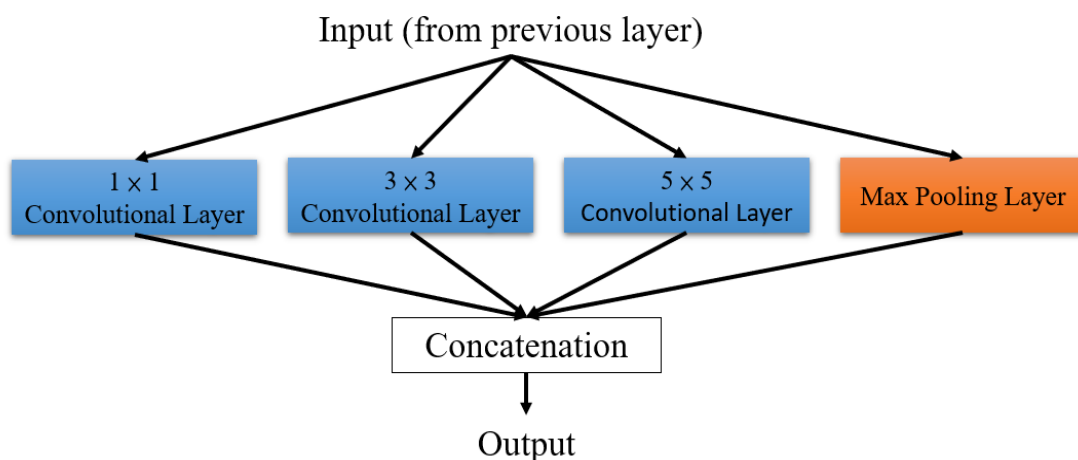


Figure 2.7: Basic architecture of the inception block [61].

GoogLeNet, also known as Inception V1, was proposed by Christian Szegedy *et al.* from Google company [61], which was the winner of the 2014-ILSVRC competition. Compared to traditional CNN, the main contribution of the GoogLeNet architecture is to reduce computation complexity [62]. GoogLeNet introduced a new concept of inception layers, which are created by different kernel sizes and have variable receptive fields. The initial architecture of inception layers is shown in Figure 2.7. The receptive fields capture sparse correlative patterns in the new feature map stack.

The inception layer uses filters of different sizes,  $1 \times 1$ ,  $3 \times 3$  and  $5 \times 5$ , to capture spatial information both at fine and coarse grain levels. GoogLeNet replaces traditional convolutional layers with small blocks, which is similar to the idea of substituting each layer with a micro neural network [63]. In addition, GoogLeNet also focuses to improve the efficiency of training parameters. Before employing large kernel sizes, GoogLeNet regulates the computation by adding a bottleneck layer with a  $1 \times 1$  convolutional filter, which uses sparse connections to overcome the problem of redundant information and neglect irrelevant feature maps. Furthermore, connection density is reduced by using a global average pooling at the last layer instead of a fully-connected layer [64]. Consequently, GoogleNet contains a deeper architecture of 22 layers than the predecessors AlexNet and VGGNet, but the number of parameters of 7 million is much lower than that of AlexNet and VGGNet, which indicates that these modifications in GoogLeNet have significant benefits in training and computation.

However, the main drawback of GoogLeNet is that its heterogeneous topology needs to be customised from module to module. In addition, another limitation of GoogLeNet is that the design of a bottleneck layer reduces the feature space passing to the next layer, which may lead to a loss of useful feature information.

#### **2.1.3.4 Residual Network**

Residual Network (ResNet), the winner architecture of ILSVRC 2015, was proposed by He *et al.* [65]. The primary contribution of ResNet is that introduces the concept of “Residual Block” in CNN architecture and devises an efficient methodology for training deep networks. The design of residual blocks aims to reduce the problem of gradient vanishing when a deep network is developed. The basic architecture of a residual block in ResNet is shown in Figure 2.8.



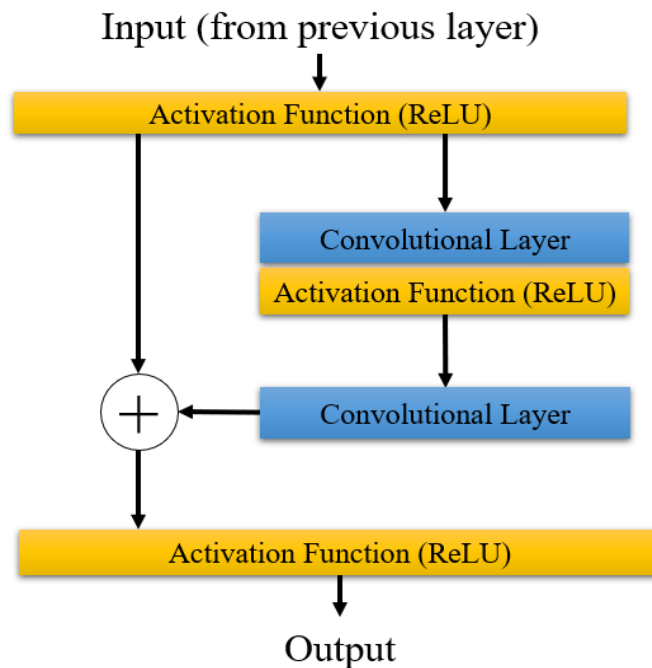


Figure 2.8: Basic architecture of the residual block [65].

The residual block is a residual connection feeding forward to the next network layer, and the inputs of a residual block can be defined by the outputs from previous operations, such as convolution with different filter sizes and batch normalisation followed by different activation functions. The ResNet is comprised of many basic residual blocks. Notably, the operations of residual blocks can be varied based on different network architectures.

ResNet has been developed with different numbers of layers, such as 34, 50, 101, 152, 1202 and so on. For instance, ResNet-50 contains 49 convolutional layers and one fully-connected layer at the end of the network for classification. Compared with AlexNet and VGGNet, ResNet is 20 and 8 times deeper respectively, but it shows less computational complexity than AlexNet and VGGNet [66]. Recently, some remarkable variants of ResNet have been proposed, which promote performance by using the idea of residual blocks. The impressive performance of ResNet-based networks shows that the depth of CNNs is a critical factor in image recognition and localisation tasks.

## 2.2 Techniques for Learning from Small Training Data

Due to plenty of parameters needed to be tuned in deep learning networks, a large amount of labelled data is one of the key factors making deep learning models reach

remarkable performance. One serious drawback of deep learning models is that the learning process always starts from a poor initial status to optimise the model, which always requires a lot of labelled data to achieve the due performance. However, compared to unlabelled data, only a small fraction of public datasets is labelled, and most of them contain copyright and usage restrictions. Consequently, techniques for dealing with the problem of training data insufficiency are explored in recent years to mitigate the impacts of labelled data scarcity.

## **2.2.1 Scarcity of Labelled Data and Class Imbalance**

### **2.2.1.1 Scarcity of Labelled Data**

Developed deep learning methods and CNNs have led to substantial progress in many computer vision applications. CNNs with many convolutional layers need a large amount of training data to fine-tune the free parameters [67]. However, the expected performance may not be easily achieved if collected data are not adequate for CNNs to learn. Due to the increasing demands on a large amount of labelled data, many practical applications are suffering from lacking sufficient labelled data for training.

Image classification is an important deep learning application relying on a set of labelled data. Convolutional layers can extract features from input images through a training process, and the possibility distributions of class scores are predicted from the extracted features. Classification algorithms have to learn critical labelled information for distinguishing object characters, such as shape, edge, colour, texture and so on, and ignore irrelevant parts [68]. The classification structures typically work well with a large amount of labelled data. However, handling labelled data scarcity and class imbalance is one of the significant issues in deep learning applications.

Various approaches and techniques have been developed to overcome the problem of data scarcity, and these methods can be divided into three major parts: data-based approaches, algorithm-based approaches and hybrid approaches [69]: Data-based approaches aim to modify data distributions of training sets to add or delete instances from training data. Algorithm-based approaches change the objective functions in a classifier to enlarge the importance of the training data. Hybrid approaches combine both data-based approaches and algorithm-based approaches to solving the data scarcity problem.

Deep networks with many layers have a very large number of parameters to be fitted, which easily leads to overfitting when trained with small datasets [70]. Data augmentation is a popular data-based approach to inflate the size of training datasets.

However, no standard techniques are available to decide whether a specific augmentation strategy can effectively improve performance until the training process is complete [71].

### **2.2.1.2 Class Imbalance**

The class imbalance problem emerges as an important issue in designing classifiers for real applications such as medical diagnosis cases where the number of positive samples is much smaller than the negative samples [72]. It is noticeable that datasets may not contain a balanced data amount in each class, which makes the classification categories unequally represented. What is worse, if a small number of instances are contained in the imbalanced class, it usually has extremely critical and significant representations for the classification task.

Many remarkable methods have been explored to deal with the problem of class imbalance, and they can be mainly divided into two categories in general, external methods and internal methods [73]: External methods aim to process training data to make them balanced. Data augmentation is a typical external method. In contrast, internal methods deal with learning algorithms to reduce the sensitivity of imbalanced classes. An advantage of external methods is that existing classifiers produced by standard deep learning algorithms can be directly used without adjusting original algorithms and structures. In comparison, internal methods adjust algorithms or structures, which may easily cause negative predictions and have serious influences on model performance compared to the original ones [74]. In contrast to internal methods, the existing learning algorithms can be directly conducted by external methods without modifications, and then the data processing will be the only concern for making the imbalanced datasets balanced.

Sampling techniques are external methods for handling class imbalance problems. Sampling can be achieved in two ways, undersampling and oversampling. Undersampling is applicable to remove samples, and the most popular method of undersampling is randomly removing the majority of class samples. However, it may lead to another problem of overfitting while a small amount of training data is involved in deep learning models. Compared to undersampling, oversampling of minority classes is a preferred method for deep networks to solve the imbalanced training data problem.

The synthetic minority oversampling technique (SMOTE) is a method to deal with the class imbalance problem [75]. In contrast to conventional oversampling techniques, which directly duplicate the minority data population instead of increasing the data information or variation, SMOTE can randomly choose data from the minority class and produce data based on the nearest neighbours of assigned data. Thus, synthetic data

will be created between the random data and the selected neighbours for enhancing the data diversity. In addition, data augmentation, another popular oversampling technique, will be further emphasised for solving the problems of class imbalance and labelled data scarcity in the next section.

## **2.2.2 Techniques for Learning from Small Training Data**

### **2.2.2.1 Transfer Learning**

Transfer learning was proposed to train deep learning models suitable for a small amount of data [76]. Training a deep learning model requires a large amount of training data, which usually is at a scale of millions of images. Transfer learning takes models trained by a large dataset and re-trains the models to other small datasets [77]. The concept is that learned weights can be transferred and generalised among different datasets if other converged networks had learned the hierarchical representations during training. Transfer learning requires further training, also regarded as fine-tuning to fit the new data. Weights of the layers for classification will be replaced when transfer learning is operated on the pre-trained convolutional neural network, and the other layers in the convolutional neural network are optionally fine-tuned.

Learned parameters from the state-of-the-art CNNs (*e.g.*, AlexNet, VGGNet, GoogLeNet, ResNet, *etc.*) are all available as alternative models for transfer learning with a small dataset. Whether transfer learning can improve performance is still debated. Many experimental results have shown the training efficiency with transfer learning, and some researchers provide experimental results that the model with transfer learning obtains superior performance compared to traditional CNN training [78], [79]. If a transfer learning method cannot efficiently promote performance, negative transfers occurred. Developing transfer learning methods to avoid negative transfers is still challenging because it is difficult to always produce positive transfers for less related tasks. To sum up, transfer learning is a prevailing technique, which is widely utilised for image classification to reduce the time consumption in model training, especially during the development period. Although it is difficult to prove that transfer learning makes deep models easily converge or improve performance, transfer learning is still an efficient technique for existing CNNs to fine-tune the trained weight values using a small number of training samples.

### **2.2.2.2 Semi-supervised Learning**

Semi-supervised learning is an approach to improve performance assessed by a small number of labelled samples along with a large amount of unlabelled or uneven data [80]. In many applications, it is impossible to acquire a large amount of labelled data as training samples. Although unlabelled data may be useful if they carry important information in prediction tasks, these unlabelled data still need human effort and expertise to process as labelled ones. Semi-supervised learning conducts a combination of supervised and unsupervised learning, which makes use of a small number of labelled samples as a training set to train a model in a supervised manner and then employ the trained model to predict the unlabelled data. In general, semi-supervised learning methods attempt to improve performance with the other associated information or data. Semi-supervised classification methods are relevant to cases where labelled data are scarce. Moreover, semi-supervised learning methods are also applied to improve classification performance when a large amount of unlabelled data contain additional representations.

Semi-supervised learning turns unlabelled data into predicted samples with a trained model, which is known as pseudo-labelling. Although semi-supervised learning and pseudo-labelling can annotate large-scale unlabelled data without human intervention, they still rely on the assumption that both labelled and unlabelled samples have the same marginal data distributions [81]. Specifically, a necessary condition of semi-supervised learning is the marginal data distribution over the input space should cover the posterior distribution information. If the input space contains no information about the posterior distributions, it is impossible to improve the performance with semi-supervised learning. Consequently, semi-supervised learning utilises a small amount of labelled data to annotate a large amount of unlabelled data and aims to resolve the performance degradation caused by training with small datasets.

### **2.2.2.3 One-shot Learning and Few-shot Learning**

The common ways to train models with one or a few labelled samples include one-shot learning, few-shot learning and zero-shot learning: The advantage of one-shot learning is using one instance to learn classes from pre-learned classes [82]. One-shot learning is beneficial for adding only one enrolled sample to train with learned classes. Few-shot learning is an extension of one-shot learning. Few-shot learning is suitable for a small number of labelled training samples, the idea of which is similar to transferring a pre-trained model trained on large data and using it in similar classification tasks with fewer training samples. The training difficulty of few-shot

learning is that a classifier needs to be generalised very well to new classes, which may not be easily achieved for a small number of training samples [83]. Zero-shot learning is an extreme case of few-shot learning and one-shot learning. Compared to few-shot learning, zero-shot learning doesn't need any visual examples of the target training classes whilst few-shot learning is supported by a few samples as the labelled categories. Zero-shot learning needs to extend the solutions of few-shot learning to update the training information through a few generated samples or auxiliary data because the features in classes are not available during the training phase.

The advantages of few-shot learning and one-shot learning are as follows [84]: Firstly, machines can learn from rare data, which can classify images with rare categories, even by collecting a very small amount of prior information. Secondly, machines can recognise the difference among very few samples, which is similar to human learning. Thirdly, the learning methods only require a small amount of data to train a deep learning model, and training with small datasets can significantly reduce computational costs. However, the drawbacks of few-shot learning and one-shot learning need to be noted that employing a small number of samples to fine-tune a deep model easily leads to overfitting, which is a challenging issue at present for reaching better performance than traditional supervised learning methods.

#### **2.2.2.4 Data Synthesis**

Good-quality labelled data are always expensive considering both time and cost, and individuals or small organisations might not afford a large amount of money to collect and maintain a large amount of ideally labelled data. Comparatively, synthetic data are freely available and fairly inexpensive for researchers to explore data that may be difficult to be acquired, such as in the fields of rare disease information and satellite photos [85]. Data synthesis is a widespread method to generate new features to acquire emulative fake data, and the common data types include sounds, images, videos and so on. Data synthesis always comes along with complicated algorithms and extensive setups, such as model designing, data testing, data validation, parameter setting, loss function, training algorithm, learning rate, optimiser and so on [86]. The main purpose of data synthesis is to meet requirements under a certain situation, in which the real data may be hard to be found or not available to be accessed in real applications. Data synthesis is theoretically used as a synthetic method to generate new types of data from realistic ones. Consequently, data synthesis often generates new targets as an acceptable benchmark to represent the original data.

There are many benefits of data synthesis compared to real data collecting [87]: First of all, due to privacy rules and other regulations, real data may be restricted in use,

and synthetic data can learn from the statistical properties of real data without exposing them. Secondly, data synthesis is either free or inexpensive regarding the collecting time and cost. Once generative models are built up, producing synthetic pictures becomes more cost-effective and faster than collecting real ones. Thirdly, if real datasets are insufficient to guarantee system performance, synthetic data can increase the size and contain representation characteristics of desired data. Moreover, synthetic data will be a unique solution to implement training or testing experiments in required systems if real data are no longer available. Finally, synthetic data can preserve the multivariate relationships of specific variables or statistics. Taking the applications of 3D data as an example, synthetic data can perfectly retain the calibrated labels or important data parameters, which may be very expensive or impossible to be collected.

Due to the introduction of GANs in 2014 [15], the applications of data synthesis have been exponentially growing. With the promotion of synthetic techniques, generative models are suitable to be used in a variety of applications, such as video synthesis, image generation and data augmentation. In contrast to the baseline of traditional synthetic techniques, synthetic data with deep learning-based approaches reach advantages in many classification tasks. Consequently, because of the progress as well as breakthrough technologies in deep learning, data synthesis based on deep learning methods has been widely applied for enlarging the data amount and diversity to augment the original datasets.

### **2.2.2.5 Data Augmentation**

It has been generally accepted that a larger amount of diverse training data can result in prominent performance in deep learning. However, collecting enormous labelled samples is an unrealistic task because of the cost and efficiency concerns [88]. Due to the constraints of privacy, ethics, security, computing resources and so on, extremely few labelled datasets are released to the public compared to unlabelled datasets, which results in collecting sufficient labelled data becoming a very difficult mission in machine learning applications. There is no doubt that deep models are hard to achieve remarkable performance with limited data. Even worse, if a small dataset is reluctantly used to train a deep model, overfitting will become a serious problem for deep learning methods, in which the overfitting problem indicates the production of analysis results is close to a particular set of training data and may statistically fail to fit other untrained data stably and reliably [89]. Therefore, training a deep learning model generally requires large amounts of data to prevent overfitting, which is a common concern when a deep model is fitted with a limited training set and makes a deep model not generalised well by the untrained data. To address the overfitting

problem, data augmentation has become a general way to increase the size and diversity of training data [90]. In terms of data augmentation, deep learning methods are the popular techniques in recent years to deal with the root problem of overfitting caused by training with insufficient samples.

The primary concept of data augmentation is to artificially inflate the size of the original training data. Data augmentation is based on the assumption that if more data information can be extracted from original training samples through data augmentation methods, deep learning models are expected to reach good performance in mitigating the negative impacts of the overfitting problem [91]. Data augmentation is also a solution to the mentioned problem of labelled data scarcity [92]. Augmentation techniques are used to design similar but alternative samples toward real data to generate extra data that may be lacking in the original datasets. For example, since convolutional neural networks may not understand objects that have been rotated or cropped, processing with rotated and cropped images will be a good augmentation technique in this case to enhance the performance of deep networks. In addition, another important issue for data augmentation is the concern of class imbalance in training a deep model [93]. Class imbalance refers to one or more classes being predicted under fewer representations in a dataset and easily makes a deep model have a bias toward over-representative classes. Therefore, data augmentation is an efficient approach to overcome the problems of class imbalance and labelled data scarcity by adding more data information over under-representative datasets. Data augmentation can be considered a regularisation technique for reducing generalisation errors in deep learning models. Regarding the importance of data augmentation in this thesis, image data augmentation and its related work will be respectively discussed in the following sections.

## **2.3 Image Data Augmentation**

Image data augmentation modifies a set of training images and additionally generates representative samples [94]. These extra images make models generalisable for improving performance on test data and avoid learning from similar features of the original training set. To make a deep network learn well, the augmented images should follow with a potential distribution of the testing set. The choice of data augmentation methods critically depends on the data types, which need to be improved or enhanced. The methods of image data augmentation can be categorised into two primary groups, traditional techniques and learning-based techniques: Traditional techniques transform images by using classic processing methods to increase the image amount. On the other



hand, learning-based techniques (also called smart methods or learnable augmentation methods) first learn the data distribution from the original training samples and then create images from the learned data distribution for augmenting the original datasets.

### **2.3.1 Traditional Augmentation Techniques**

A challenge to computer vision applications is how to obtain robust representations and diverse visual features to remain unaffected in transformations. The influence of the real environment, *i.e.*, occlusion, deformation, light intensity, *etc.*, may cause transformations that easily result in image feature changes over the object's appearance. Traditional techniques for mitigating the negative transformations are to increase the data diversity, and augmentation techniques have become common methods to be employed in many computer vision applications. Image data augmentation includes two major transformation methods, geometric transformations and photometric transformations, both of which consist of classic image manipulation processes [95]: Geometric transformations map images with different spatial positions; photometric transformations manipulate the intensity values to produce augmented images. The data augmentation processes are performed on the original images, and the modified results will add back to the original dataset to increase the diversity of the original datasets. Several popular and classic augmentation techniques are listed in the following sections, where an overview of geometric transformations as well as photometric transformations will be further discussed.

#### **2.3.1.1 Geometric Transformations**

Geometric transformations, such as scaling, translating, rotation, flipping, reflecting, shearing, cropping and so on, are the most common and easy augmentation techniques [96]. Since geometric transformation strongly relies on data likelihood, the transformations should confidently refer to the data similarity without causing a label transformation. Therefore, the policy in different cases needs to be carefully considered whether the adopted transformation techniques are beneficial for improving the final performance. For instance, in terms of the traditional augmentation applications, rotation and flipping are common techniques to generate different images with data similarity but may not be confident in the case where the sign of 9 may be recognised as 6 for different digit labels through rotation and flipping. In the following sections, several common geometric transformation techniques with different processing methods are described.

### 2.3.1.1.1 Flipping

Horizontal and vertical flipping are common methods in geometric transformations because this augmentation technique is one of the easiest ways to be implemented [97]. Flipping helps to maximise the image number without complicated image processing, and it has proven advantageous to enlarging the diversity of datasets, such as applied in datasets CIFAR-10 and ImageNet. Samples using the flipping technique are shown in Figure 2.9.

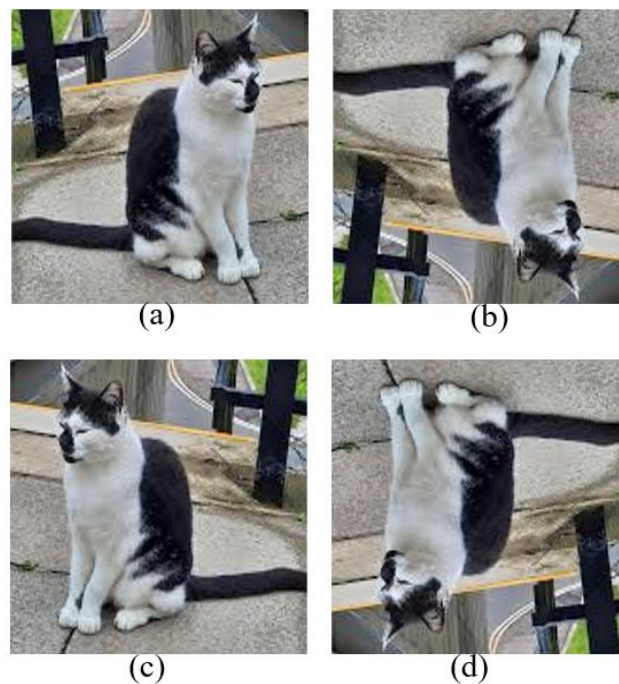


Figure 2.9: Flipping technique, where (a) is the original image, (b) is vertical flipping, (c) is horizontal flipping, and (d) is vertical and horizontal flipping.

### 2.3.1.1.2 Rotation

Rotation is another type of geometric transformation for common data augmentation requirements [98]. Rotation is done by turning images in right or left directions based on the axis between 1 to 359 degrees. An efficient augmentation technique of rotation is heavily designed by the degrees of rotation parameters. Slight changes, for instance from 1 to 15 degrees or -1 to -15 degrees, could be more useful than large degrees of rotations. If the rotation degrees increase on a large scale, label information may be no longer preserved in various datasets. Samples of rotated images are illustrated in Figure 2.10.

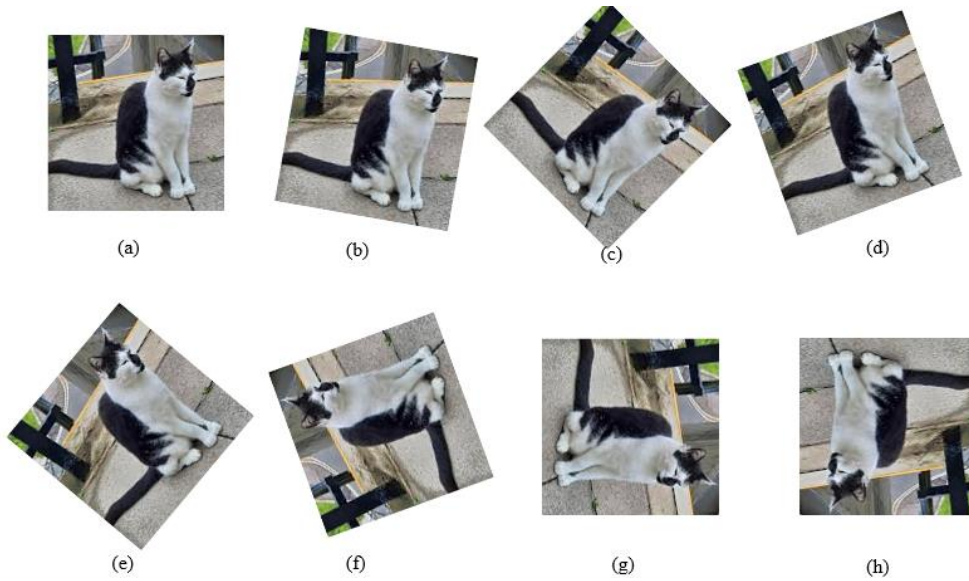


Figure 2.10: Samples of rotated images.

### 2.3.1.1.3 Translation

An operation of translation is to shift images in different directions, such as up, down, right and left [99]. The translation techniques beneficially avoid positional bias in normalised datasets. Taking facial recognition tasks as an example, if a face dataset put all the facial images in the centre, it will be useful to image pre-processing as well as normalisation. However, a deep model may receive good features only on centred images. In this case, the original images can be translated into different directions except for the centre, and the other remains could be filled with specific values or random noise in terms of data augmentation. Samples of translation images are shown in Figure 2.11.

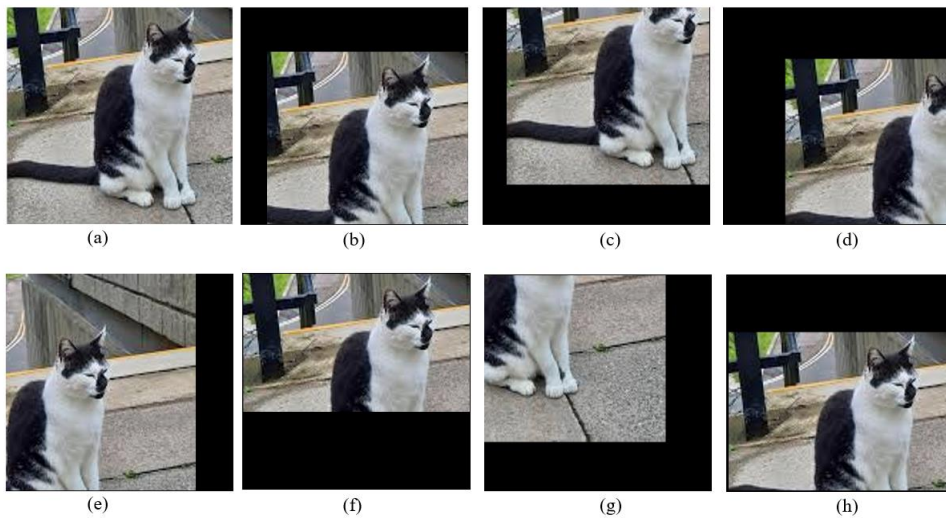


Figure 2.11: Samples of translation images.

### 2.3.1.1.4 Cropping

Cropping is an image processing method, which is practical to mix the height and width by cropping images at a central patch of images [100]. Cropping could be analogous to zooming or scaling images as well. Two types of cropping are usually used for data augmentation: 1) two locations in images need to be set as a starting and ending point. 2) two range values of the height and width are used to rescale images. In addition, random cropping can be used to provide various similar outcomes of augmented data. Compared with other techniques, cropping will reduce the size or resolution of the original input images, but the other techniques preserve the spatial dimensions of the original ones. Samples of cropping images are demonstrated in Figure 2.12.

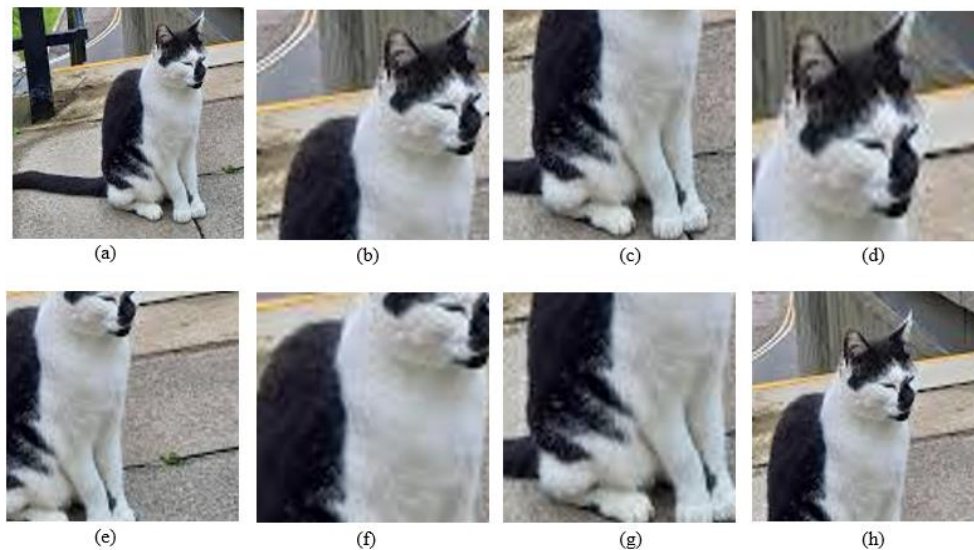


Figure 2.12: Samples of cropping images.

### 2.3.1.2 Photometric Transformations

#### 2.3.1.2.1 Noise Adding

Adding noise is a technique to insert a noise matrix, which is created by random values [101]. Different from geometric transformations of changing positions presented in training data, adding noise is an efficient solution for data augmentation to add or change the data distribution of images to make deep models learn more robust features. For data augmentation purposes, the common noise types added in target images are Gaussian noise and salt & pepper noise, as shown in Figure 2.13.



Figure 2.13: Sample images of noise added by different percentages. (a) to (d) is the images with salt & pepper noise, and (e) to (h) add noise with Gaussian distribution.

### 2.3.1.2.2 Colour Space Shifting

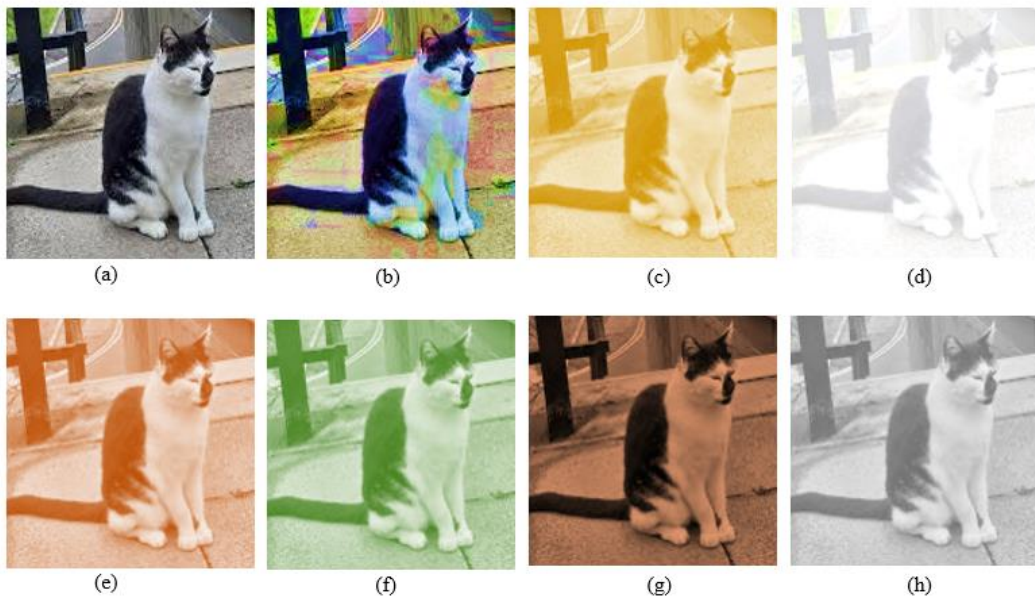


Figure 2.14: Sample images of colour space shifting.

Shifting colour spaces is a technique by changing the pixel values instead of pixel positions for data augmentation requirements [102]. Humans can distinguish objects via their colour properties, such as brightness, contrast, saturation, hue, lighting and so on. In photometric transformations, colour space transformation is one of the critical

techniques to increase the diversity of pixel values, which can not only enlarge the number of images but also discover significant features hidden behind a specific colour space. For example, a constant pixel value can be quickly added or subtracted by changing the image brightness (or darkness), and the transformations to a certain colour space involve a restriction of pixel values in digital images, which meets the requirements of generating diverse colour representations for data augmentation purposes. Sample images of colour space shifting are illustrated in Figure 2.14.

### 2.3.1.2.3 Kernel Filter

Kernel filters are the techniques to process images by controlling pixel values with kernel filters, such as histogram equalisation, sharpening, blurring and so on. The kernel filter works by sliding a kernel matrix across whole images [103]. For instance, histogram equalisations adjust intensity values to enhance image contrasts. Image sharpening is based on a high-contrast vertical or horizontal edge filter to boost the edge details and image blurring uses averaging processes to blur pixel values. The use of different kernel filters can result in diverse images with distorted or sharpened outcomes. Kernel filters are popular and helpful to augment original image data with high data diversity. Sample images using different kernel filters are shown in Figure 2.15.



Figure 2.15: Sample images using different kernel filters.

### 2.3.1.2.4 Random Erasing

Random erasing is one of the image data augmentation techniques, which fundamentally erases some pixel values of images [104]. Random erasing can be regarded as a specific design to combat the recognition challenges of image occlusion referring to missing or unclear parts of image information. Apart from the occlusion challenges, random erasing is a convincing technique for making a deep network focus on entire images rather than preferentially learning from certain visual features.

Image erasing generally works on square regions and masks images with specific pixel values, such as mean values, random values, maximum values, assigned values and so on. Random values have been found as well-chosen values in image data augmentation. Therefore, random erasing becomes a popular augmentation technique to directly prevent overfitting by altering images. However, the disadvantage of random erasing is also obvious it may not be a good technique to preserve labels. For example, a label error may happen by random erasing when a digit sign of 8 transforms into 9 in a number recognition task. Sample images of random erasing are shown in Figure 2.16.

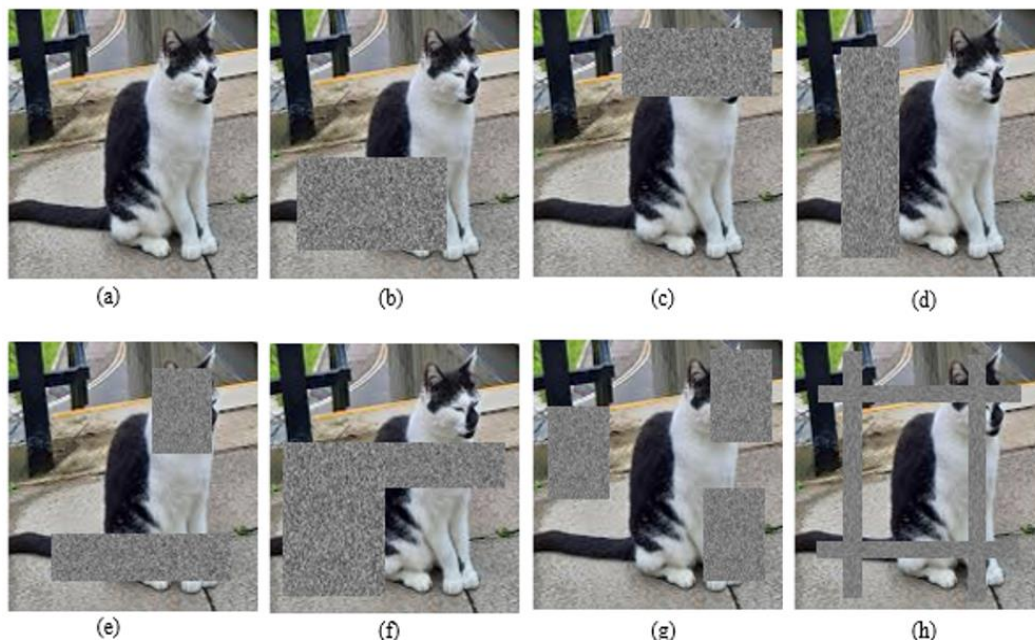


Figure 2.16: Sample images of random erasing.

## **2.3.2 Image Data Augmentations Based on Deep Learning**

### **Methods**

Since many deep learning methods have made breakthroughs in recent years, it is expected to use deep learning techniques to solve the problems of labelled data scarcity and class imbalance. Image data augmentation techniques based on deep learning approaches have shown advantages in many studies.

#### **2.3.2.1 Meta-metric Learning**

Meta-metric learning is a framework accompanied by more than two deep networks for objective reconstruction with loss functions [105]. The concept of meta-metric learning is to use a deep network to optimise other deep networks. Meta-metric learning conducts regression approaches to promote training precision and further minimise the overfitting problem. Meta-metric learning is one of the deep learning techniques applicable in the field of data augmentation and has successfully built deep models to produce image data by reducing failures through a training phase. Meta-metric learning substantially improves the precision of training a deep model, which is efficient to mix non-homogenous image features with a strong generalisation. However, compared with traditional data augmentation techniques, the drawback of meta-metric learning is that the training efficiency needs to be tested and proven by experts. Moreover, the implementation of meta-metric learning is relatively time-consuming in practical applications.

#### **2.3.2.2 Feature Space Augmentation**

Feature space augmentation uses neural networks to map high-dimensional inputs into lower-dimensional representations [106]. Feature space augmentation is easily implemented by autoencoders, in which new instances can be reconstructed from input features. Representative features are possibly processed by isolated vectors with a specific feature space, also denoted as a latent space, which is conducted by decreasing the output layers of a neural network. For instance, the outputs of a deep network could be low-dimension representative vectors instead of class labels. Autoencoders work well on mapping images into low-dimension representative vectors, and decoders can reconstruct these representative vectors back to the original ones. The use of representative vectors is profitable for discovering a new feature space based on the



input features.

The disadvantage of feature space augmentation is hard to interpret the whole vector information through neural networks or other deep networks. Although feature space augmentation is implementable to map input images into representative vectors with an autoencoder, the requirement of reconstructing entire encoded parameters will be extremely difficult to process by the decoder structure, which is not only time-consuming but also hard to train with high precisions.

### **2.3.2.3 Augmentation Using Generative Adversarial Networks**

The universal data augmentation techniques based on deep learning methods utilise generative models, which can create artificial data from initial datasets, to produce data and enhance the performance of deep learning applications [107]. One of the popular generative models is generative adversarial networks (GANs). GANs are typically composed of two distinct deep networks, the generator and discriminator, and extract features by a competitive learning mechanism to learn representations from real data [108]. Many studies have shown that GANs not only are simple and useful as a data-driven manipulation strategy to produce additional data but also have caused major changes in many deep learning applications, such as image synthesis, style transferring, image segmentation, imaged editing, super-resolution manipulation and so on [109]. Since GANs are the primary generative models used in this thesis, an overall literature review will be presented in the next chapter, which includes the GAN theory, structure variants, loss functions, training difficulties and applications.

## **2.4 Conclusion**

In this chapter, significant topics of machine learning are reviewed in terms of learning data representations efficiently from a small training dataset. Firstly, three different categories of machine learning methods, including supervised learning, unsupervised learning and reinforcement learning, are discussed. Secondly, since convolutional neural networks (CNNs) are one of the state-of-the-art methods in deep learning applications (*e.g.*, image classification, object detection, object recognition, *etc.*), a comprehensive review of CNNs is presented in Section 2.1. Thirdly, the basic components of CNNs are presented along with different functional layers and activation functions. In addition, several important CNNs proposed over the past few years as well as the contributions of these networks, including AlexNet, GoogLeNet, VGGNet and ResNet, are reviewed. Finally, despite the success and remarkable advances in CNNs,

they rely on massive labelled datasets for achieving outstanding performance. In general applications, the scarcity of labelled data and datasets with class imbalance have become serious problems for deep learning to reach expected performance and avoid overfitting.

To deal with the problems of labelled data scarcity and class imbalance, techniques for learning from small training data and image data augmentation are respectively reviewed in Sections 2.2 and 2.3. The techniques described in Section 2.2 include transfer learning, semi-supervised learning, one-shot learning & few-shot learning, and data synthesis & augmentation. Among the above methods, data augmentation is one of the most useful techniques to produce improved datasets, which can be suitable for all existing CNNs without modifying the existing model structures and algorithms. Therefore, in Section 2.4, image data augmentation is further discussed in two main parts: the first part is traditional methods, and the other is deep learning methods. Traditional techniques of image data augmentation consist of geometric transformations and photometric transformations. Comparatively, the deep learning techniques for image data augmentation include meta-metric learning, feature space augmentation and GANs. Since this thesis aims to propose novel GAN models to automatically augment image data from small datasets, a comprehensive review of data augmentation using GANs will be presented in the next chapter.

# Chapter 3

## Literature Review on Generative Adversarial Networks

### 3.1 Introduction

Novel generative models have been proposed and dedicated to specifically discovering data distributions with probability and statistics methods. In general, these generative models can be grouped into three main categories: variational autoencoders (VAEs), auto-regression networks, and generative adversarial networks (GANs) [110], [111]. Firstly, VAEs are probabilistic models and attempt to model the probability distribution of real data. However, the outcomes of probabilistic simulations usually have a bias, which makes the generated samples blurry. Secondly, auto-regression networks translate image generation into a pixel prediction task, and each pixel needs to be processed one by one. Finally, GANs are composed of two primary networks, the generator and the discriminator. In an adversarial learning process, a generator, which is responsible to generate fake samples from random noise, creates data to fool the discriminator. Oppositely, a discriminator classifies samples and distinguishes between real data and fake data. The learning goal of a generator is to fool the discriminator to believe that the generated samples are real. On the other side, a discriminator is trained by both real and fake samples to identify the samples generated by the generator as fake.

GANs are not only one specific type of generative model based on learning techniques but also one of the most popular models, which discover the maximum likelihood and approximate inference of real data distributions [112]. In contrast to other generative models, in which a large number of parameters need to be fine-tuned for discovering the approximate distributions of real data, both VAEs and auto-regression networks contain serious generalisation issues and limited processing efficiency. Moreover, the restricted frameworks, *e.g.*, autoencoder for VAEs and no latent variables for auto-regression networks, deteriorate the generative capabilities and lead to unclear results. GANs provide the following advantages for efficiently producing desired samples [113]: 1) Compared to VAEs or auto-regression networks, GANs can produce more realistic outcomes. 2) GAN frameworks have good compatibility with deep neural networks and other existing deep models. In contrast to other generative models, GANs do not need a pre-requirement for specific network

frameworks, and the flexible frameworks are suitable for various types of real applications. 3) GANs can generate many types of probability density whilst VAEs and auto-regression networks are difficult to produce diverse and different types of synthetic results. 4) The restriction on the size of latent variables of GANs is less than VAEs and auto-regression networks, which makes GANs more efficient for various real applications.

Even though GANs are not the perfect models to deal with all comprehensive generative problems and have potential drawbacks as discussed in Section 3.5, the advantages mentioned above still lead GANs to great success, especially in the field of computer vision, such as the thriving applications of image synthesis, image segmentation, image translation, super-resolution, and so on [114]. In this chapter, the topic of image synthesis based on GANs is comprehensively introduced and discussed in 6 sections, including fundamental framework, structure variants, loss function variants, training challenges, evaluation metrics and applications.

## 3.2 A Fundamental Framework of GANs

### 3.2.1 Typical GANs

The idea of GANs was introduced by Goodfellow *et al.* in 2014 [15], and a typical GAN framework is composed of two neural networks, namely the generator and discriminator. The conceptual idea of the GAN structure is shown in Figure 3.1. The common analogy is to take one network as an art forger and the other as an art expert. The generator  $G$  is treated as the forger to create forgeries and aims at making realistic images. The discriminator  $D$  receives both forgeries and real data and aims at distinguishing between them. GANs have been widely used in many synthetic fields. Meanwhile, new methodologies have also been proposed in recent years to make model training stable and generate high-quality results for broadening the applications of GANs.

Through a training phase, the generator generates highly realistic samples that can deceive the discriminator. Besides, the discriminator also tries to improve the capability to recognise real and fake data. The adversarial learning between the generator and discriminator networks can be considered as a completed status when both networks stop improving and stay in equilibrium [115]. The training process resembles a cat-and-mouse game. By competing with each other between the two networks, a GAN framework can produce realistic synthetic data with adversarial learning.

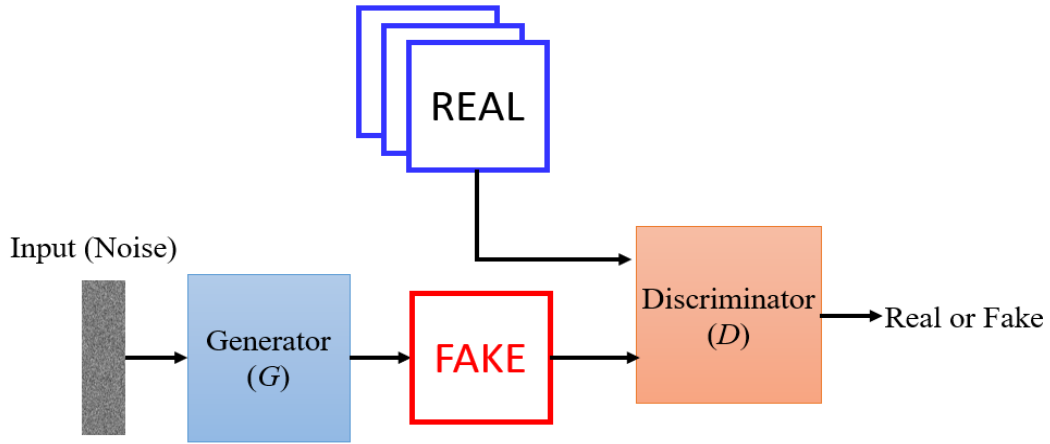


Figure 3.1: Conceptual idea of the GAN structure [15].

To be specific about the training process, the discriminator is characterised as a mapping network to discover the data distribution from the real data. The discriminator is trained to classify the training data and recognise fake ones. On the other side, the generator continues to be trained to lower the accuracy of the discriminator when the discriminator is optimal. If the generated data distribution in the generator perfectly matches the real data distribution, then the discriminator will be maximally confused by the generated data. Due to the generator having no access to real images, the information from the discriminator is the only way to interactively learn. In contrast, the discriminator has access to both fake data and real data, so that the information provided by the discriminator knows whether the data came from real or fake, and the information can be also used to train the generator to produce forgeries with quality improvement. Formally, the minimax operation between generator  $G$  and discriminator  $D$  with the loss function (objective function)  $V(D, G)$  is formulated as follows [15]:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim P_z(z)} [\log (1 - D(G(z)))] \quad (3.1)$$

where  $P_{data}$  is the distribution of real data,  $P_z$  is the noise distribution, and  $P_z(z)$  indicates the data distribution from input noise  $z$ , which follows uniform distribution or Gaussian distribution,  $x$  is the real data, and  $\mathbb{E}$  means the expectation value. Initially,  $G$  accepts a data distribution from random vectors  $z \sim P_z$  and generates synthetic samples from the certification of  $D$ . The parameters of  $G$  are then fine-tuned and updated by using the signals from  $D$  through back-propagation.

### 3.2.2 Convolution-based GAN

Both generator and discriminator in a typical GAN framework can be deep neural networks or other machine learning models. Convolutional neural networks have demonstrated outstanding performance in many image-processing applications. Therefore, the convolution-based framework with a deep convolutional generative adversarial network (DCGAN) was formalised in 2016 [116], which can not only produce high-quality images but also has the advantage of stabilisation during training. Consequently, the DCGAN structure has been widely used as the fundamental GAN framework in many image-processing applications. One of the disadvantages of the DCGAN is that a large amount of labelled data is required for the DCGAN to achieve photorealistic results. Additionally, the mode collapse problem frequently appears when the model is over-fitted to a few samples, leading to an oscillating mode.

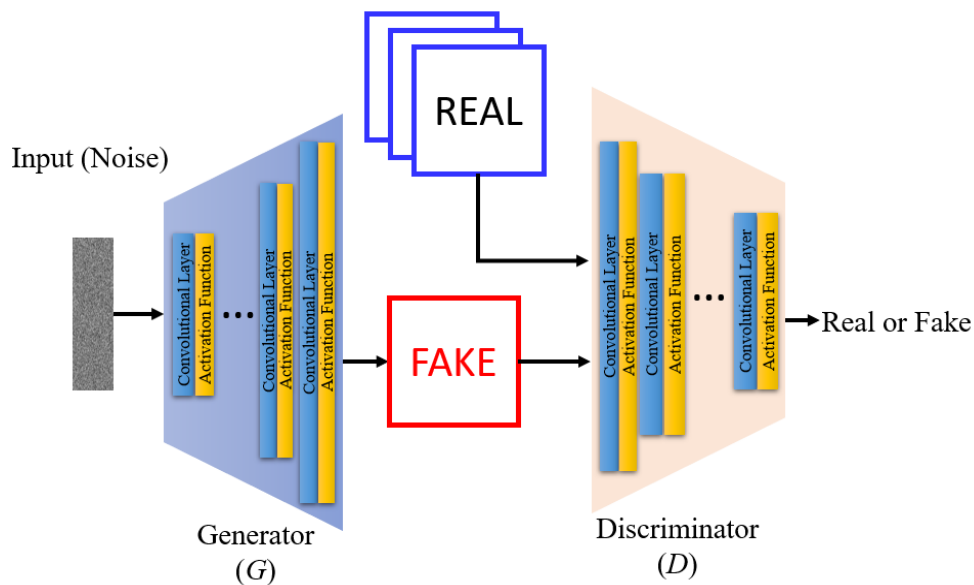


Figure 3.2: The structure of a deep convolutional generative adversarial network.

Figure 3.2 shows the structure of a DCGAN, which uses a high-dimensional uniform distribution for generating the noise vector to extend a small spatial convolutional representation with various feature maps. Each convolutional component at least contains a convolutional layer and an activation function as the basic components, and the convolutional process converts the input signals of the noise vector to generated images with a high-level representation, such as colour images with  $3 \times 64 \times 64$  pixels.

## 3.3 Structure Variants of GANs for Image Synthesis

A typical GAN learns the real data distribution from training samples and then generates demanded distributions of real-like data. However, the basic GAN structure might not be strong enough to learn complex data distributions over various application tasks. Many variants combined with different network structures have been proposed in recent years for improving the efficiency and effectiveness of image synthesis, leading to various GAN structure variants, which are expected to be more functional and efficient to deal with complex data distributions. Several structure variants of GANs, including condition-based GAN, auxiliary classifier GAN, autoencoder-based GAN and attention-based GAN, are reviewed in the following subsections, where the improvements and benefits for image synthesis will be discussed.

### 3.3.1 Condition-based GAN

In the basic structure of typical GANs or convolution-based GANs, both of their inputs in generators are random noise vectors, which easily lead to mode collapse and other negative impacts on the model training. For mitigating these drawbacks, a variant structure, condition-based GAN or conditional GAN, was proposed in 2014 [117]. The basic structure of a condition-based GAN is shown in Figure 3.3. In contrast with typical GANs and DCGANs, condition-based GANs have an extra conditional variable, which could be obtained from labels, texts, images, or other condition data. The conditional variable presenting in a condition-based GAN structure can be inputted into both generators and discriminators, which efficiently assists the generator to synthesise more reliable and less collapsed outcomes than merely inputting noise vectors. For a generator, the conditional variable as the auxiliary conditional information is determined by the feeding data combined with random noise to discover the hidden representations. Comparatively, for a discriminator, the conditional variable is also presented as the input data of the discriminative process. In addition, the loss functions of condition-based GANs are similar to the typical GAN, but an additional data distribution of the conditional variable should be considered during training.

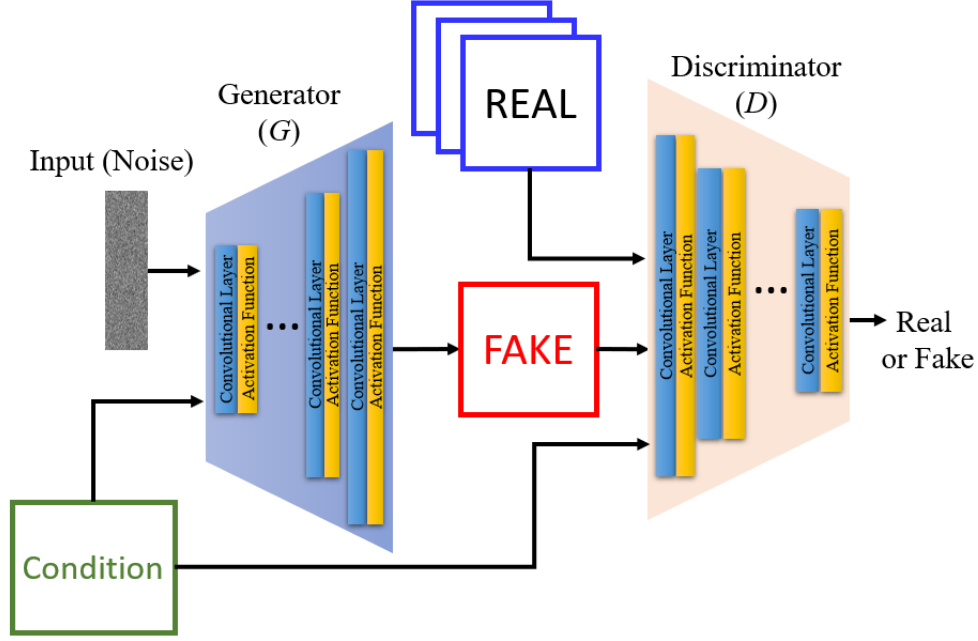


Figure 3.3: The basic structure of a condition-based GAN [117].

Compared to unconditional generative models difficult to control the created results, condition-based GANs provide a novel idea of directing information with additional condition variables, which can control the properties of the generated data and mitigate the mode collapse problem. The condition-based GANs have shown promising and interesting synthetic results in many applications (*e.g.*, image translation, image repairing, *etc.*) [118]. However, the conditional variable applied by condition-based GANs still relies on good labelled data based on supervised learning. The input conditions are taken as the probabilistic distribution, and prediction errors or mapping mistakes could result in a huge impact on generative quality.

Conditional variables are fed as an additional input layer in condition-based GANs. In other words, the original input noise and conditional variables are combined as new hidden representations, which allow a condition-based GAN to additionally consider the composition of the hidden representations. The loss function of condition-based GANs is formulated as follows:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}(x)} [\log D(x|y)] + \mathbb{E}_{z \sim P_z(z)} [\log (1 - D(G(z|y)))] \quad (3.2)$$

where  $x$  is the input data,  $y$  is the condition data,  $z$  is the input noise,  $P_z(z)$  is a data distribution generated from noise, and  $\mathbb{E}$  is the expectation value.



### 3.3.2 Auxiliary Classifier GAN

The fundamental structure of an auxiliary classifier GAN, abbreviated as ACGAN, is very similar to a condition-based GAN. The slight difference lies in that the auxiliary classifier GAN extends the structure of condition-based GAN with an additional auxiliary classifier, and this structure variant was first proposed in 2017 [119]. The basic structure of an auxiliary classifier GAN is shown in Figure 3.4. The main concept of the auxiliary classifier GAN is to attach an extra network as a classifier to help the discriminator classify complex data, and the auxiliary classifier is extensively used to extract complex features. A pre-trained model, which is trained with other big datasets instead of the provided datasets, can be used as an auxiliary classifier [120]. The adoption of an additional auxiliary classifier is expected to boost the capabilities of feature recognition in image synthesis. Although the improvements by using an auxiliary classifier in GAN structures could be valuable to generate great visual quality or highly diverse images, the drawback is obvious when the auxiliary classifier needs a large scale of labelled datasets to improve the performance. Therefore, auxiliary classifier GANs still need to face the general challenge of labelled data scarcity in real applications.

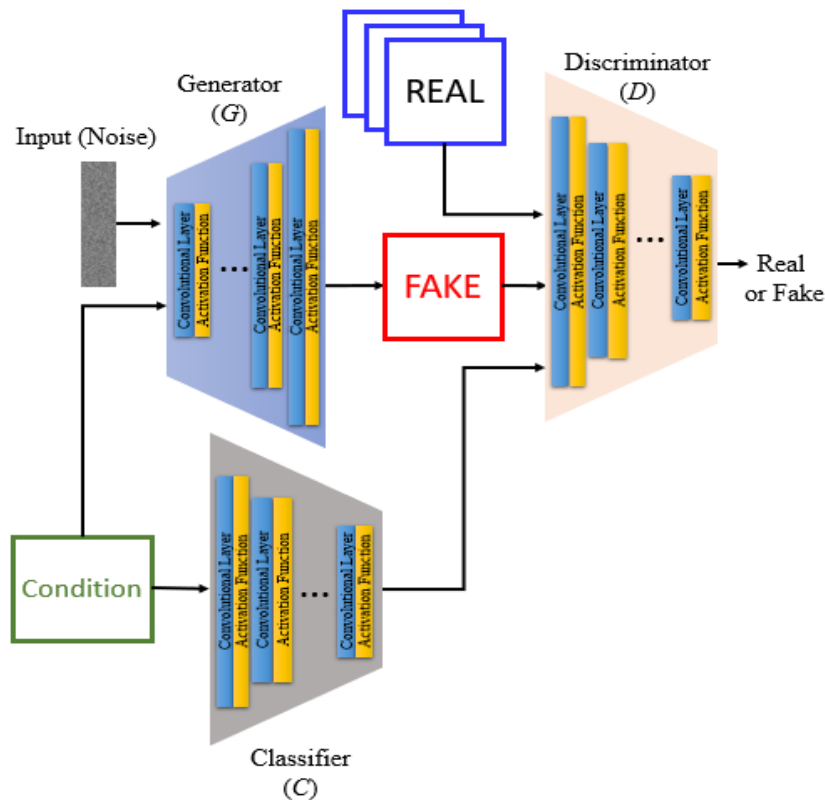


Figure 3.4: The basic structure of an auxiliary classifier GAN [119].

### 3.3.3 Autoencoder-based GAN

Autoencoder is a typical structure of neural networks that are trained to produce a latent space (or a hidden layer) and then reconstruct data from the latent space. In general, an autoencoder consists of two main components, encoder and decoder. The encoder is used to project input data onto a latent space for decreasing the dimension of input data, and the decoder uses the vectors received from the latent space as its inputs to recover the original data.

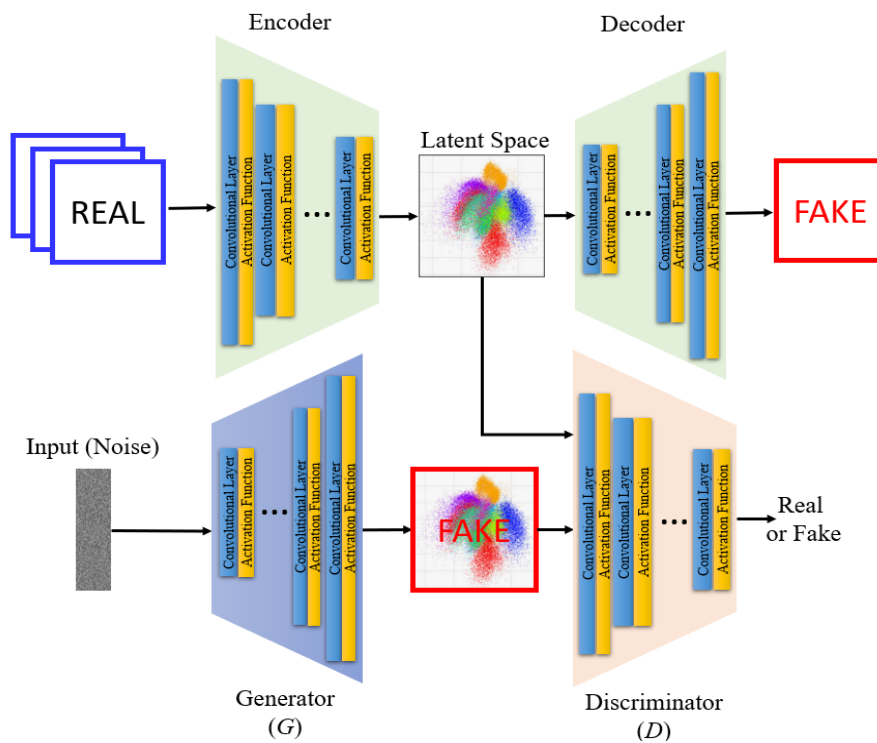


Figure 3.5: The basic structure of an adversarial autoencoder [121].

A disadvantage of autoencoders is that a latent space produced by encoders may not eventually be distributed well, which will bring about a large number of synthetic gaps in final data distributions. Many researchers still work on overcoming this disadvantage, and a new structure variant combining a GAN structure with an autoencoder, represented as the adversarial autoencoder, was proposed in 2016 [121]. With the adoption of the latent space, the input data distribution can be further imposed to mitigate the synthetic gaps in the adversarial autoencoder structure, which will ensure that the gaps in synthesised images can be reduced and force a decoder to produce more meaningful and realistic samples. Figure 3.5 shows the basic structure of an adversarial autoencoder, where the latent space represents a data distribution generated by the

encoder. Furthermore, the generator uses noise signals to synthesise a specified distribution similar to the data in a latent space. The discriminator is designed to recognise the real or fake data from both the encoder and the generator. After training, the encoder can learn the expected distribution, and the decoder can finally generate the fake samples, which are reconstructed by the required data distribution in the latent space.

With recent developments, modern generative models conjunct autoencoder structures and GAN structures with a shared latent space. Compared with the mentioned variants of condition-based GANs and auxiliary classifier GANs, the labels are not necessary for an autoencoder structure because autoencoder-based GANs can be designed as an unsupervised mechanism in image synthesis. In addition, many remarkable models only employ an additional encoder in a GAN structure. If the GAN generator can automatically learn features from a latent space to capture the changes in real data distributions, it will beneficially reduce the data requirements on labels or other conditional information.

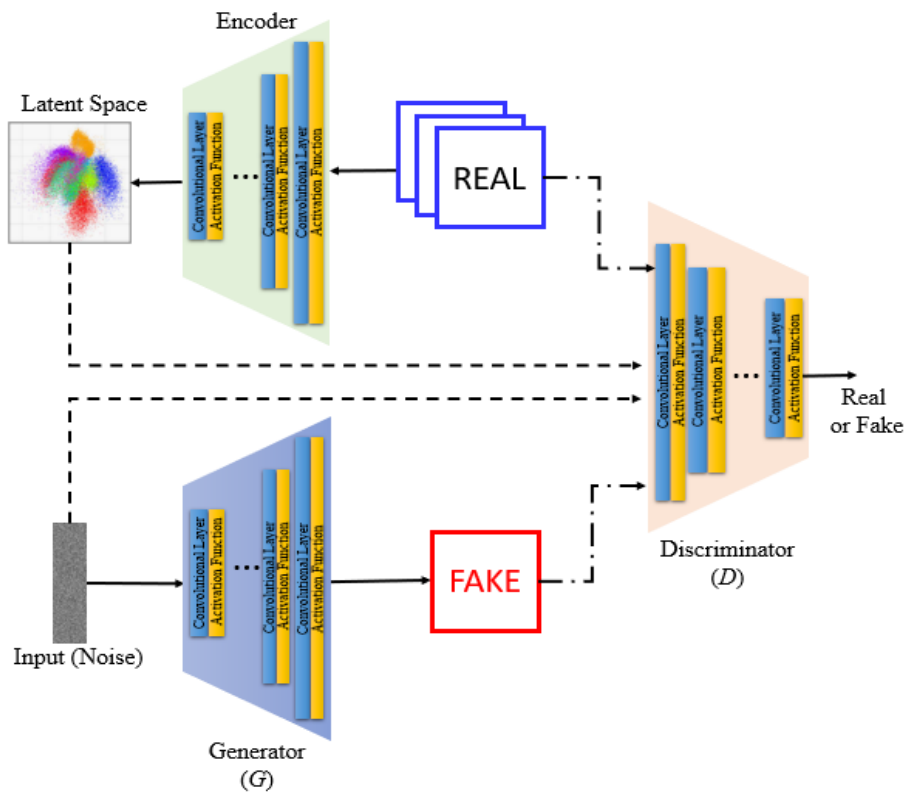


Figure 3.6: The basic structure of BiGAN [122].

Due to the pros and cons of the autoencoder structure, many models partially adopt encoders into GANs; this type of GAN structure variant uses the encoder as a tool to capture features in a discriminator. The generator can learn the features in a latent space

generated by an encoder and capture the semantic changes for the data distributions. Nevertheless, this GAN variant cannot learn the mapping relationships of real data distributions. To address this problem, BiGAN was proposed in 2017 to make valid inferences for generating high-quality samples [122]. The basic structure of the BiGAN is demonstrated in Figure 3.6. Besides the generator and discriminator, an encoder is additionally adopted in the proposed model. The encoder is used to inversely map data back to a latent space and evaluate the difference between the paired encoder and generator data. Since the encoder and generator do not communicate directly, the generator needs to learn to inversely fool the discriminator.

Another remarkable model using an encoder in the generator was proposed in 2018, named adversarial generator-encoder network (AGE) [123]. AGE applies adversarial learning between the generator and encoder, but this structure does not contain discriminators. In AGE, the generator is to reduce the gap between the latent distribution and synthetic data distribution whilst the encoder aims to maximise the divergence between latent and synthetic data. Figure 3.7 demonstrates the AGE structure, where  $R$  denotes the reconstruction loss, and the function of reconstruction loss  $R$  is expected to avoid the possibility of mode collapse and other training drawbacks.

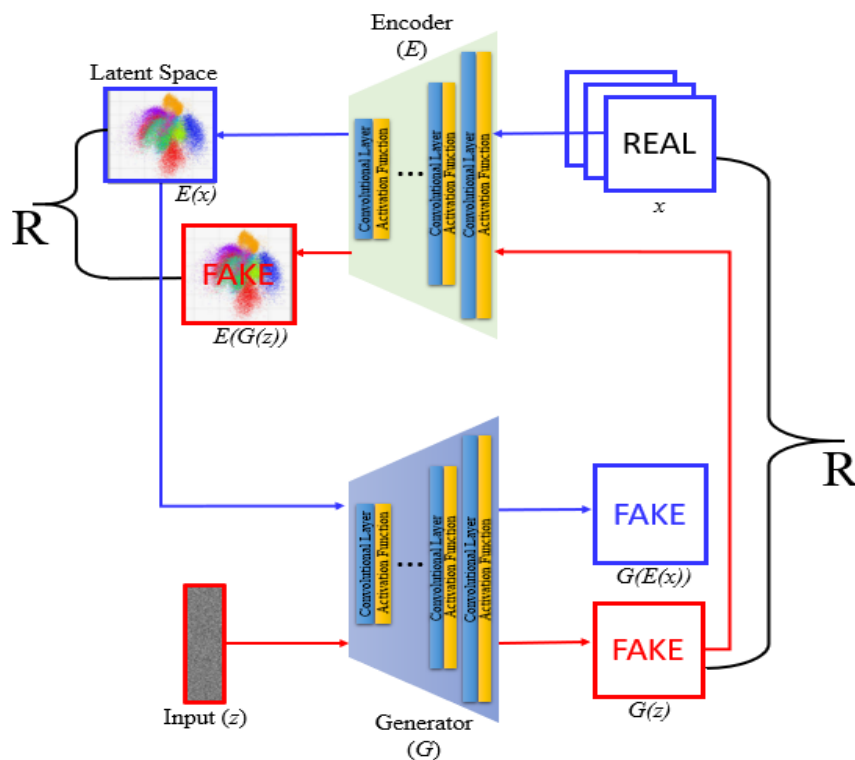


Figure 3.7: The basic structure of an adversarial generator-encoder network (AGE) [123].

### 3.3.4 Attention-based GAN

The attention concept was introduced in 2015 to extend the autoencoder-based GAN structure [124]. For specific contexts as a critical component in generative models, the attention mechanism contains a capability to regionally learn from these important contexts rather than using a latent space. The attention mechanism simulates human vision to learn from image features, which avoids saturation from overloaded information related to an entire view. The uses of attention mechanisms, especially self-attention, have been widespread in deep learning or representation learning.

Self-attention or intra-attention is defined as the attention applied to a single context instead of across multiple contexts [125], which is efficient to capture wider spatial information. In tradition, GANs are applied in image synthetic tasks to capture local spatial information by convolutional neural networks. The receptive fields may not cover enough spatial ranges, which makes GANs have difficulties in learning multi-class datasets and key components merely depending on convolutional neural networks. For example, the synthetic eyes in a human face may be slightly shifted to different positions, and it will make generative results distortive or unrealistic when using convolutional neural networks as the generator and discriminator. To solve this problem, a self-attention mechanism was proposed to ensure the spatial information was captured with a large receptive field. Compared with neural networks and convolutional neural networks, a self-attention mechanism can be used to discover a larger spatial range by computing the response at spatial positions and has led to many state-of-the-art models in various computer vision applications, such as video classification, object detection and so on.

Self-attention GAN, abbreviated as SAGAN, was proposed in 2019, with the self-attention mechanism in the generator and discriminator [126]. The self-attention GAN is beneficial for acquiring global long-range dependencies to synthesise images and has demonstrated great performance in multi-class image generation. By adopting the self-attention mechanism, the generator can draw detailed images with the locations, which obtain clear details for the distant portions of the images. Additionally, the discriminator can accurately enforce complicated geometric constraints. It has been proven by experiments that the adoption of a self-attention mechanism can be advantageous to enlarging feature mapping relationships and improving the generative diversity using GANs.

A progressive attention GAN (PA-GAN) is proposed by Zhenliang He *et al.* in 2020 [127]. The approach uses a progressive structure from high to low feature levels, which constrains the features by using an attention mechanism at each level. With the

proposed attention mechanism, the encoder extracts original images to generate features containing the information of target attributes, and the proposed model uses the attention maps generated with different levels to blend the features into the original images for editing the attributes in a reasonable area. Based on the experimental results, the PA-GAN forces a GAN to learn from more meaningful attributes, and these attributes can be preserved to specifically generate more realistic results with proper data boundaries.

## 3.4 Loss Function Variants of GANs

Loss functions measure predictive accuracy, which can be used to monitor the progress during a GAN training phase [128]. Typical loss functions of GANs measure the similarity or diversity by comparing the generated images and the original ones. Many loss function variants are directly or indirectly designed to estimate the difference between the ground truth and synthetic data [129]. These loss function variants facilitate the selection of hyperparameter optimisation, and a good loss function does not need to use extra networks or additional functions to measure the generative similarity. Various loss function variants of GANs have been proposed for achieving good performance, but many of them are limited to specific scenarios or application purposes. Several important GAN variants based on advanced loss functions are discussed in the following subsections.

### 3.4.1 Wasserstein GAN

The loss function in typical GANs evaluates the similarity between two probability distributions to make sure the generated data  $y$  is close to the real data  $x$ , in which one of the probability distributions is over the fake data ( $\mathbb{P}_g$ ), and the other is over the real data ( $\mathbb{P}_r$ ). In contrast to typical GANs using the discriminator as a binary classifier to identify the difference between two probability distributions, Wasserstein GAN (or WGAN) proposed by Martin Arjovsky *et al.* in 2017 [130] employs the Earth-mover distance, namely Wasserstein distance, to replace the original quality measurement in typical GANs. From the experimental results, Wasserstein GAN successfully improved the optimisation for GAN training.

It is noticeable that the primary difference between the Wasserstein GAN and the typical GAN is the loss function in the discriminator. The discriminator  $D$  of a typical GAN is implemented as a binary classifier. However, in the Wasserstein GAN, a fitting

function based on the Wasserstein distance (or Earth-mover distance) is used, which removes the sigmoid function in the last layer and converts the adversarial learning into a regression task. The Wasserstein distance is formulated as follows:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|] \quad (3.3)$$

where  $W$  is the Wasserstein distance,  $\gamma$  is the moving plan to transport data from  $x$  to  $y$ , and  $\Pi(\mathbb{P}_r, \mathbb{P}_g)$  denotes a set of all joint probability distributions of  $\gamma(x, y)$ , whose data marginals are  $\mathbb{P}_r$  and  $\mathbb{P}_g$  respectively. Compared with Kullback-Leibler (KL) and Jensen-Shannon (JS) divergence, Wasserstein distance reflects the distance even if  $\mathbb{P}_r$  and  $\mathbb{P}_g$  do not overlap. Wasserstein distance has a smooth gradient for training a generator spanning the complete space. However, the *inf* (infimum) in the equation is highly intractable for real computations. Therefore, the Wasserstein distance can be transformed by the Kantorovich-Rubinstein duality [130] and reformulated as follows:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r} [f(x)] - \mathbb{E}_{x \sim \mathbb{P}_g} [f(x)] \quad (3.4)$$

where *sup* (supremum) is the lowest upper bound,  $L$  is 1-Lipschitz functions,  $\mathbb{P}_r$  is the probability distribution over the real data  $x$ ,  $\mathbb{P}_g$  is the probability distribution representing a family of the parameterised density,  $f$  is the Wasserstein metric for transferring real data  $x$  to new data with distribution  $\mathbb{P}_g$ .  $f$  and  $x$  are constrained by  $|f(x_1) - f(x_2)| \leq K|x_1 - x_2|$ , where  $K$  is the Lipschitz constant to function  $f$ . To minimize the Wasserstein distance between  $\mathbb{P}_g$  and  $\mathbb{P}_r$ , the loss function of the discriminator in the Wasserstein GAN is defined as follows:

$$\mathcal{L}_D = \mathbb{E}_{x \sim \mathbb{P}_r} [\log D(x)] + \mathbb{E}_{x \sim \mathbb{P}_g} [\log (1 - D(x))] \quad (3.5)$$

### 3.4.2 Wasserstein GAN with Gradient Penalty

Although Wasserstein GAN has successfully shown a significant improvement in training GANs, it is still difficult to well generalise a deeper model. Due to the problem of vanishing gradients, the loss functions in the Wasserstein GAN easily fail to converge. To deal with the vanishing gradients, Wasserstein GAN with gradient penalty (WGAN-GP) was proposed by Gulrajani *et al.* [131]. WGAN-GP suggests that adding

a gradient penalty term for solving the problem of weight clipping can improve the model performance and training stability. Weight clipping is used to enforce the Lipschitz constraint in calculating the Wasserstein distance, which usually takes a long time to optimise the weight values and easily leads to optimization difficulties when the number of clipping weights is large. WGAN-GP demonstrated its stabilisation in training by using Adam optimiser and convergence faster than typical GANs. Furthermore, WGAN-GP has an outstanding convergence capability to improve training speed, and the quality of generated samples is more robust by pushing the discriminator network to learn smoother decision boundaries. The modified loss function of WGAN-GP is formulated as follows:

$$\begin{aligned} \mathcal{L}_D = & \mathbb{E}_{x_g \sim \mathbb{P}_g} [D(x_g)] - \mathbb{E}_{x_r \sim \mathbb{P}_r} [D(x_r)] \\ & + \lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \end{aligned} \quad (3.6)$$

where  $x_r$  is sample data drawn from the real data distribution  $\mathbb{P}_r$ , and  $x_g$  is sample data drawn from the generated data distribution  $\mathbb{P}_g$ ,  $\mathbb{P}_{\hat{x}}$  is a data distribution uniformly sampled with straight lines between pairs of points, which are sampled from the real data distribution  $\mathbb{P}_r$  and the generated data distribution  $\mathbb{P}_g$ . The first two terms are the original loss in Wasserstein GAN, and the modified gradient penalty is in the last term.

### 3.4.3 Least Square GAN

Least square GAN (LSGAN) was proposed in 2019 [132], which is a new approach to remedy the gradient vanishing problem with a perceptiveness of the decision boundary determined by a discriminator. In the typical GAN, the decision boundary in the discriminator may become very small to update the generator, which may be far from the expected decision boundary. LSGAN uses a least square loss to replace the typical GAN loss of sigmoid cross-entropy. The proposed loss function is as follows:

$$\mathcal{L}_D = \frac{1}{2} \mathbb{E}_{x \sim \mathbb{P}_r} [(D(x) - b)^2] + \frac{1}{2} \mathbb{E}_{z \sim \mathbb{P}_z} [(D(G(z)) - a)^2] \quad (3.7)$$

$$\mathcal{L}_G = \frac{1}{2} \mathbb{E}_{z \sim \mathbb{P}_z} [(D(G(z)) - c)^2] \quad (3.8)$$



where  $a$  is the label of generated samples,  $b$  is the label of real samples, and  $c$  is a value that the generator expects the discriminator to believe in the generated data.  $\mathbb{P}_r$  is the real data distribution, and  $\mathbb{P}_z$  is the generated data distribution started from random noise  $z$ .

LSGAN introduces two benefits: Firstly, the new decision boundary generated by the discriminator can penalise large errors caused by the generated sample far away from the decision boundary. This makes the generated sample move forward to the decision boundary and generates great results in terms of image quality. Secondly, the penalty to the generated samples away from the decision boundary provides sufficient gradient to update the generator and mitigate the gradient vanishing problem in training a deep network.

## 3.5 Challenges in Training GANs

Among learning-based generative models, GANs are not perfect even though they are attractive, applicable and powerful. The two most significant concerns about GANs are that GANs are difficult to train and the synthetic results are hard to be evaluated. On one hand, in terms of GAN training, the main goal is to achieve Nash equilibrium, which is a concept in game theory and represents the state of a game player to achieve the desired outcome without deviating from initial strategies, after considering the choices of game players. The optimal strategies of GANs need to consider the decisions of other opponents, but it is extremely hard to stabilise the training process in practical implementation and promise the most optimal strategy being chosen. Due to the difficulty for both discriminator and generator to reach equilibrium during training, the generator easily fails to learn from a full distribution of real data. On the other hand, regarding performance evaluation, the primary issue is how to measure the performance of generative diversity (or dissimilarity) between real data and generated data. In specific, traditional estimation methods of measuring image accuracy are not suitable to be applied in generative cases with GANs. Therefore, it is still challenging to produce an appropriate evaluation metric to estimate the correspondent distributions between real data and generated data. In the following subsections, the problems associated with the above-mentioned issues are addressed.

### 3.5.1 Mode Collapse

A lack of diversity in generative results is identified as mode collapse, where GANs

capture a single or a few major modes but ignore other minor modes. One of the main problems in GAN development is mode collapse, which is short of diversity in the generated samples when a few modes are concentrated [133]. Therefore, mode collapse is one of the serious problems that need to be watched in training GAN models. A complete collapse is not common, but partial collapses happen very often. Figure 3.8 shows a simple example of mode collapse using the CelebA dataset, where 200 different human faces were input as the training samples, and the shown phenomenon in the right column can be regarded as mode collapse when only a few modes of data are generated compared with the results in the left column. For the generative target of various outcomes, improving training methods or algorithms to prevent mode collapses has become an important research issue.



Figure 3.8: The images generated by GANs using the CelebA dataset. The left column shows more diverse generative results while the right column presents a mode collapse when only a few modes of facial data are generated.

To be specific, mode collapse is a common phenomenon in GAN training. The ultimate objective of a generator is to create realistic images that can fool a discriminator. In the training phase, training data information is detected by a discriminator, constantly updated and sent to the generator. If a generator is trained without receiving updated information from the discriminator, generative results will easily converge into a few modes, which indicates realistic outcomes can be generated from a clear perspective of the discriminator [134]. Hence, concerning the full dynamic

view of training processes, the most effective way should progressively generate images with averagely diverse results among different modes instead of a few precise ones. However, in reality, a generator regularly produces imbalanced modes due to the mode collapse problem, which deteriorates the capabilities of creating various results in terms of generative diversification. Mode collapse makes both the generator and discriminator overfit to exploit a short-term local optimisation rather than global optimisation [135]. Fortunately, it is good news that mode collapses may not be always negative in some synthetic cases. For example, the application of style transfer is beneficial to transfer analogous styles of a few specific modes rather than diverse ones. Therefore, a specialisation of mode collapses can sometimes create results depending on different synthetic requirements.

A solution to mode collapse is to apply sample batches by increasing the diverse assessment during training, and minimising the batch size is one of the techniques to mitigate the mode collapse problem. Another solution is to use multiple generators for acquiring many possible modes, which combines generated samples with different modes.

### **3.5.2 Gradient Vanishing**

Compared to mode collapse happening on generative models only, gradient vanishing is a common problem in machine learning when training models with gradient-based learning techniques. Gradient-based learning is a method to fit parameters by understanding the gradient changes [136]. If the change of parameter values cannot result in a difference in generative results, there may be the problem of gradient vanishing. To train GANs optimally, both the generator and discriminator have to produce meaningful outcomes and valuable feedback determined by loss functions. Gradient vanishing may happen when a well-trained discriminator squashes the loss functions to a minimal value, which makes the gradients approximately close to zero and delivers a very small amount of feedback to the generator [137]. Consequently, gradient vanishing will make a generator completely stop the progress during training.

Two common situations halt a generator from progress: Firstly, since the GAN training is dynamic, the gradient among parameters descends to an optimal value, and the training has reached a dynamic balance between the generator and discriminator [138]. It is a training phenomenon that the dynamic balance has been achieved and makes the generator stop updating. The second situation is the gradient vanishing problem [139]. Because of the gradient vanishing problem, the generator may fail to improve on producing good-quality images, but generative results are not of good

quality. In contrast, the discriminator no longer accepts the generated sample and reduces the learning capacities. The gradient vanishing problem coupled with the over-confidence makes the GAN training challenging because the discriminator does not forward meaningful information to the generator anymore. What is worse, the generator might receive wrong feedback to mislead the generative features with poor or inaccurate outcomes because of the gradient vanishing. To deal with the problem of gradient vanishing, over-training should be avoided in the discriminator, and the improvement between the discriminator and generator needs to be carefully detected.

### 3.5.3 Non-convergence

Convergence is a mathematical term commonly used in series or sequences studies, and non-convergence is a universal problem for machine learning, especially training with small datasets. To use loss functions to optimise the free parameters of a neural network, iterations are necessary to minimise the loss values by updating the weights, and back-propagation is designed to find an arbitrary point defined by the loss functions. However, non-convergence occurs if a strictly converged point fails to be found, and the loss value will vary within a smaller range. In training GANs, the feasibility of convergence can also be explained as the desirability of equilibrium, and a converged situation is to find a balance between the discriminator and generator [140]. Since generators try to synthesise the best images for fooling discriminators, generators could keep progressing to meet the requirements of discriminators. The GAN training falls into a permanent cycle like the eternal cat-and-mouse game, in which the difficulty of finding a training balance will become one of the primary reasons causing convergence failure, namely the problem of non-convergence [141]. What is worse, even though a training balance has been presented during training, it is still very difficult to reach the global Nash equilibrium when the GAN training is frequently under oscillation or cyclical phenomenon, which is prone to converge to a local Nash equilibrium instead of the global one.

A possible solution to the non-convergence problem is to set an appropriate batch value in a training phase. The real image features are computed by minimal batches which fluctuate with every training cycle. If the batches are introduced as randomness, it could make it difficult for a discriminator to overfit the input data [142]. A dynamic feature match can be helpful to find a balanced situation between generated data and real data. Consequently, a suitable setting of batch values can not only improve a training balance between the generator and discriminator to prevent the non-convergence problem but also maintain a static ratio of iterations to mitigate other

training problems, such as gradient vanishing.

### 3.5.4 Hyperparameter Optimisation

Hyperparameters are some learning parameters needing to be controlled in a learning process, which can be regarded as multi-objective optimisation with multiple hyperparameters related to objectives simultaneously optimised for training a deep network [143]. Apart from the serious problems mentioned in previous sections, hyperparameter optimising is a significant issue related to all the above problems in GAN training [144]. For instance, hyperparameter selection has an impact on convergence caused by the gradient vanishing problem, and mode collapse is also led by over-optimisation with inappropriate parametric values.

However, hyperparameter optimisation is very time-consuming and needs a lot of patience for good training strategies. From the perspective of discovering good hyperparameters, training a GAN model based on weights and biases must minimise either training errors or model complexity to robustly trace the changing of generative results. Due to the used loss functions that might conflict with each other, the performance tradeoff between the generator and discriminator is difficult to achieve because a large number of variables need to be fine-tuned to achieve stable performance, especially when multiple loss functions are set up to understand the correlations between the hyperparameters and final performance [145]. Therefore, the hyperparameters for the optimal solution are hard to be discovered in GAN training. Additionally, the training in both generator and discriminator is a dynamic process that is generally more complex and unstable than traditional deep network training.

It is impossible for GAN training to perform well without good hyperparameters. Several advanced techniques for optimisation have been developed to potentially trace the correlations between the chosen hyperparameters and final performance. First of all, stochastic gradient descent (SGD) [146] is an important iterative optimising method, which uses a gradient descent procedure to produce the expectation of the gradient, and SGD will be more efficient than gradient descent. Secondly, the Adam optimiser [147] is an optimisation of a first-order and gradient-based stochastic algorithm with an adaptive learning rate. It has been proven robust and suitable for non-convex optimisation. Thirdly, batch normalisation [148] is a technique to reduce the internal covariate shift in a deep neural network. Each batch is used to estimate the mean and variance by training iterations. The advantages of batch normalisation are employing a higher learning rate, paying less attention to the initialisation and reducing the requirement of dropout. Finally, regularisation techniques [149] are efficient to mitigate

the optimisation problems. There are numerous regularisation methods proposed to stabilise the GAN training, such as regularisation at output layers, regularisation with the modified loss function, weight penalty, gradient penalty and so on.

## **3.6 Evaluation Metrics**

GAN models have been adopted in many different applications, and each application contains specific evaluation metrics with different requirements. Universally evaluating the performance of GAN models is extremely challenging, and it is still an open question of how to select an appropriate evaluation metric among various GAN variants. There are still no universal quantitative evaluation metrics that can objectively and comprehensively access the GAN performance.

There are two major problems in evaluating GANs. Firstly, model collapse and generative diversity may not be objectively detected and evaluated. Secondly, quantitative evaluation methods based on probability or likelihood scores may not correspond to human perceptions, which is the main reason why human perception is currently a reliable method to evaluate the quality of generated samples. Consequently, determining appropriate metrics as well as objective evaluation methods among various GAN applications is still a challenging task.

Several popular approaches to accessing the performance of generative results are discussed in the following subsections. In terms of appropriately evaluating the GAN performance, these methods attempt to seek improvements to deal with the evaluation difficulties. The listed approaches are commonly used as quantitative evaluation metrics in many GAN studies.

### **3.6.1 Likelihood Estimation**

Likelihood, related to similarity, uses a statistical approach to describe observed data with a joint probability value [150]. There is an assumption of the likelihood that the generative samples follow the Gaussian distribution of true data, and the generative data can ideally match the true data distribution. In the synthetic process, an easy metric should be used to measure the likelihood between the generated data and the true data. For example, KL divergence is a common measurement to calculate the difference between two probability distributions in machine learning, and an easy way to discover the maximum likelihood between the data distributions of the generated and true samples is to optimize the parameters by minimising the KL divergence value using

gradient descent. The log-likelihood, KL divergence and cross-entropy are commonly used to evaluate the data likelihood in GAN training. An overall likelihood is the sum of the evaluation values of discriminators and generators in a generative event.

Likelihood estimation has disadvantages: Methods of likelihood estimation are generally applied at a low dimension. Furthermore, having a high score of likelihood estimation may not truly reflect the synthetic quality of samples generated by a GAN model.

### 3.6.2 Inception Scores

Inception scores are widely used to evaluate the GAN performance and were proposed by Salimans *et al.* in 2016 [151]. Inception scores are based on image classification with a pre-trained model to classify the generated images. The inception scores combine both the confidence of class predictions to evaluate the generative quality and the integral of marginal probability predictions to evaluate the generative diversity. The development of the inception scores attempts to replace human perception with a quantitative method, which is correlated to subjective evaluations among classes. Specifically, the probability of generated images belonging to each class is predicted, and then these prediction values are summarised as inception scores.

Inception scores rely on two desired properties of conditional label distribution and marginal label distribution. The conditional label distribution is calculated by fitting generated data into an inception model, which should contain low entropy in terms of fidelity. Oppositely, the marginal label distribution has to reach a high entropy for the diversity of generated data. Based on the Inception V3 network, the inception score ( $IS$ ) of a generator  $G$  is formulated as follows:

$$IS(G) = \exp (\mathbb{E}_{\mathbb{P}_g} [D_{KL}(p(y|x)||p(y))]) \quad (3.9)$$

where  $\mathbb{P}_g$  is the generated data distribution,  $\mathbb{E}$  indicates the expectation value,  $x$  is the generated image  $x$  as inputting to Inception V3,  $y$  is the output label,  $p(y|x)$  is conditional label distribution,  $p(y)$  is marginal label distribution, and  $D_{KL}(p(y|x)||p(y))$  is KL divergence between conditional label distribution  $p(y|x)$  and marginal label distribution  $p(y)$ .

In comparison to the disadvantage of likelihood estimation, which cannot correctly reflect the quality of diverse synthetic data, a high inception score indicates that the generative model can create high-quality samples even if the samples are diverse or dissimilar to the original data. However, the adoption of inception scores also remains

serious restrictions: Firstly, if generative models fall into mode collapse, the performance based on inception scores may be excellent, but the generated samples are still in a low-quality situation. Secondly, inception scores are sensitive to general classes based on the pre-trained models and fail to evaluate a specific label in datasets. Therefore, inception scores need a large number of training samples to get reliable results, such as an Inception V3 network trained with ImageNet. In addition, a simple way to calculate inception scores is to use a trained model, but it may not truly reflect human perception except for the classes in the trained model. Finally, inception scores do not compare synthetic samples with real data.

### 3.6.3 Fréchet Inception Distance

The Fréchet inception distance (FID) was proposed by Heusel *et al.* in 2017 to detect the intra-class mode dropping [152]. In the FID approach, generated samples are embedded into a feature space using a pre-trained network, and the FID values are calculated based on the means and covariances of the feature vectors obtained from the pre-trained network. Rather than directly comparing images pixel by pixel, FID is based on the assumption that generated samples follow a multi-dimensional Gaussian distribution and the distance value is calculated based on the means and covariances of the two Gaussian distributions of features of the generated images and real images.

The core of FID is the distance between the distributions of the synthetic data and real data. By evaluating the data distributions, it is possible to measure the similarity between two probability distributions. A smaller FID value represents more similarity between the distributions of the two data groups. The FID can be formulated as follows:

$$FID(\mathbb{P}_r, \mathbb{P}_g) = \|\mu_r - \mu_g\|^2 + Tr\left(\sum_r + \sum_g - 2\left(\sum_r \sum_g\right)^{\frac{1}{2}}\right) \quad (3.10)$$

where  $\mu_r$  and  $\mu_g$  are the feature-wise mean of the real and generated images respectively,  $\sum_r$  and  $\sum_g$  are the covariance matrix of the real and generated feature vectors,  $Tr$  indicates the trace linear algebra operation,  $\mathbb{P}_g$  is the generated data distribution, and  $\mathbb{P}_r$  is the real data distribution.

Compared to inception scores, FID is more powerful for handling data disturbances, and FID can detect intra-class mode dropping.



### 3.6.4 Kernel Inception Distance

The major problem of FID is highly biased to small image datasets, so the sample size has to be large enough to obtain reliable FID values. To mitigate the problems of FID, kernel inception distance (KID) was proposed by Bińkowski *et al.*, which measures the squared maximum mean discrepancy (MMD) between inception features with polynomial kernels [153]. Compared to FID, KID has the following advantages: Firstly, for the data distribution in activated functions, KID does not assume a parametric form of the activation. Secondly, with the use of cubic kernels, KID additionally compares the values of skewness, mean and variance. Finally, in contrast to FID, KID with the polynomial kernels is a more unbiased estimator.

Similar to FID, KID computes the squared maximum mean discrepancy between the features from a pre-trained Inception Network with the real and generated images as inputs respectively. Although KID is strongly correlated to FID, KID can produce unbiased estimates to make the values fairly and truly reflect the difference between generated data and real data along with more inception channels. A lower KID value indicates more visual similarity between real and generated images, and the KID is formulated as follows:

$$\begin{aligned}
 KID(\mathbb{P}_r, \mathbb{P}_g) = & \mathbb{E}_{x_r, x'_r \sim \mathbb{P}_r} [k(x_r, x'_r)] + \mathbb{E}_{x_g, x'_g \sim \mathbb{P}_g} [k(x_g, x'_g)] \\
 & - 2\mathbb{E}_{x_r \sim \mathbb{P}_r, x_g \sim \mathbb{P}_g} [k(x_r, x_g)]
 \end{aligned} \tag{3.11}$$

where  $k$  denotes a polynomial kernel function,  $k(x, x') = (\frac{1}{d}x^T x' + 1)^3$ ,  $d$  is the representation dimension,  $\mathbb{P}_g$  is the generated data distribution, and  $\mathbb{P}_r$  is the real data distribution.

### 3.6.5 Classification Accuracy as an Evaluation Metric

A common problem for inception score, FID and KID is that these three metrics have a heavy reliance on pre-trained models, which do not consider a dataset containing different classes from the datasets for the pre-trained model [154]. As a result, inception score, FID and KID may not correctly or truly capture the class properties, and they will not appropriately reflect the real quality of generative samples. It is reasonable to assume that if a generator has captured the distributions of the real data, the difference

between the real and fake data should be small. Therefore, classifiers, apart from pre-trained models, can be trained by real data to evaluate the quality of fake data generated by GANs. In terms of the performance evaluation of GANs, various classifiers could be directly used, and the classification accuracy of the fake data can be used to evaluate the generative quality.

## 3.7 Applications of GANs in Image Synthesis

The most direct application for a generative model is to create new data. GAN, as one of the generative models, can efficiently learn from the data distributions of real images and generate new images following the distributions of real data with desired diversity. The applications of GANs in image synthesis are reviewed in this section. At present, there are many popular applications of GANs for computer vision, *i.e.*, image synthesis, image transformation and video generation, and many GAN variants were proposed for solving different synthetic problems. It needs to be noted here that these remarkable applications strongly rely on a large number of training samples, and several thriving image synthesis applications with GANs, including image super-resolution, image repairing, face synthesis, image translation and video synthesis, are presented in the following subsections.

### 3.7.1 Image Super-resolution

A super-resolution generative adversarial network (SRGAN) was proposed in 2017 [155]. Low-resolution images are taken as the inputs, and SRGAN generates high-resolution images as the outputs, with four times up-scaling in image resolutions. However, SRGAN has a serious problem for real applications, which is that the generative textures are blurry and not clear enough compared to real images. Moreover, the synthetic results are always accompanied by noises or distortions. To deal with this problem, an enhanced super-resolution generative adversarial network (ESRGAN) was proposed to improve the performance [156]. Compared to SRGAN, ESRGAN improved the structure of the network, adversarial loss, and perceptual loss. According to experimental results, the images generated by ESRGAN are of higher quality than those by SRGAN.

### 3.7.2 Image Repairing

Image repairing is a common application for image processing. The operations of image repairing generally need to find out missing parts and replace the marked regions with synthetic contents. GANs perform well for repairing backgrounds if the missing parts are very similar to the backgrounds. On the other side, if the important parts are lost rather than backgrounds, such as repairing the missing parts of a human face, the images should be divided into critical patterns (*e.g.*, eyes, mouth, eyebrows, nose, *etc.*) and the lost features can be filled with corresponding objects. Li *et al.* [157] proposed a structure using an autoencoder and a GAN coupled with two adversarial losses for image repairing. Vitoria *et al.* in 2019 [158] proposed an improved version using Wasserstein GAN to complete the missing regions of images. Dharmo *et al.* [159] adopted convolutional neural networks and GANs to generate the scene background by removing the object in the image foreground and using methods of background subtraction to detect motions. Although the above GAN-based methods have provided great experimental results in image repairing, the processing time and cost for computational efficiency are still serious issues to be considered, compared to traditional repairing methods.

### 3.7.3 Face Synthesis

Face synthesis is a popular area in image synthesis and has been an important direction for GAN applications. Many researchers make efforts to generate photorealistic face images using GANs. The face images generated by many GAN models can well retain identical features and produce a large number of look-like fake faces. GANs can efficiently recognise facial attributes to generate high-quality as well as high-resolution synthetic results. To generate identical facial features with GAN frameworks, many strategies and structures are employed in face synthesis, and several remarkable GAN models were proposed. A two-pathway generative adversarial network (TP-GAN) was presented to generate high-quality front face images from a single face image [160], Dual-agent GANs (DA-GAN) tried to synthesise profile faces [161], and CR-GAN manipulated multi-view facial generation [162]. A style-based generator architecture for GANs (StyleGAN) was proposed in 2019 [163], in which high-level facial attributes are generated and can be controlled by the intuitive mixing or interpolation operation. In StyleGAN, a new latent space free from the restriction that latent space always follows the probability density of the training data was newly

introduced to improve the weaknesses in face synthesis.

In addition, other notable works adopted marks, symbols, segmentations, or other reference information to synthesise photorealistic faces from random seeds. GP-GAN, for instance, attempted to generate samples by landmark-guided samples [164]. However, these methods heavily depend on reference features based on very narrow and specific facial representations, which generally degrades the performance in facial generation tasks. The StarGAN [165] overcame the drawbacks of requiring reference features to edit facial attributes, and only employed labelled data with adversarial loss, attribute classification loss and reconstruction loss to successfully modify facial attributes without using any additional reference feature information.

### **3.7.4 Image Translation**

Image translation converts image contents from one data domain to another, and the main objective is to learn the mapping relationships between output images and input images. Many image-to-image translation approaches were proposed and achieved remarkable performance. The experimental results showed that both pix2pix [166] and pix2pixHD [167] were effective for most graphic as well as visual applications with a pixel-level image translation. Although pix2pix and pix2pixHD can be used to solve primary image translation problems, it still needs corresponding features (or paired images) as the required training data. However, paired images with corresponding features are very difficult to be collected in real applications. Different from pix2pix and pix2pixHD, CycleGAN [168] adopted the concept of cycle consistency to achieve two domain translations, which enforces a mapping from one domain to another domain that is roughly the same in each direction and does not need paired data to learn the mapping information between images. Other GAN models, such as DiscoGAN [169] and DualGAN [170], were also proposed to solve the similar issue of training on unpaired data. In addition, to deal with the translation problem among multiple domains instead of two domains, StarGAN was proposed by Choi *et al.* in 2018, which can translate images among multi-domains with one single GAN model [165]. The details of paired image-to-image translation and unpaired image-to-image translation will be further described in the following subsections.

#### **3.7.4.1 Paired Image-to-image Translation**

Image-to-image translation is a typical image translation method, which learns a mapping relationship to synthesise the desired images from conditional inputs and

random noise. Pix2pix was proposed by Isola *et al.* in 2018 [167], which is based on structures of both convolution-based GANs and condition-based GANs. Pix2pix is a GAN model with one-to-one image migration and has demonstrated efficiencies in various image-to-image translation tasks. Pix2pix adopts an encoder-decoder structure in the generator, which is comprised of convolutional layers. Figure 3.9 shows the basic structure of pix2pix, where the conditional image and the real image are composed of a set of paired data.

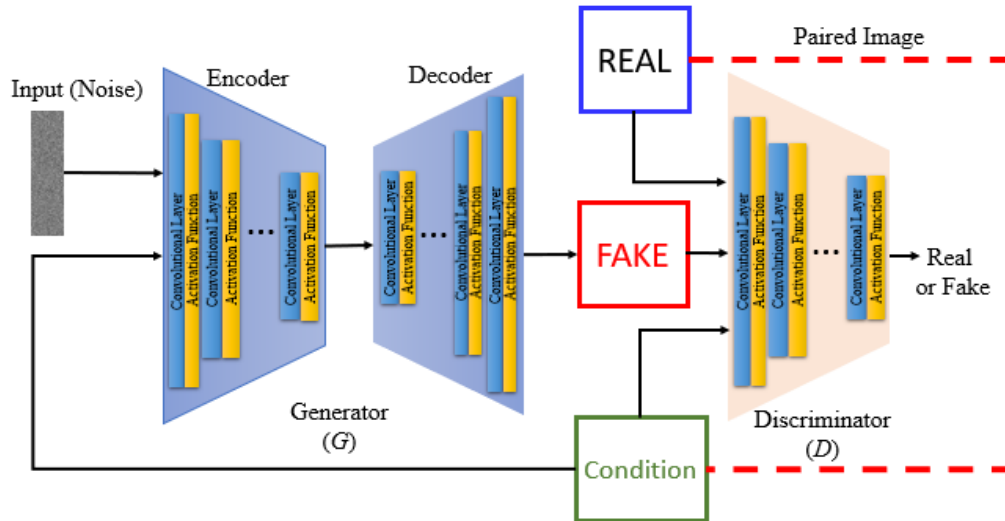


Figure 3.9: The basic structure of the pix2pix model [167].

Pix2pix model has three important advantages: 1) Pix2pix is a general generative model to solve the real problem of image translation pixel by pixel, which is usually suitable for almost all one-to-one synthetic cases. 2) Pix2pix provides specific loss functions, and the networks learn the mapping relationships between the conditions and generative results with specific loss functions. 3) Pix2pix takes the advantage of shared information between the encoder and decoder as the network framework for acquiring high-quality mapping results. However, its drawback is still evident that pix2pix requires a large amount of labelled data with corresponding features, which are generally not available or difficult to be collected in practical implementations.

In terms of pix2pix training, since the generated images are desired to be close to ground truth, an additional content loss, besides the adversarial loss, is added to the objective function, which measures the  $L_1$  distance between the output images and the ground truth images. The overall loss functions in pix2pix are shown as follows:

$$\begin{aligned} \mathcal{L}_{cGAN}(G, D) = & \mathbb{E}_{x,y}[\log D(x, y)] \\ & + \mathbb{E}_{x,z} \left[ \log \left( 1 - D(x, G(x, z)) \right) \right] \end{aligned} \quad (3.12)$$

$$\mathcal{L}_{L_1}(G) = \mathbb{E}_{x,y,z} [\|y - G(x, z)\|_1] \quad (3.13)$$

$$\mathcal{L}_{all} = \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L_1}(G) \quad (3.14)$$

where  $x$  is the observed image,  $z$  is the random noise vector, and  $y$  is a mapping result to the observed image of  $x$ . The discriminator  $D$  and generator  $G$  are trained by the content loss to produce the mapping results from  $x$  to  $y$ , which is hard to be distinguished by the adversarial loss.

It can be emphasised that, before the presence of pix2pix, most researchers used the mean square loss of  $L_2$  to train transformation networks and autoencoder-based GANs, which have been proven unable to transform images with clear results between two domains. In pix2pix, a loss function with the conventional  $L_1$  loss was newly used for training an encoder-decoder network. The adoption of the  $L_1$  loss function can be treated as a benchmark for image-to-image translation methods based on condition-based GAN structures. Consequently, pix2pix enlarges the applications of image-to-image transformation by mixing the structure of convolution-based GANs and condition-based GANs.

### 3.7.4.2 Unpaired Image-to-image Translation

A large number of paired images may not be available in many applications. A novel GAN structure, known as CycleGAN, was proposed by Jun-Yan Zhu *et al.* in 2017 [168] for unpaired image-to-image translation. A cycle consistency is adopted as the primary approach of CycleGAN to learn the mapping relationships between two different domains. Before the appearance of CycleGAN, most GAN models are based on supervised methods and highly rely on paired images for image-to-image translation. The structure of CycleGAN makes a great improvement by training with unpaired data, which significantly enlarges potential translation applications without using paired images. The basic structure of CycleGAN is shown in Figure 3.10: .

CycleGAN typically consists of two generators and two discriminators. The cycle consistency loss is added to make the generated images roughly the same with a translation between two unpaired data domains. In CycleGAN, the generator structure is similar to the autoencoder structure, which contains two main components, *i.e.*, encoder and decoder. The basic principle of CycleGAN focuses on domain data

adaption and tries to identify the data distribution learned from labelled images between two different domains.

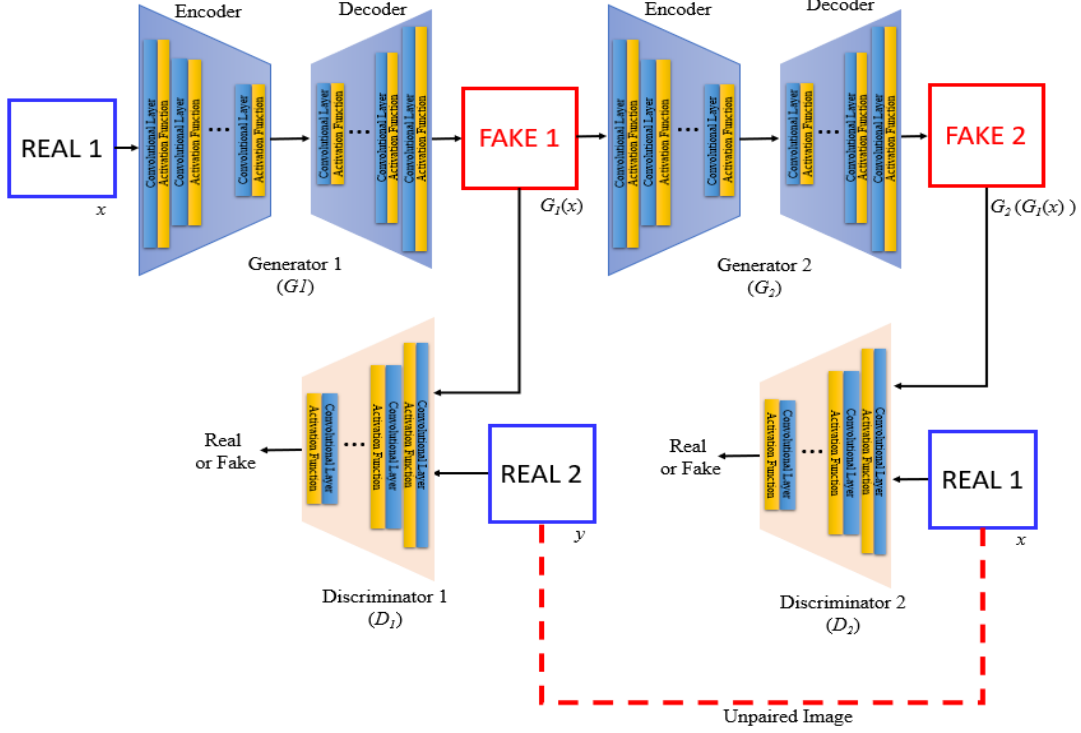


Figure 3.10: The basic structure of the CycleGAN [168].

CycleGAN discovers the marginal matching relationships, which map the outputs to match the empirical distribution between two domains. Two generators are trained to fool two discriminators and then enforce the marginal matching over the target domain and source domain separately. The learning objectives in the form of loss functions are formulated as follows to minimise the loss function with respective generators:

$$\begin{aligned} \mathcal{L}_{GAN}(G_1, D_1, X, Y) &= \mathbb{E}_{y \sim P_{data}(y)} [\log D_1(Y)] + \mathbb{E}_{x \sim P_{data}(x)} [\log (1 - D_1(G_1(X)))] \end{aligned} \quad (3.15)$$

where  $G_I$  is the generator,  $G_I(X)$  is the generated images from real data in domain  $X$ , and  $D_I$  is the discriminator aiming to distinguish generated images  $G_I(x)$  and real images in domain  $Y$ . The generator tries to minimise the training objective whilst the discriminator aims to maximise it.

On the other hand, the cycle consistency enforces the transformed results and

reconstructive results remain close to the original images. In image translation cases, the similarity is typically measured by  $L_1$  or  $L_2$  normalisation. The cycle consistency with  $L_1$  can be formulated as:

$$\mathcal{L}_{cyc}(G_1, G_2) = \mathbb{E}_{x \sim \mathbb{P}_{data}(x)} \|G_2(G_1(x)) - x\|_1 \quad (3.16)$$

where  $x$  is the image in domain  $X$ ,  $G_1$  indicates the generator transferring images from domain  $X$  to domain  $Y$ ,  $G_2(G_1(x))$  is the image reconstruction, which is expected to be similar to the original image  $x$ .

### 3.7.5 Video Synthesis

Inspired by the success of GANs in image synthesis, researchers have extended GAN applications to video synthesis. Compared to image synthesis, video synthesis based on GAN structures requires faster computing facilities and larger memory to deal with video frames, especially in real-time video synthesis tasks. The generative adversarial network for video (VGAN) was proposed by Vondrick *et al.* [171], which combines a static background and a moving foreground video. The generator needs to process two different data streams: the background stream generated with 2D convolutional layers and the foreground stream generated as a 3D foreground with spatial-temporal 3D convolutional layers. Since VGAN manipulates videos as 3D objects, it requires powerful hardware to process video data. In addition, MoCoGAN was proposed by Tulyakov *et al.* for video generation [172], which decomposes videos into motion and content vectors respectively and employs a recurrent network to map a sequence of motion vectors and content vectors. Two discriminators are used in MoCoGAN, with one discriminator distinguishing the real from fake frames whilst the other distinguishing fake videos.

Although recent works attempt to extend the remarkable performance of GANs, generating realistic videos remains a significant challenge in practical applications. Owing to the limitations of hardware and training stability, high-quality videos are still more difficult to be generated than images. Reviewing the current development of video generation with GANs, it is clear that video synthesis techniques can be improved from the following perspectives: Firstly, it is expected to produce more high-resolution videos. Secondly, the quality of generated videos can be promoted along with the frame number increased. Finally, more realistic results are expected to reduce the blurs of generated video content.



## 3.8 Conclusion

A literature review for image synthesis based on GANs is presented in this chapter. The main topics in this chapter include the GAN theory, structure variants, loss variants, training difficulties, evaluation matrices and applications. Firstly, the basic principles of GANs commonly used for image generation are introduced. The derived structure and loss variants are discussed in the first part. The challenges in training GANs are reviewed with a focus on how to find a balance between generators and discriminators for generating high-quality images. For performance evaluation, generally used matrices are presented to assess the synthetic performance of GANs. Finally, the applications of GANs in image synthesis are reviewed.

## Chapter 4

# Small Training Data Augmentation Using GANs Based on One-to-many Image Mapping for Enhancing the Performance of Image Classification

### 4.1 Introduction

Computer vision approaches are used to analyse visual data, such as images and videos, for making decisions or predicting results. Recently, the field of deep learning has rapidly grown due to the enhancement of computational capacity, and visual data can be well recognised by analysing the feature and contextual information in computer vision applications. Convolutional neural networks (CNNs) can automatically extract features from the given training images, which significantly improves the accuracy of image classification [173]. However, traditional deep learning methods require a large number of labelled samples for the CNNs to learn sufficient features to prevent the problem of overfitting [174]. To deal with this problem, many regularisation methods have been proposed for the structure of CNNs. On the other hand, increasing data diversity has been proven to effectively overcome overfitting by using data augmentation methods (*e.g.*, by traditional image transformation, adding noise, *etc.*) [175]. Generally, a good learning approach in computer vision should involve diversified information that can be flexibly adapted to various real environments. Data diversity ensures the training data contain more discriminative information and enforce models to learn from the complement information. In other words, to achieve robust and reliable results for image classification tasks, it is necessary to collect as many representative sample images as possible. Consequently, having a diversity of training data for a deep neural network can prevent the problem of overfitting and make the training results more robust.

However, in some application areas, obtaining sufficient labelled data requires a great deal of expertise and time [176]. Limited training data would exacerbate the performance of classification, due to lacking the diverse data to learn the various representations. Therefore, training with sufficient diversified training data is a solution to overcome the overfitting problem in image classification. Data augmentation

methods have been proven capable to increase the diversification from a limited number of image data [177]. The commonly used traditional techniques of image data augmentation include geometric transformation and photometric transformations, such as reflection, rotation, translation, scaling, cropping, noise adding, kernel filtering and so on. They all aim to enlarge the variations of the existing images as diverse data so that the neural networks can learn from the augmented differences to increase the diversification of original datasets.

Taking an autopilot system or gesture recognition as an example, if one single ideal pattern is fed into a deep network and expected to recognise all the similar traffic signs or gestures in a deep learning system, the network is impossible to attain the desired performance due to the problem of training data scarcity. Deep models cannot comprehensively learn the realistic representations from a small but perfect dataset because the learning effectiveness needs to be promoted by a large number of diverse training samples to achieve the expected performance. Figure 4.1 shows a huge expectative gap that training with a small number of perfect samples cannot achieve expected performance in deep models, and a large number of diverse samples of images are essential as good training data. In real cases, data diversity and data amount have become serious training problems to bring deep learning to real-world applications.

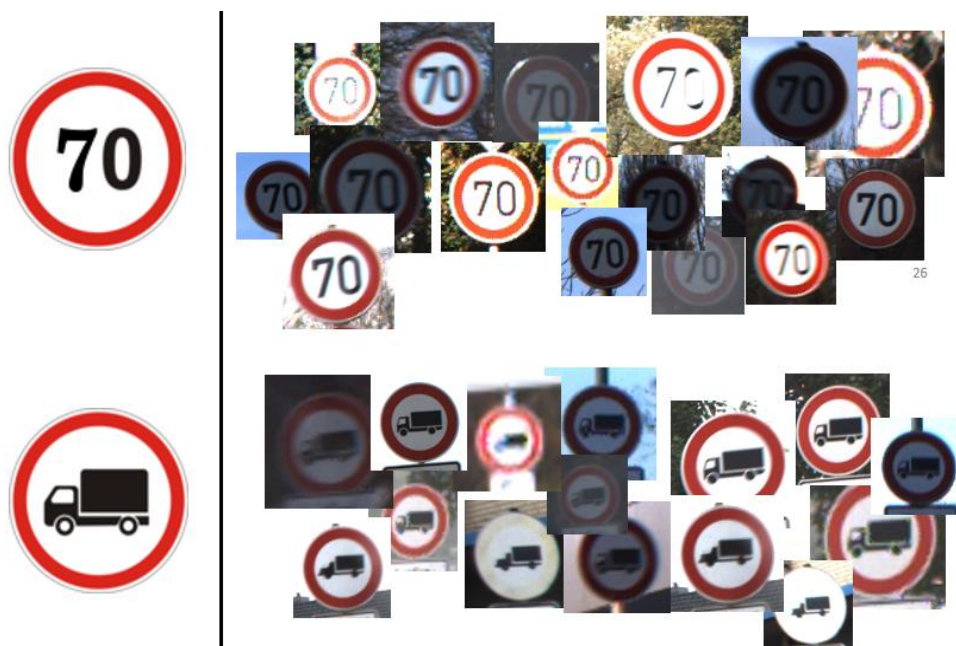


Figure 4.1: Comparison of perfect training samples in human vision (left column) and good training samples in deep learning models (right column).

In terms of one-to-many image mapping, a novel unconditional GAN model is proposed in this chapter for applications where one single image is typically impossible

to carry sufficient features for training a deep model to learn representations comprehensively. The proposed approach aims to transfer images from a single sample and create diverse augmented images that are suitable to train a deep network. The proposed method is based on an unsupervised deep generative model using a one-to-many mapping method. In terms of data augmentation, the images generated by the proposed model are designed to increase the diversity and amount from a small dataset, which can be used to promote image classification performance when a large number of augmented samples are involved to train a convolutional neural network.

A convolution-based GAN is adopted as the fundamental structure of the proposed model to produce fake images, which can enlarge the data amount and data diversity. Additionally, a proposed perturbation mechanism is newly introduced with the one-to-many image mapping method to generate images from a single sample. A transformation matrix  $M$  can be conducted to normalise the original image; another transformation matrix  $M'$  is used to generate extended images to simulate the results of good training samples. In the proposed perturbation mechanism, matrix  $M$  is responsible for image quality, and matrix  $M'$  controls the diversity of the generative results. With an appropriate design of these two transformation matrices of  $M$  and  $M'$ , the proposed GAN framework can synthesise good-quality and diverse images from one training image only. In contrast to one-shot learning adding one sample to each learned class, the proposed model learns from one image without any pre-training requirements. These augmented images that are class-informative are similar to the original training samples and can be used as good training data in deep learning models. The details of the proposed model will be presented in Section 4.2.

For evaluating the performance of data augmentation, image classification experiments were applied as an evaluation metric to analyse the effectiveness of the data augmented by the proposed GAN framework. Four common CNNs, including AlexNet, GoogLeNet, VGGNet and ResNet, were involved in the evaluation experiments. Notably, two datasets of MNIST [178], [179] and RPS [180] were conducted, and the reduced training samples, from 1 to 20 images per class, were used as the small training dataset, but a large amount of validation data has remained for reliable validation results. In addition, in the statistical evaluation, the validation accuracies with augmented training data were statistically employed to compare those without augmentation with a student's t-test method. Furthermore, the confusion matrix [181] and t-distributed stochastic neighbour embedding (t-SNE) [182] were applied to further evaluate the classification effectiveness of augmented data generated using the proposed GAN model. From the experiments, results show that the proposed GAN model used to generate additional images as the augmented training data can significantly improve the accuracies of image classification among the CNNs. The

contributions in this chapter are as follows:

- A novel GAN framework based on one-to-many image mapping is proposed to synthesise many realistic images with desirable diversity from one original image only, which overcomes the drawback of traditional GANs that need plenty of original images to generate high-quality images.
- A perturbation mechanism based on two transformation matrices is proposed to balance the quality and diversity of the images generated by GANs using a single original image only and thus provide an effective approach for image training data augmentation, as demonstrated by the experimental results in Section 4.4.

## 4.2 Methods

In this section, a GAN framework is proposed to generate images of both high quality and good diversity from a small number of real images, even with just one single perfect sample. Based on the proposed method, the straightforward approach to create diverse results from a single image is to intentionally adopt the perturbative information into input instances for diversified results, which can not only fool the GAN model to generate various possible predictions but also prevent the happens of overfitting.

Based on adversarial learning with the perturbation mechanism, the well-defined parameters in the proposed GAN model serve as an alternative way to dynamically generate various possible samples. Contrasted to typical GANs, the proposed perturbation mechanism is located in the input of the discriminator and controls the inputting instances, which has the advantage of independently manipulating the inputs in the discriminator. For instance, during a training phase, the discriminator learns the feature information from a single image and forwards the gradients to a generator for image synthesis. If the overfitting problem and gradient vanishing happen, the generated results will not be improved anymore through the training process, which indicates the discriminator is over-optimised to the input data without passing updated gradients to the generator in a training phase. By using the perturbation mechanism, the fitted parameters in the discriminator can be re-updated by the renewed input instances. In short, the perturbations mechanism rules the discriminator to learn renewed perturbative features and mitigate the drawbacks caused by training with a small number of images, such as gradient vanishing, overfitting and mode collapse.

## 4.2.1 Network Framework

Figure 4.2 shows the architecture of the proposed method illustrating an overall view of the proposed GAN framework. Inspired by the deep convolutional GAN (DCGAN) proposed by Radford *et al.* [116], convolutional neural networks usually have a strong visual fidelity and spatial localisation of image objects in the input instances. The proposed GAN model adopted the architecture of convolution-based GANs as the basic structure because it is composed of a simple and foundational framework to create high-quality synthetic images from an original image dataset with reasonable training time [183], [184]. Moreover, compared to typical GANs and other structural variants of GANs, convolution-based GANs can be used to create diverse images by appropriate hyperparameters settings between the discriminator and generator with an unsupervised learning method [185].

To solve the problems led by giving a single training image or a very small number of images as the training dataset, a perturbation mechanism with two transformation matrices is conducted to update the input instances in front of the discriminator when training the proposed GAN model. The perturbation mechanism is introduced to shift the input images into diverse appearances, and the generated images should be recognisable but diverse from the original training samples for data augmentation requirements. The detail of the perturbation mechanism, model building and training process will be discussed in the following sections.

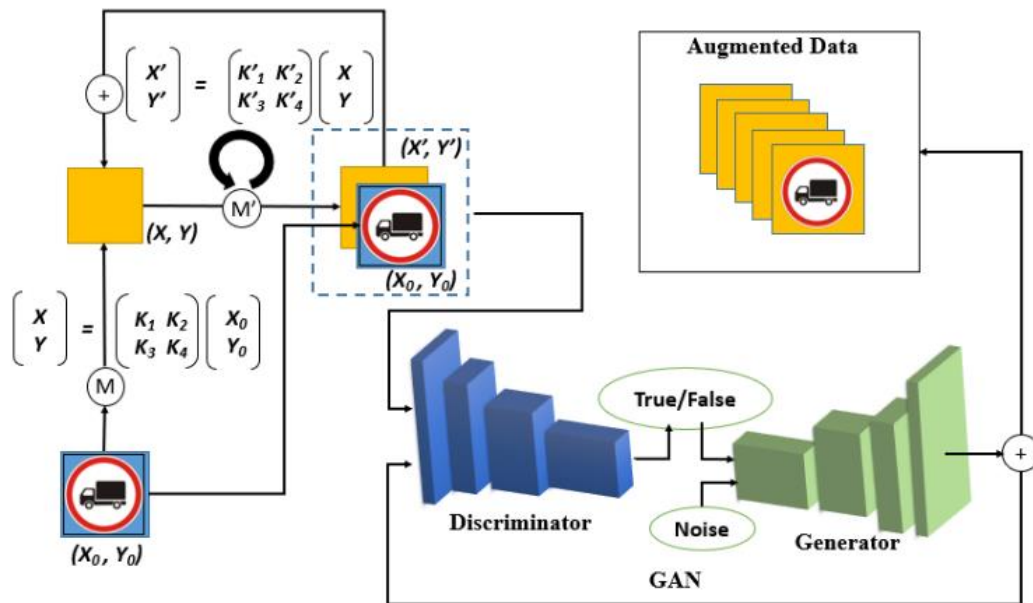


Figure 4.2: Overview of the proposed GAN framework for data augmentation from a single original image.

## 4.2.2 Perturbation Mechanism

In the proposed perturbation mechanism, the parameters are updated by the input data with batch values, which are fine-tuned by processing real images passing to the discriminator and can be regarded as the continuously updating input information. With an appropriate setup of different transformation matrices and updating frequency, the proposed GAN framework will be able to balance the similarity and diversity of the images generated from a small training dataset. The main function of the transformation matrix  $M'$  is to transform image pixels to generate new images to introduce diversity, such as rotation and scaling whilst the transformation matrix  $M$  is used to normalise the original image by resizing, cropping and alignment.

Additionally, the perturbation mechanism passes the renewed features to the discriminator at an appropriate setup by using the transformation matrices  $M$  and  $M'$ . It forces the discriminator re-learn the updated inputs with batch values. Different from the traditional DCGAN, where the input batch to the discriminator is renewed epoch by epoch, the updating of transformation matrices additionally controls the timing of the input batches. Setting a smaller updating frequency of transformation matrices is profitable for generating realistic images by the DCGAN with a larger number of training epochs. It is noticed that the updating frequency of transformation matrices affects generated images, and a larger frequency value will result in more diverse results for the synthetic outputs. Therefore, the updating frequency should be determined in a balanced manner, which specifically depends on the used image datasets and will be discussed in Section 4.2.5.

## 4.2.3 Model Building

The proposed GAN model is made up of two separate networks, the generator network  $G$  and discriminator network  $D$ . The discriminator is responsible to learn the distributions of the original data whilst the generator simulates the data distributions for data generation purposes. The designed generator receives a noise signal of  $z$  and generates data from the random noises, called random vectors. A discriminator determines whether the generated samples are realistic or not. The input parameter of the discriminator is  $x$ , which is image data forwarded to the proposed perturbation mechanism. The output in the discriminator  $D(x)$  represents the probability of real pictures. On the other hand, the input information of the generator is a mixed value of  $D(x)$  and noise vectors  $z$ . Consequently, the discriminator can be regarded as a binary

classifier that outputs 0 to generated samples and 1 to the real ones.

In contrast to the typical GANs, both networks of the proposed generator and discriminator are composed of convolutional layers rather than fully-connection layers because the convolutional neural network has been proven to perform well in both image classifications and image generations. Moreover, the algorithms of the generator and discriminator are extended from the applications of convolution neural networks.

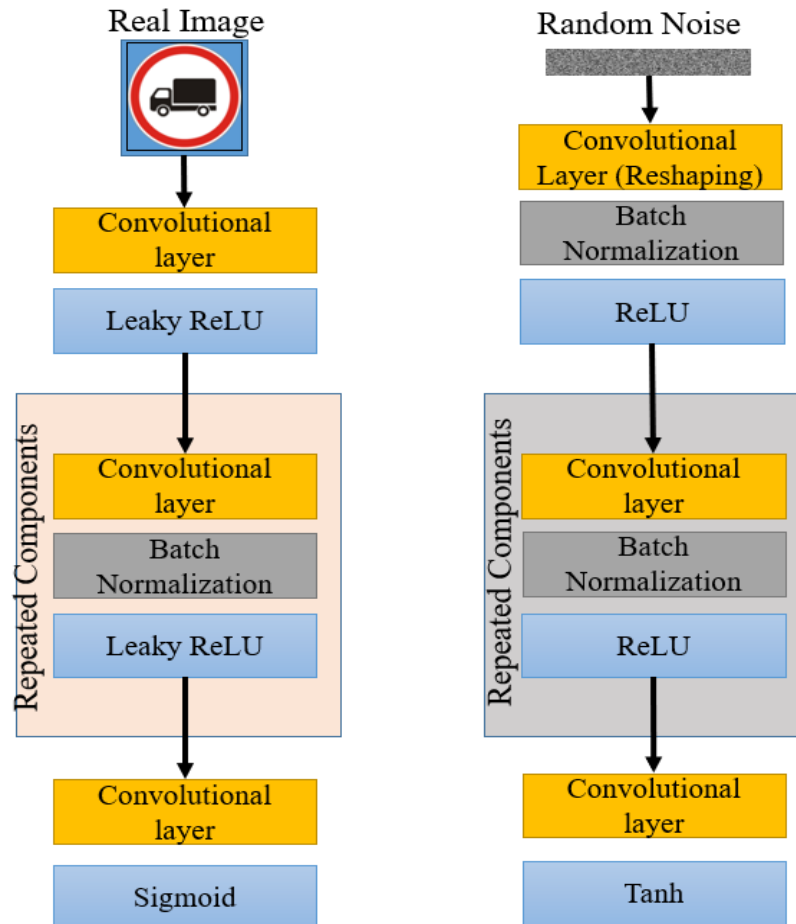


Figure 4.3: The model components of discriminator (left column) and generator (right column), where the repeated components indicate the upsampling or downsampling number of convolutional layers in the proposed GAN model.

Inspired by the framework of DCGAN, the proposed GAN model is designed on convolution-based structures. In the discriminator, batch normalisation is not required at the first convolutional layer because of the adoption of the perturbation mechanism, and the following combination components are the convolutional layer, batch normalisation, and activation function of LeakyReLU. On the other hand, in the generator, the first layer is the fully-connection layer and the following combination



components are the convolutional layer, batch normalisation, and activation function of ReLU. The last convolution layer is activated by the hyperbolic tangent (tanh) activation function. The network components of the discriminator and generator are shown in Figure 4.3.

The discriminator used for image classification in the proposed method inputs colour images of 64 by 64 pixels with 3 channels (red, green and blue). Because of the diversification of various image datasets, a mixture of transformation matrices as matrix multiplications can be used in the proposed GAN model, which not only normalises the original images but also meets the requirements for automatically generating diverse images with transformation matrices updated at appropriate frequency during the training process.

### 4.2.3.1 Generator

Regarding the similarity between original images and augmented ones, a small difference in image features needs to be specifically created by the generator. Generating slightly different results of augmented data relies on the learning capability of the generator network. To synthesise augmented features continuously, the generator should maintain a balance between the discriminator and the generator in the training phase. Based on the training balance concerns, a generator network is implemented similar to convolution-based structures. To meet the required channel numbers and generative size through convolution layers, the generator network expects to produce 3 channels of colour augmented images with a resolution of  $64 \times 64$  pixels, and 4 repeated components, shown in Figure 4.3, are designed. The ReLU is empirically used as the activation function in the generator network.

The details of the generator network and the released parameters are shown in Table 4.1. The generator model consists of a series of convolutional layers. The inputs of the generator are random vectors as the noise signal, and the size of the input is a  $100 \times 1 \times 1$  random vector, which is drawn from a Gaussian distribution. The size of the inputting vector is changed to  $1024 \times 4 \times 4$  by the convolutional transposition in the first layer, and the number of channels is decreased in the next convolutional layer. The subsequent convolutional layers are up-sampling layers till reaching the outputting vector of  $3 \times 64 \times 64$ . To generate features smoothly, the number of upsampling channels in each convolution layer is halved, and the output tensor is doubled. Finally, the last generated image is activated by the tanh activation function.

Table 4.1: The generator network and related parameters.

<i>Name</i>	<i>Type</i>	<i>Input Size</i>	<i>Output Size</i>
Conv. Layer 0	Conv. Transposition	$100 \times 1 \times 1$	$1024 \times 4 \times 4$
Batch Nor. Layer 0	Normalisation	$1024 \times 4 \times 4$	$1024 \times 4 \times 4$
ReLU 0	Activation	$1024 \times 4 \times 4$	$1024 \times 4 \times 4$
Conv. Layer 1	Conv. Upsampling	$1024 \times 4 \times 4$	$512 \times 8 \times 8$
Batch Nor. Layer 1	Normalisation	$512 \times 8 \times 8$	$512 \times 8 \times 8$
ReLU 1	Activation	$512 \times 8 \times 8$	$512 \times 8 \times 8$
Conv. Layer 2	Conv. Upsampling	$512 \times 8 \times 8$	$256 \times 16 \times 16$
Batch Nor. Layer 2	Normalisation	$256 \times 16 \times 16$	$256 \times 16 \times 16$
ReLU 2	Activation	$256 \times 16 \times 16$	$256 \times 16 \times 16$
Conv. Layer 3	Conv. Upsampling	$256 \times 16 \times 16$	$128 \times 32 \times 32$
Batch Nor. Layer 3	Normalisation	$128 \times 32 \times 32$	$128 \times 32 \times 32$
ReLU 3	Activation	$128 \times 32 \times 32$	$128 \times 32 \times 32$
Conv. Layer 4	Conv. Upsampling	$128 \times 32 \times 32$	$64 \times 64 \times 64$
Batch Nor. Layer 4	Normalisation	$64 \times 64 \times 64$	$64 \times 64 \times 64$
ReLU 4	Activation	$64 \times 64 \times 64$	$64 \times 64 \times 64$
Conv. Layer 5	Conv. Transposition	$64 \times 64 \times 64$	$3 \times 64 \times 64$
Tanh	Activation	$3 \times 64 \times 64$	$3 \times 64 \times 64$

### 4.2.3.2 Discriminator

To reach data augmentation requirements, the capability of feature extraction is a critical factor that impacts the results of augmented images. However, the feature information and gradients easily vanish through backward propagation among convolutional layers. Therefore, a small number of convolutional layers is applied in the discriminator network to extract features based on a very small number of training samples. Although a deep network has more strong capabilities to classify features, it easily results in the problems of overfitting and gradient vanishing when training with small datasets. Furthermore, training both generator and discriminator networks is time-consuming and needs powerful computing abilities. Considering that the difference of input images can be controlled by the proposed perturbation mechanism, the discriminator need not use a very deep network to identify the feature difference. As a result, the proposed discriminator network is merely designed with 4 repeated downsampling components, as shown in Figure 4.3. The repeated components in the discriminator network are composed of convolutional layers, batch normalisation layer and activation function, where convolutional layers are responsible for feature

extraction.

Normalisation plays an important role in training the discriminator because one image might be merely involved during a training phase. According to the idea of model designing, normalisation could focus on one single image. The batch normalisation layer pays attention to the overall distribution of input images and often ensures the consistency of data distribution. Batch normalisation calculates the mean and standard deviation values in each batch, which can affect the correction of renewed images sent from the perturbation mechanism. Therefore, the output information in the discriminator network can be controlled by normalisation with a progressive view of each batch. In image generation tasks, the information obtained by batch normalisation will also provide benefits with a whole view of the updated data. Consequently, batch normalisation learns information directly from a single image, and it can maintain the independence of renewed images among batches.

Table 4.2: The Discriminator network and related parameters.

<i>Name</i>	<i>Type</i>	<i>Input Size</i>	<i>Output Size</i>
Conv. Layer 0	Conv. Transposition	$3 \times 64 \times 64$	$64 \times 64 \times 64$
Leaky ReLU 0	Activation	$64 \times 64 \times 64$	$64 \times 64 \times 64$
Conv. Layer 1	Conv. Downsampling	$64 \times 64 \times 64$	$128 \times 32 \times 32$
Batch Nor. Layer 1	Normalisation	$128 \times 32 \times 32$	$128 \times 32 \times 32$
Leaky ReLU 1	Activation	$128 \times 32 \times 32$	$128 \times 32 \times 32$
Conv. Layer 2	Conv. Downsampling	$128 \times 32 \times 32$	$256 \times 16 \times 16$
Batch Nor. Layer 2	Normalisation	$256 \times 16 \times 16$	$256 \times 16 \times 16$
Leaky ReLU 2	Activation	$256 \times 16 \times 16$	$256 \times 16 \times 16$
Conv. Layer 3	Conv. Downsampling	$256 \times 16 \times 16$	$512 \times 8 \times 8$
Batch Nor. Layer 3	Normalisation	$512 \times 8 \times 8$	$512 \times 8 \times 8$
Leaky ReLU 3	Activation	$512 \times 8 \times 8$	$512 \times 8 \times 8$
Conv. Layer 4	Conv. Downsampling	$512 \times 8 \times 8$	$1024 \times 4 \times 4$
Batch Nor. Layer 4	Normalisation	$1024 \times 4 \times 4$	$1024 \times 4 \times 4$
Leaky ReLU 4	Activation	$1024 \times 4 \times 4$	$1024 \times 4 \times 4$
Conv. Layer 5	Conv. Transposition	$1024 \times 4 \times 4$	1
Sigmoid 1	Classifier	1	1

The related components and parameters of the discriminator network are shown in Table 4.2. The first convolutional layer is connected by the output of the perturbation mechanism, which is a 3-channel colour image with  $64 \times 64$  pixels, shown as  $3 \times 64 \times 64$ . The following convolutional layers are downsampling the vectors and increasing

the number of channels to the next layer as well. The ReLU is implemented as the activation function.

Connectivity between the perturbation mechanism and discriminator network is a critical strategy to improve the capability of feature extraction. To augment images, data diversity is a critical considering factor that impacts the performance of image classification. The feature diversity is expected to be identified by the discriminator network, and the perturbation mechanism can input controllable and renewable features to the discrimination network. This process takes advantage to extract more different representations by renewing the prior fine-tuned parameters in the discriminator. An appropriate design of the discriminator network coupled with the perturbation mechanism will alleviate the serious training difficulty contrasted to typical GANs. Therefore, the operations between the perturbation mechanism and discriminator are supposed to mitigate the negative influence caused by a small number of training samples.

#### 4.2.4 Loss Functions

Two main loss functions are used for training the proposed GAN model. First of all, the loss of the generator is the sigmoid of cross-entropy given by the discriminator to score the chosen image and generated images. Secondly, another loss is to compute the similarity between the original image and generated image with weights on the end layer of the discriminator. Consequently, the overall loss for the proposed method is the sum of these two losses, shown in the following function.

$$\begin{aligned} \min_G \max_D V(D, G) \\ = \mathbb{E}_{x \sim P_{data}} [\log D(x)] + \mathbb{E}_{z \sim P_{data}} [\log (1 - D(G(z)))] \end{aligned} \quad (4.1)$$

where  $x$  is the real image, and  $D(x)$  is the probability values calculated by the discriminator network when inputting image  $x$  as the training sample.  $G(z)$  is a sample generated by the generator network with the input vectors of noise  $z$ . The  $D(G(z))$  indicates the probability of the generated sample being created by the generator.

#### 4.2.5 Training Process

The entire training process is described in Algorithm 1. It is mainly a recurrent and

iterative training process based on the perturbation mechanism, generator network and discriminator network. In each training epoch, the transformation matrix of  $M$  normalises the input image and passes the calculated vectors to the next transformation matrix of  $M'$ , where  $M$  and  $M'$  are  $2 \times 2$  matrices used to transfer the original set of image pixels to a new set. Furthermore, the transformation matrix of  $M'$  updates the vector received from the matrix of  $M$  until the end of the repetitive loops at this step. The perturbation mechanism is updated with  $j$  times. The outputs of the perturbative results are updated toward the discriminator and generator at each iteration. A standard gradient-based optimisation method is conducted to learn with the loss functions, where the SGD and Adam optimiser are implemented in the experiments [186], [187].

---

**Algorithm 1** Proposed GAN Model for Data Augmentation Purposes

---

1. **Input:** training image  $(x, y)$
  2. **Normalisation:** transferring image  $(x, y)$  to the size of  $64 \times 64$  pixels with 3 channels.
  3. **for** the number of training epochs **do**
  4.     transfer image  $(x, y)$  to  $(x_0, y_0)$  by transformation matrix  $M$ :  

$$(x_0, y_0) = (x, y) \times M$$
  5.     **for**  $k$  steps **do**
  6.         a random boolean value  $\{ b_i = 1 \text{ or } 0 \mid i = 1, \dots, k \}$
  7.         **if**  $(b_i = True)$  **do**
  8.             transfer image  $(x_0, y_0)$  to  $(x', y')$  by transformation matrix  $M'$ :  $(x', y') = (x_0, y_0) \times M'$
  9.         **else**  $(x', y') = (x_0, y_0)$
  10.         **end if**
  11.         **for**  $j$  steps **do**
  12.             generated image  $(x_j, y_j) = G\{D[(x', y'), (x, y)]\}$
  13.             saving image  $(x_j, y_j)$  as one of the augmented samples
  14.             optimising Generator and Discriminator with loss functions and Adam Optimiser
  15.         **end for**
  16.     **end for**
  17. **end for**
- 

As seen in Algorithm 1, the proposed model has two main phases, the perturbation mechanism phase and the GAN training phase. In the perturbation mechanism phase, two primary matrices of  $M$  and  $M'$  are conducted. For the first matrix of  $M$ , one of the target images of  $(x, y)$  in the dataset is chosen as the input data  $(x, y)$ . The input image

of  $(x, y)$  will be transferred to  $(x_0, y_0)$  using the transformation matrix of  $M$ . Then, a perturbation is repetitive by  $k$  times, and a random boolean set controls the activation of  $M'$ , which changes the  $(x_0, y_0)$  as a new image data  $(x', y')$ .

In the GAN training phase, there are several steps for the proposed model to complete the training process: 1) At the first step of the GAN training phase, the perturbative outputs  $(x', y')$  pass through the computing processes as the forwarded input vectors. 2) Secondly, the input vectors pass through the convolutional layer and batch normalisation layers to obtain the feature maps. 3) Thirdly, the feature maps go through the activation function and get the activation maps to the next convolutional layers. 4) Furthermore, the value of a designed loss function is computed. 5) Finally, the loss value is backpropagated to update the weights. During the training phase, the steps are repeated until the loss is converged.

On another side, the forward and backward loss values should keep a balance status when the two convolutional neural networks of the generator and discriminator are trained in the proposed GAN model. Considering both the quality and diversity of generated images, the following setups should be taken: 1) Firstly, the number of channels inputting to the discriminator for each batch should be appropriately set with different image types. For instance, 3 channels are set in every batch for colour images because  $(x_0, y_0)$  and  $(x', y')$  respectively contain channels of red, green and blue. 2) Secondly, the learning rate for the generator should be larger than that for the discriminator. This setting forces the generator to learn faster than the discriminator. If the learning rate in the generator is equal to or smaller than that in the discriminator, it will be difficult to attain qualitative results by the proposed frameworks with the transformation matrices updated at a certain frequency. 3) Thirdly, for the sake of image diversity, the transformation matrix  $M'$  can be designed as a mixture of various basic transformation matrices (*e.g.*, rotation, scaling, brightness adjustment, *etc.*). Updating the transformation matrix  $M'$  with small changes after an interval of a certain number of epochs, which is called a training cycle, will be helpful for the proposed GAN to generate highly realistic images without large distortion. 4) Finally, there is no doubt that if the discriminator can learn from one single batch with more epochs, it can be expected to generate images of higher quality. The updating frequency or the number of epochs for the training batch can be controlled by the parameter value of  $k$  and  $j$  to update the input features by using transformation matrices. More detail about the parameter setting will be discussed in the experimental section.

## 4.3 Experiments with the Proposed GAN Framework

### 4.3.1 Data Preparation

Two image datasets were used in the experiments to evaluate the performance of the proposed methods. A small part of the datasets, from 1 to 20 images per class, were used as training data, and the remaining images were for validation data. The first dataset is the MNIST [178], [179], which contains grayscale images of handwritten single-digit numbers with  $28 \times 28$  pixels. MNIST has a training set of 60,000 samples and a test set of 10,000 samples. The second dataset is the rock paper scissors (RPS) dataset [180], which contains 840 training images and 124 test images in each class from 3 hand gestures of the rock-paper-scissors game. Each image has  $300 \times 300$  colour pixels. In our experiments, a small number of images per class were randomly selected from the original training set to form a small training dataset. Moreover, for the MNIST dataset, 1,000 images per class were randomly selected as our validation dataset; for the RPS dataset, the remaining images, except for the chosen training set, and all the test images were mixed as the validation dataset in our experiments.

### 4.3.2 Hardware & Software

In the implementation, Python and TensorFlow were used as deep learning frameworks to build the models and networks, and Matlab was used to compute the accuracies of image classification tasks. All of our experiments from Chap 4 to Chap 6 were conducted on a desktop computer with a processor of Intel Core i7-6700 (3.4GHz) and 16G RAM. An exception in this chapter was all the generative experiments without using any Graphics Processing Units (GPUs) to illustrate the generative efficiency of the proposed model. Table 4.3 and Table 4.4 shows the hardware environment and software version separately.

In terms of the synthetic efficiency and basic hardware requirements, the hardware specifications intentionally skip the machines of GPUs to accelerate the experimental process. The executive time is 20 to 30 minutes to generate around 256,000 augmented results with the size of 64 pixels by 64 pixels, all of which are trained by a single original image within 4,000 epochs based on the proposed algorithm.

Table 4.3: Hardware environment.

<i>Configuration</i>	<i>Value</i>
<i>Processor</i>	Intel(R) Core(TM) i7-6700 @ 3.40 GHz 3.41 GHz
<i>Installed RAM</i>	16.0 GB
<i>System Type</i>	64-bit Operation System, x64-based Processor
<i>Hard Disk</i>	500 GB
<i>GPU</i>	None

Table 4.4: Software version.

<i>Configuration</i>	<i>Value</i>
<i>Operation System</i>	Windows 10 Education
<i>Anaconda</i>	Individual Edition 2020.11
<i>Python</i>	3.7.2
<i>TensorFlow</i>	1.14.0
<i>Matlab</i>	R2019b

### 4.3.3 Hyperparameters Setting

The setting of parameter values is demonstrated in Table 4.5. For data augmentation, finding a balance between the generator and discriminator is critical to generate fake images with both similarity and diversity, especially when there are only a few images as the training data. This can be done by properly setting hyper-parameter values, such as learning rates and frequency for updating transformation matrices. In other words, when the GAN model is trained with inappropriate hyper-parameter settings, the synthetic images could be over-diversified, as shown in Figure 4.4, where the images generated by the proposed GAN framework are implemented with a rotation angle of 50 degrees of the transformation matrix  $M'$  using one training cycle only. Additionally, generative results with the same learning rate of both discriminator and generator are hard to be recognised as a similarity compared to the original image. Obviously, adding the type of generated images as the augmented data cannot enhance image classification performance.



Table 4.5: The setting of parameter values.

<i>Parameters</i>	<i>Values</i>
<i>Image Size</i>	$3 \times 64 \times 64$
<i>Batch Size</i>	1
<i>Noise Vector Size</i>	$100 \times 1 \times 1$
<i>Learning Rate of Generator</i>	0.0002
<i>Learning Rate of Discriminator</i>	0.0001
<i>Optimiser</i>	Adam
$\beta_1$	0.5
$\beta_2$	0.999
<i>Kernel Size</i>	$4 \times 4$
<i>Batch Normalisation</i>	Discriminator & Generator
<i>Dropout</i>	None
<i>Dimension Extension</i>	Upsample to $2 \times 2$
<i>Dimension Deduction</i>	Convolution with strike 2
<i>Number of Kernel</i>	1024 to 64

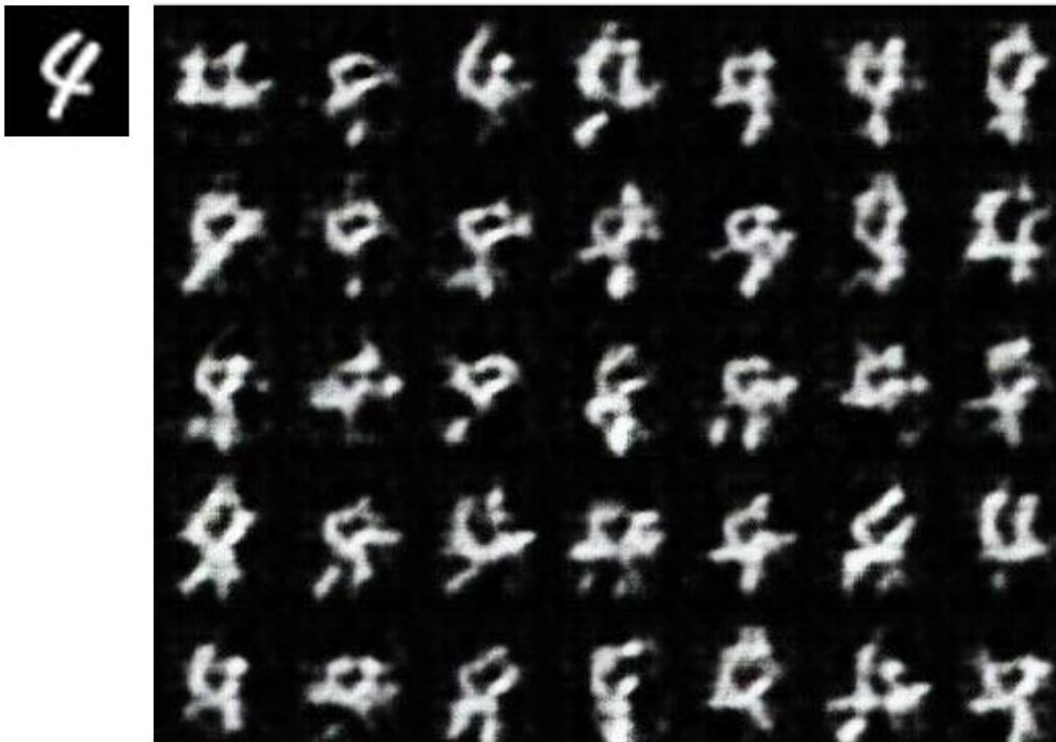

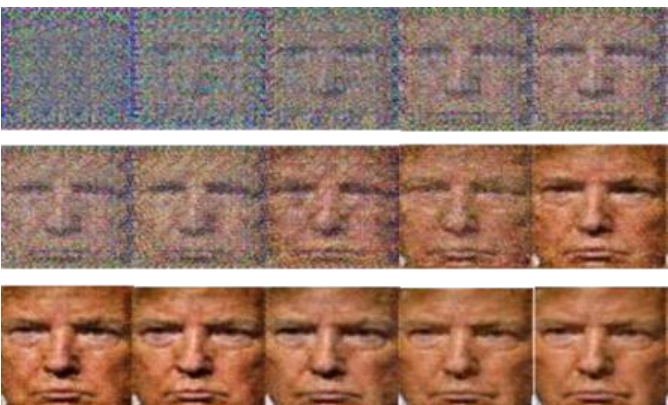

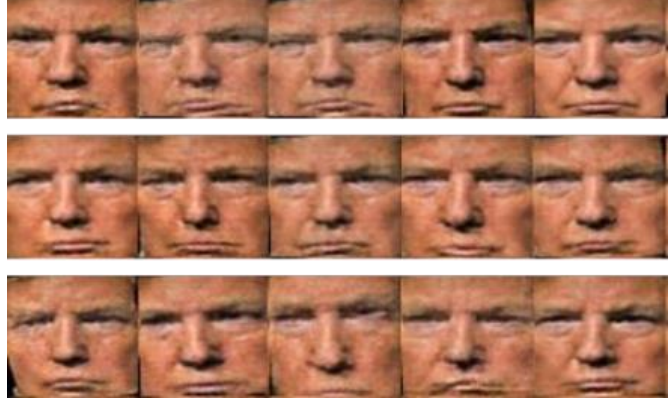
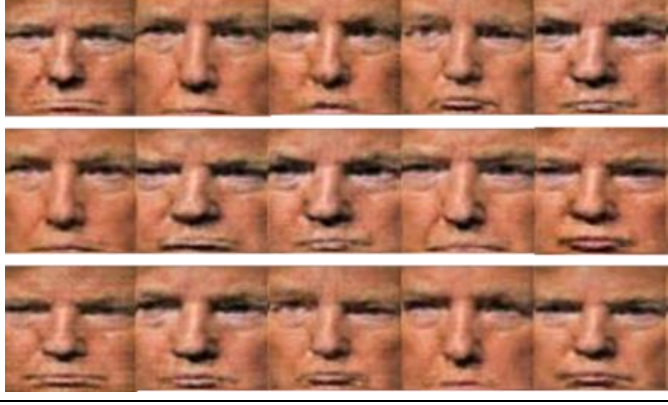





Figure 4.4: The original image and generated images with a bad setup of generating images from a single original image using the proposed GAN framework.

Table 4.6: Augmented images from the original image using different transformation matrices  $M'$ .

$(x_0, y_0)$	$(x', y')$	Augmented Images
	$K'_1 = 1$ $K'_2 = 0$ $K'_3 = 0$ $K'_4 = 1$	
		
	$K'_1 = \cos 15^\circ$ $K'_2 = -\sin 15^\circ$ $K'_3 = \sin 15^\circ$ $K'_4 = \cos 15^\circ$	
	$K'_1 = 1.3$ $K'_2 = 0.0$ $K'_3 = 0.0$ $K'_4 = 1.1$	
Assigned $(x', y')$		

As described in the previous section, the designation of transformation matrix  $M'$  should be modified with a small range for each training cycle, and the learning rate between discriminator and generator should be adjusted, if necessary, when training the proposed GAN to prevent extreme distortions. The number of epochs within a training cycle and thus the frequency for updating transformation matrices can be optimised through a trial-and-error approach in the experiments. For generating high-quality images with large rotation angles, transformation matrix  $M$  can be used to normalise the original image in rotation transformation to reduce unexpected distortions.

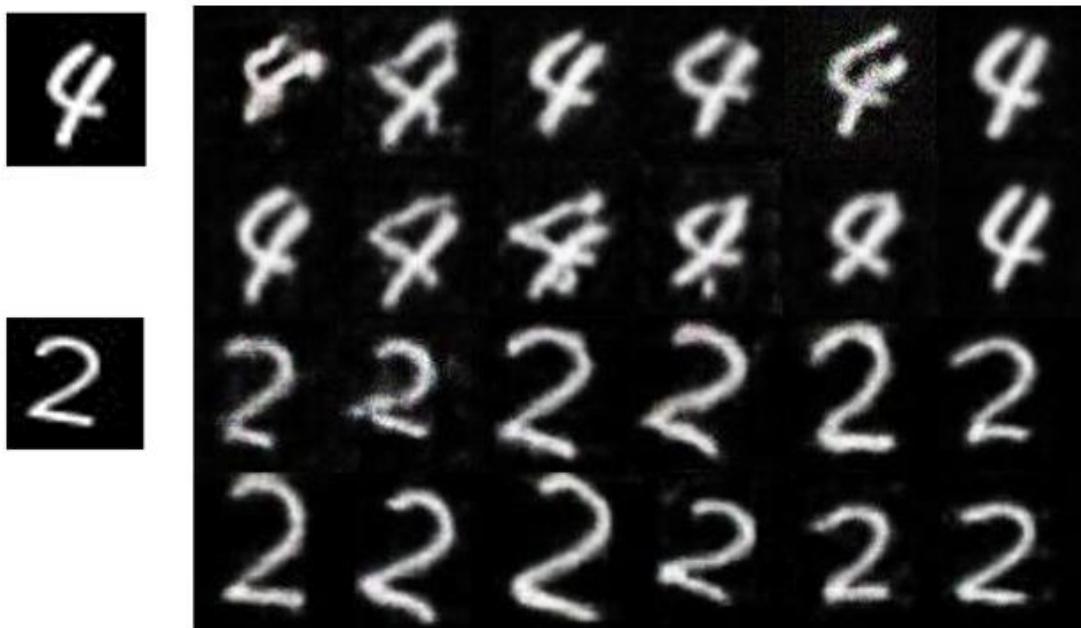


Figure 4.5: Original images (left column) for training and the generated images (right column) using the proposed model with different transformation matrices  $M'$  on the MNIST dataset.

There are different methods available for designing the transformation matrices, which can be adaptive or fixed with values assigned based on prior knowledge or experience. In our experiments, some simple transformation matrices  $M'$  demonstrated the effect with an identity matrix  $M$ . As shown in Table 4.6, the original image is a face, and various high-quality images with good diversity can be generated using three different  $M'$  matrices for rotation and scaling with the proposed GAN framework. On the other hand, if the desired transformation is very hard to implement by using the transformation matrix, it is practical to directly assign a new picture rather than using a transformation formula to guide the synthetic processes, as shown at the bottom of Table 4.6. Take a face mixture task for instance, two different faces are required to be

fused as one, but the generated results cannot be extended from one face by merely using the transformation matrices. Since the augmented images cannot be derived from the input image  $(x_0, y_0)$ , it is applicable to directly allocate another image as the assigned  $(x', y')$ . However, when applying an assigned image, attention should be paid to the alignment and normalisation using the matrix  $M$  in the proposed method. In our experiments, the GAN training ran for 4,000 epochs using the assigned transformation matrix  $M'$ . The Adam optimiser was used with a learning rate of 0.0002 for the generator and a learning rate of 0.0001 for the discriminator.

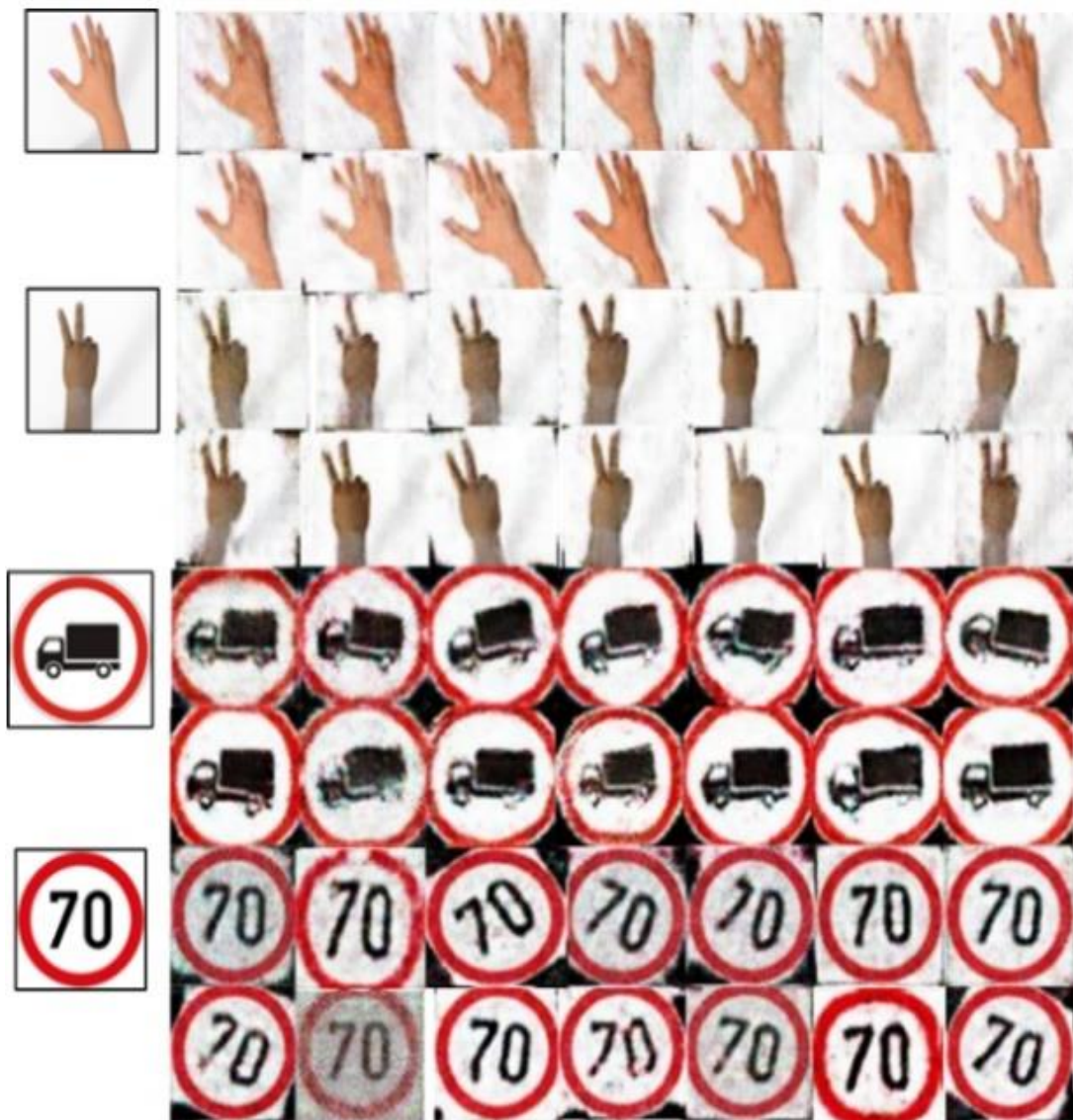


Figure 4.6: Original images and the generated images with small-scale rotations implemented by matrix  $M'$  and larger-scale rotations implemented by matrix  $M$ .

Due to the different requirements of image classification, the transformation matrix  $M'$  can be designed for the GAN to simulate real images with class-informative features.

Taking the MNIST for instance, the provided images contain many similar features with diverse rotation angles, and assigning the transformation matrix  $M'$  with different rotation angles can generate augmented training data for better classification performance than other transformation formulas. In addition, the MNIST dataset contains features without complicated lines, colours or textures. Diverse images of high quality can be generated using a well-designed matrix  $M'$  with the proposed GAN framework. Figure 4.5 shows the images generated by the proposed GAN using transformation matrices  $M'$  mixed with rotations of 10 to 30 degrees and scaling of 1 to 1.4 times. It can be seen that the proposed GAN framework can generate more diverse images than the traditional rotation and scaling transformation.

It is worth noting that if transformation matrix  $M'$  varies on a large scale at every training cycle, it will cause dramatic distortions and fail to generate high-quality images. Therefore, a slight variance of the transformation matrix  $M'$  in each cycle will be useful to generate meticulous texture and colour on the RPS dataset, as well as on some realistic images. On the other hand, transformation matrix  $M$  can be used to make large-scale image changes rather than directly applying large transformation using transformation matrix  $M'$ . Figure 4.6 demonstrates images generated by the proposed GAN framework with large-scale rotations implemented by matrix  $M$  and slight rotations implemented by matrix  $M'$ . It can be seen that the generated images are diverse and of good quality without dramatic or unidentical distortions.

## 4.4 Evaluation of Training Data Augmentation for Image Classification

In this section, several comparative experiments are conducted to evaluate the performance of augmented datasets generated by the proposed GAN model: Firstly, the augmented images are used to compare the classification accuracies with the original datasets via the method of transfer learning. Secondly, the student's t-test is designed to compare the performance from a statistical perspective. Thirdly, based on the worse performance in the student's t-test, a further experiment on the confusion matrix is implemented to evaluate that the augmented images can confidently improve the performance of CNNs. Finally, the augmented images generated by our proposed GAN model are compared with popular traditional augmentation methods, and which experiment aims to compare the difference between other common augmentation techniques.

## 4.4.1 Performance Comparison of Image Classification with Augmented Image Data

In order to evaluate the effectiveness of using the images generated by the proposed GAN framework for training data augmentation, transfer learning with 4 famous pre-trained CNNs, AlexNet, VGGNet-16, GoogLeNet and ResNet-18, were used for image classification on the MNIST and RPS datasets respectively. A very small number of training images is formed by random selections from the training set of MNIST and RPS. Due to divergent loss values often appearing during fine-tuning the pre-trained CNNs, especially when there are only 1 to 10 training samples per class, the validation accuracy has been calculated as the mean value of the best 10 runs to reach reliable results.

Figure 4.7 and Figure 4.8 show the validation accuracy values of the 4 used CNNs with different numbers of original training samples and augmented training samples based on the MNIST dataset and RPG dataset respectively. The number of original training samples was set to 1, 5, 10, 15, and 20 respectively for producing results with a small training dataset. The same number of augmented samples per training image was selected although more images could be generated using the proposed GAN framework, and the remaining samples on the MNIST and RPS datasets were used as the validation data. At most 20 original training samples per class were used in the experiment for performance comparison, because the four CNNs trained with 20 samples per class can achieve quite reliable validation accuracy, up to 85% to 92%, which is robust enough for real-world classification on a small dataset and suitable to investigate performance enhancement by training data augmentation. From Figure 4.7 and Figure 4.8, it can be seen that using the images generated by the proposed GAN model for training data augmentation can improve the validation accuracy of the four CNNs by 3~35%, depending on the number of original training samples used.

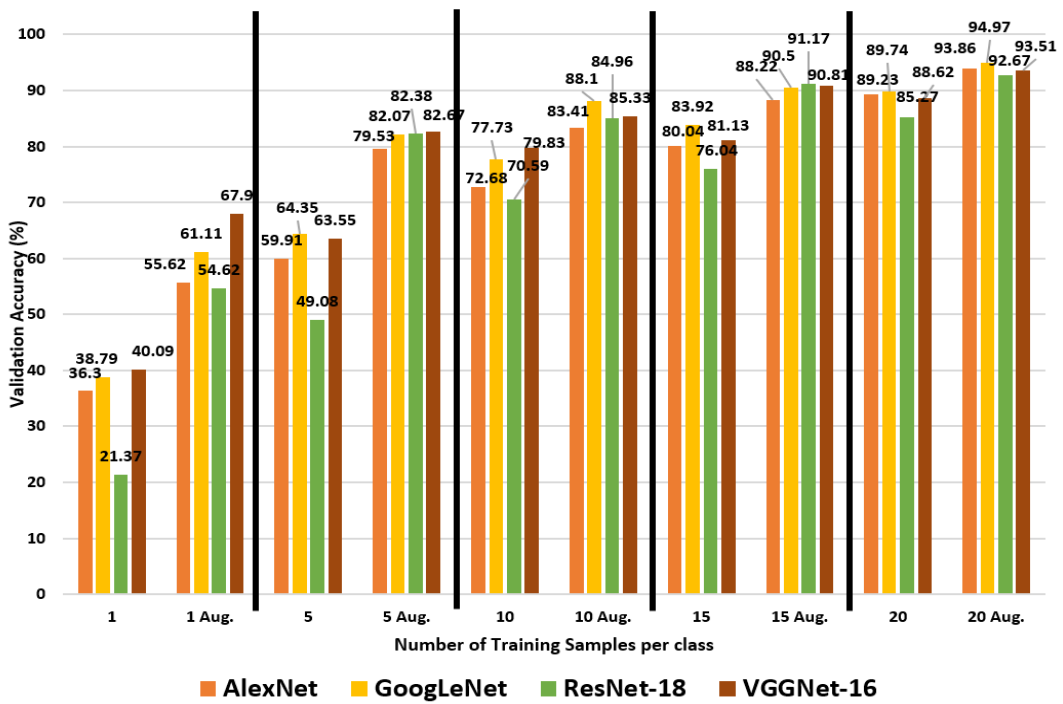


Figure 4.7: Comparison of validation accuracy of CNNs on the MNIST dataset.

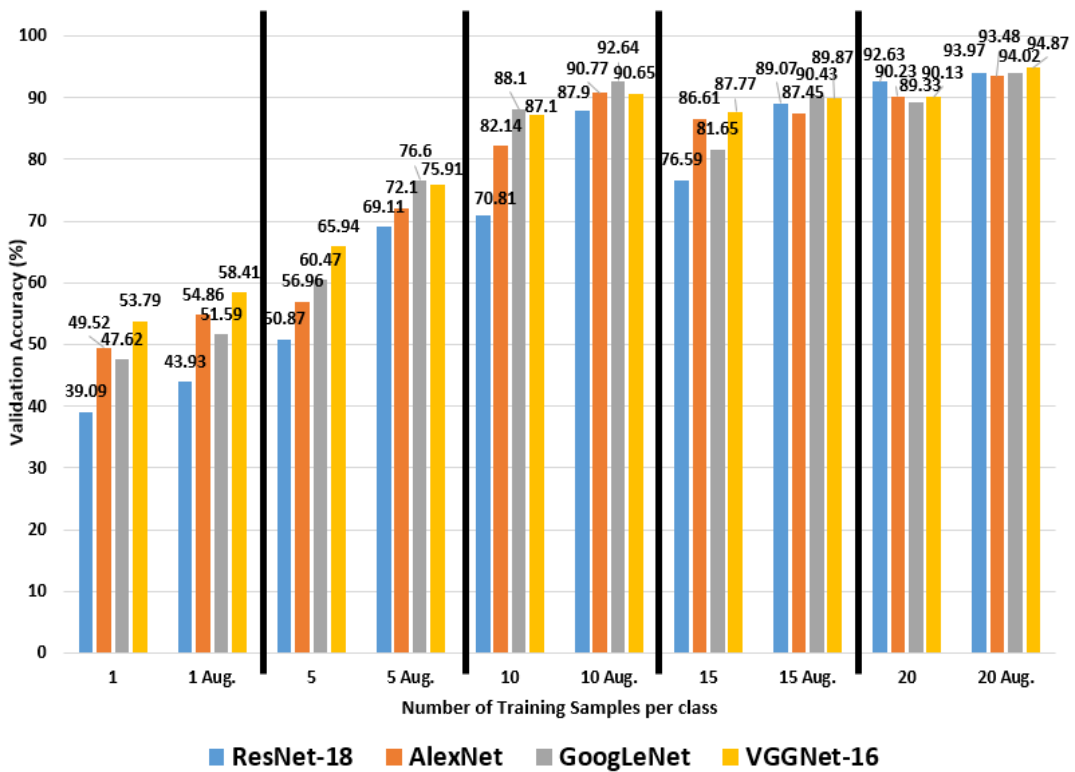


Figure 4.8: Comparison of validation accuracy of CNNs on the RPS dataset.

#### 4.4.1.1 Student's T-test

The student's t-test with the following statistical hypotheses has been conducted to find out whether the performance improvement of the training data augmentation is statistically significant in the classification trials:

- Null hypothesis ( $H_0$ ) - the mean accuracy achieved with augmented training data is equal to the mean accuracy achieved without using augmented training data at the 5% significance level.
- Alternative hypothesis ( $H_1$ ) - the mean accuracy achieved with augmented training data is greater than the mean accuracy achieved without using augmented training data at the 5% significance level.

Table 4.7 presents the p-values obtained from the t-test comparing the validation accuracy with augmented training data and the validation accuracy without augmented training data for each CNN on the MNIST and RPS datasets respectively. The p-values were calculated by the 15 best classification results between augmented and non-augmented data based on the same 10 training samples of each class. It can be seen that all the t-test results reject hypothesis  $H_0$  and accept hypothesis  $H_1$ , with p-values much smaller than 0.05. Therefore, the performance improvement by the proposed method for training data augmentation is statistically significant.

Table 4.7: Significance test results (p-values) for comparing validation accuracy of CNNs trained with vs without augmented training data.

	<i>MNIST</i>	<i>RPS</i>
<i>AlexNet</i>	<u><b>2.09e-06</b></u>	<u><b>1.01e-04</b></u>
<i>VGGNet</i>	3.08e-07	2.42e-05
<i>GoogLeNet</i>	2.48e-06	1.36e-05
<i>ResNet</i>	2.68e-06	5.24e-06

#### 4.4.1.2 Evaluation by Confusion Matrix

Referring to the p-values in the student's t-test, shown in Table 4.7, all four CNNs in the experiment had the statistical confidence that the augmented data achieve significant impacts on the final classification accuracies. However, AlexNet achieved the worst performance on both MNIST and RPS datasets, shown as the red marks. In this experiment, further evaluations were conducted with confusion matrices to further



analyse the performance of the AlexNet. Figure 4.9 to Figure 4.12 demonstrate the confusion matrixes for the testing data when trained with and without augmented images on MNIST and RPS datasets respectively. The confusion matrices are based on the classification results of the AlexNet trained with 10 randomly chosen images per class as a small and balanced training dataset.

**Confusion Matrix for Testing Data**

True Class	paper	458	382		54.5%	45.5%
	scissors	27	813		96.8%	3.2%
	stone	66	66	708	84.3%	15.7%
		83.1%	64.5%	100.0%		
		16.9%	35.5%			
		paper	scissors	stone		
		Predicted Class				

Figure 4.9: Confusion matrix for testing data on the RPS dataset. The AlexNet is trained without using augmented data.

**Confusion Matrix for Testing Data**

True Class	paper	553	144	143	65.8%	34.2%
	scissors		840		100.0%	
	stone	5	11	824	98.1%	1.9%
		99.1%	84.4%	85.2%		
		0.9%	15.6%	14.8%		
		paper	scissors	stone		
		Predicted Class				

Figure 4.10: Confusion Matrix for testing data on the RPS dataset. The AlexNet is trained with the original and augmented data.

### Confusion Matrix for Testing Data

True Class	0	416	2	1	23			1	1		56	83.2%	16.8%
	1	1	406	13	3	15			62			81.2%	18.8%
	2		13	246	105	7			120	3	6	49.2%	50.8%
	3		4	2	364	12	49		61	8		72.8%	27.2%
	4		38		21	390			50	1		78.0%	22.0%
	5	1	8		207	2	243		35	3	1	48.6%	51.4%
	6	29	10		43	8	4	356	2	24	24	71.2%	28.8%
	7		169	3	3	9			316			63.2%	36.8%
	8				30	11		2	1	443	13	88.6%	11.4%
	9	4		2	22	14			28	1	429	85.8%	14.2%
		92.2%	62.5%	92.1%	44.3%	83.3%	82.1%	99.2%	46.7%	91.7%	81.1%		
		7.8%	37.5%	7.9%	55.7%	16.7%	17.9%	0.8%	53.3%	8.3%	18.9%		
		0	1	2	3	4	5	6	7	8	9		
		Predicted Class											

Figure 4.11: Confusion Matrix for testing data on the MNIST dataset. The AlexNet is trained without using augmented data.

### Confusion Matrix for Testing Data

True Class	0	495						5				99.0%	1.0%
	1		409	25		23			43			81.8%	18.2%
	2	2	25	442	10			8	7		6	88.4%	11.6%
	3			30	411		46		7	6		82.2%	17.8%
	4		9	10	1	467			7	6		93.4%	6.6%
	5		12	12	87		381	1	5	1	1	76.2%	23.8%
	6	15	1	2	4	2	7	465			4	93.0%	7.0%
	7	10	124	9					357			71.4%	28.6%
	8	7		18	4	2	2	13		454		90.8%	9.2%
	9	29		6			2	15	11		437	87.4%	12.6%
		88.7%	70.5%	79.8%	79.5%	94.5%	87.0%	91.7%	81.7%	97.2%	97.5%		
		11.3%	29.5%	20.2%	20.5%	5.5%	13.0%	8.3%	18.3%	2.8%	2.5%		
		0	1	2	3	4	5	6	7	8	9		
		Predicted Class											

Figure 4.12: Confusion Matrix for testing data on the MNIST dataset. The AlexNet is trained with the original and augmented data.

The validation data in the MNSIT dataset consisted of the same number of 500 images per class, while the RPS dataset had 840 images per class. The same training parameters were used to compare the efficiency between a small training dataset and the augmented images generated by the proposed GAN modes. AlexNet was applied as a convolutional network. For the dataset of MNIST and RPS, Figure 4.9 and Figure 4.11 respectively illustrate the confusion matrix without using the augmented data, and Figure 4.10 and Figure 4.12, on the other side, show the analysis results with augmented data.

It can be found that augmented data enhance image classification performance, which significantly reduces the negative effects of training with a very small dataset. According to the experimental results, for the MNIST dataset, adding augmented data into original datasets acquired a higher validation accuracy of 86.37% than the original accuracy of 71.59% when only 10 images per class are involved to fine-tune the hyperparameters in AlexNet. Similarly, the performance of the RPS dataset also exhibits a better classification accuracy of 87.8% with the augmented data than that of 78.08% without using augmented data.

For further evaluating the classification performance between the augmented data and the original small dataset. A set of 4 estimation factors (sensitivity, specificity, accuracy and precision) was calculated to evaluate the performance of augmented data in AlexNet. The provided equations to compute the four factors are shown as follows.

$$Sensitivity = \frac{T_P}{T_P + F_N} \quad (4.2)$$

$$Specificity = \frac{T_N}{T_N + F_P} \quad (4.3)$$

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad (4.4)$$

$$Precision = \frac{T_P}{T_P + F_P} \quad (4.5)$$

where the sensitivity is measured by true-positive ( $T_P$ ) and false-negative ( $F_N$ ), and the specificity is calculated by true-negative ( $T_N$ ) and false-positive ( $F_P$ ). The accuracy is based on true-positive ( $T_P$ ), false-positive ( $F_P$ ), true-negative ( $T_N$ ), and false-negative ( $F_N$ ). The precision is measured with true-positive ( $T_P$ ) and false-positive ( $F_P$ ).

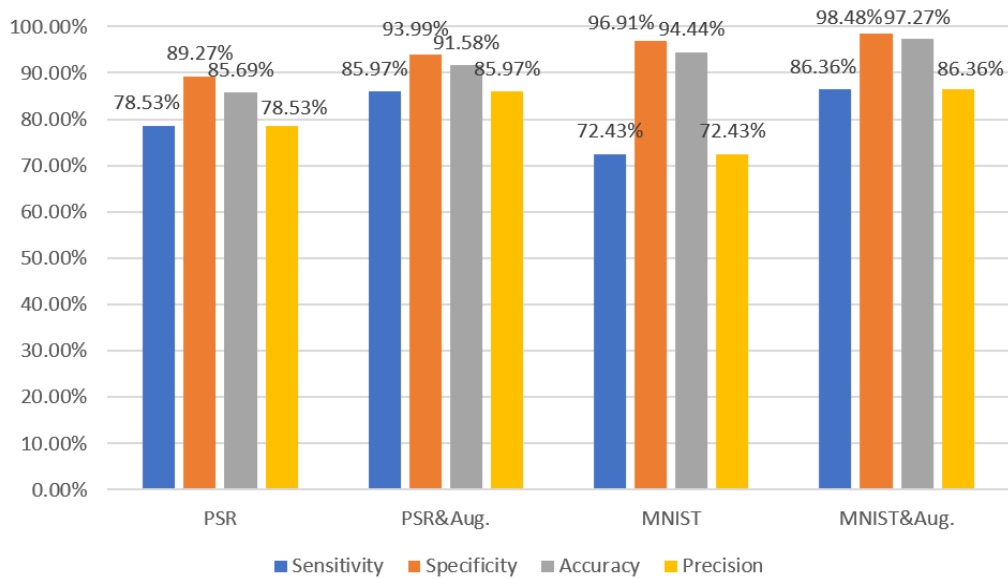


Figure 4.13: A comparison of the sensitivity, specificity, accuracy and precision with augmented data and original small data to train the AlexNet.

Figure 4.13 shows the calculated results with and without using the augmented data in terms of accuracy, sensitivity, specificity, and precision. The results depict that the augmented images on the dataset of MNIST can be effectively classified by the high factor scores of sensitivity, specificity, accuracy, and precision of 86.36%, 98.48%, 97.27%, and 86.36% respectively. On the other hand, the augmented images on the dataset of RPS also receive higher scores for the sensitivity, specificity, accuracy, and precision of 85.97%, 93.99%, 91.58%, and 85.97% respectively. In short, compared to the small datasets without using augmented data, the generated images can comprehensively enhance the classification performance evaluated by the four factors of sensitivity, specificity, accuracy, and precision.

Both the results for confusion matrices in Figure 4.10, Figure 4.12 and estimation factor values in Figure 4.13 show the aggregate statistical results with augmented datasets; augmented images generated by the proposed GAN model are advantageous to boost the final classification performance based on the same validation set, even when AlexNet statistically reaches the worse classification performance among the other CNNs in the student's t-test experiment. Moreover, a small number of images (10 random images) was used as the training samples, and the AlexNet trained by augmented images can still comprehensively improve the classification performance. To sum up, it is proven from the experimental results that when a small number of samples is involved in the training dataset, the validation accuracies in image classification can be boosted with the augmented images generated by the proposed GAN model.

### 4.4.1.3 Evaluation by t-SEN Plot

To visualise the final distributions of validation performance data by training with the augmented data and original data in the previous experiment, t-distributed stochastic neighbour embedding (t-SNE) was applied to evaluate the data distribution of the last softmax layer of the AlexNet. The t-SNE is a common dimensional reduction algorithm to visualise the data in high dimensions. For instance, the MNIST contains 10 dimensions (10 classes) in the final softmax layer, and the t-SNE can transform the high-dimensional data into a two-dimensional distribution as the t-SNE plot for the visualisation of classified data distributions. A t-SNE plot applies the distance in each dataset and weights the correlation with another dataset. Figure 4.14 and Figure 4.15 illustrate the two-dimensional scatter plot using the t-SNE method to visualise the data distribution of the softmax layer when the original training data and augmented training data are used separately. According to the t-SEN plots for the validation data distribution in the last softmax layer of the AlexNet, a conclusion can be made that the classification results using augmented images keep the validation data distributions more uncovered and less overlapped than training with original images. Training with augmented data distributes the validation data into more correct dimensions with a high validation accuracy of image classification when a clear distribution margin is acquired in the last layer. A difference between the t-SNE plots in Figure 4.15 may not be visually clear due to the large class number obscuring the classification improvement.

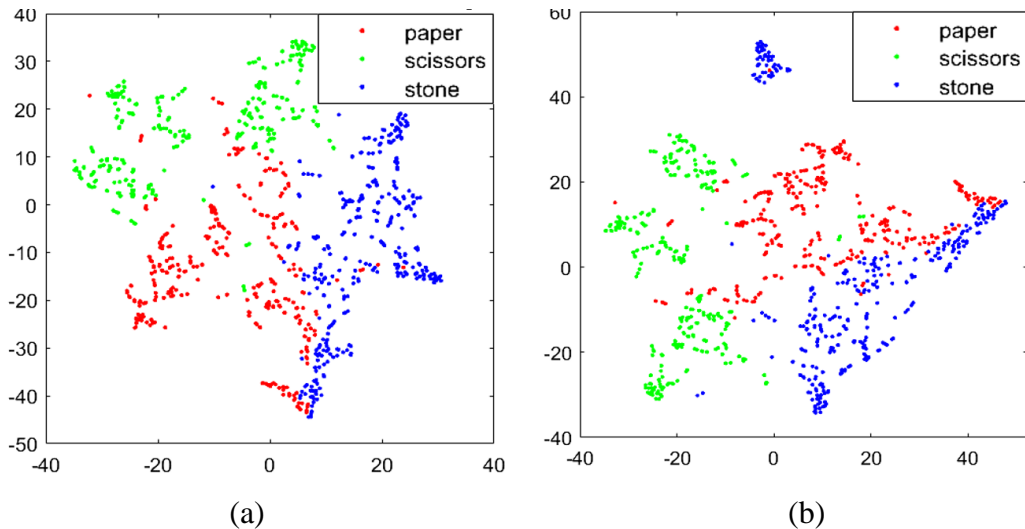


Figure 4.14: Two-dimension t-SEN plot of the RPS dataset. (a) The validation data distribution with a validation accuracy = 78.08% when the AlexNet is trained by the original small training data. (b) The validation data distribution with a validation accuracy = 87.8% when the AlexNet is trained by the original small training data and augmented data.

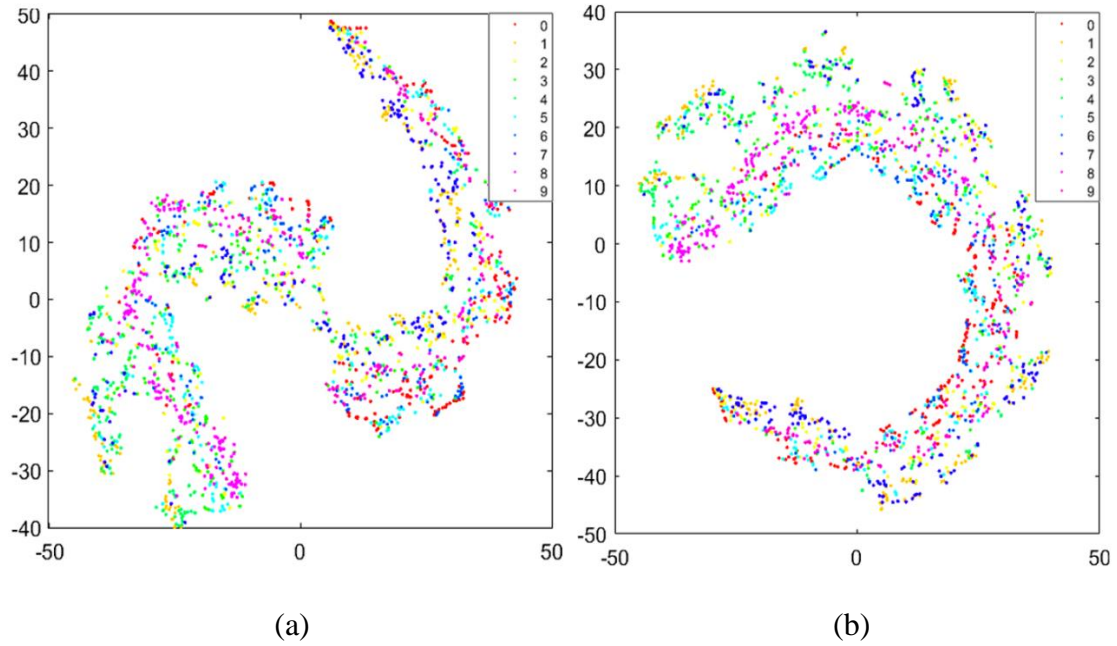


Figure 4.15: Two-dimension t-SEN plot of the MNIST dataset. (a) The validation data distribution with a validation accuracy = 71.59% when the AlexNet is trained by the original small training data. (b) The validation data distribution with a validation accuracy = 86.37% when the AlexNet is trained by the original small training data and augmented data.

#### 4.4.2 Comparison of Image Classification Accuracies with Traditional Image Augmentation Methods

An important performance is observed with our augmented method that can efficiently improve the classification accuracies. However, it is proven that traditional techniques of data augmentation are also able to increase the performance in image classification tasks. Therefore, to compare the difference between non-augmentation, our GAN-based method and usual data augmentation methods, the parameters were set in rotation and scaling with 20 training samples in each class, which are the very common augmentation methods in traditional augmentation, to check the final validation accuracy in image classification. Moreover, the same settings of traditional augmentation as our parameters were used in the transformation matrix of  $M'$ . The main reason for the parameter setting is to simplify the feature transformation to find out the differentiation between traditional augmentation and our method with the same parameter setting.

Table 4.8: Validation accuracy of 20 sample images per class for our method and traditional data augmentation on the MNIST dataset. (Unit: %)

<i>Dataset</i>	<i>MNIST</i>			
	<i>AlexNet</i>	<i>GoogLeNet</i>	<i>ResNet</i>	<i>VGGNet</i>
Non-augmentation	89.23	89.74	85.27	88.62
Tradition augmentation with 30 degrees rotation in 50% of training images per epoch	87.27	89.32	87.22	88.37
Ours with 30 degrees rotation of matrix $M'$	92.77	93.96	93.51	93.66
Tradition augmentation with 30 degrees rotation and 1.3 times scaling in 50% of training images per epoch	86.67	90.61	86.72	87.02
Ours with 30 degrees rotation and 1.3 times scaling of matrix $M'$	89.87	93.51	90.26	91.51

Table 4.9: Validation accuracy of 20 sample images per class for our method and traditional data augmentation on the RPS dataset. (Unit: %)

<i>Dataset</i>	<i>RPS</i>			
	<i>AlexNet</i>	<i>GoogLeNet</i>	<i>ResNet</i>	<i>VGGNet</i>
Non-augmentation	89.43	89.33	92.63	90.13
Tradition augmentation with 30 degrees rotation in 50% of training images per epoch	92.08	91.68	91.67	91.72
Ours with 30 degrees rotation of matrix $M'$	95.73	95.44	94.91	94.69
Tradition augmentation with 30 degrees rotation and 1.3 times scaling in 50% of training images per epoch	92.86	93.24	91.65	92.67
Ours with 30 degrees rotation and 1.3 times scaling of matrix $M'$	95.12	94.63	93.51	93.78

Additionally, in terms of image classification, the augmentation performance strongly relies on the generative similarity of the validation data, and different learning-based models usually involve many parameters to set, which easily leads to diverse results in image classification. To prevent the bias caused by the parameter setting among the learning-based models, simple transformation settings, rotation and scaling, were merely conducted in the experiment to evaluate the difference between the traditional augmentation methods and our proposed methods. Table 4.8 and Table 4.9 show the validation accuracy in different CNNs and augmentation methods. Based on the experimental results, traditional data augmentation methods are not obvious enough to entirely promote the performance of training with small datasets. What's worse, in some specific cases, such as the AlexNet on MNIST and ResNet on PRS, the traditional data augmentation cannot efficiently enhance the final performance depending on the parameter settings with only 20 images in each class. Consequently, our method can not only increase the validation accuracies among the four CNNs but keep the validation accuracies in a more stable range than the traditional augmentation methods when deep networks are conducted as classifiers for image classification applications.

## 4.5 Conclusion

In this chapter, a one-to-many image augmentation method is proposed, which adopts the convolution-based GAN architecture and the perturbation mechanism to generate realistic but diverse augmented images. Compared to the traditional augmentation techniques, the augmented images generated by the proposed GAN model offer more advantages to promote classification accuracies when a small number of training samples are used to train CNNs. Based on the analysis using confusion matrix, t-SEN plot, student's t-test, and accuracy in image classification, the experimental results demonstrated that the proposed model can mitigate the problem caused by labelled data scarcity, especially when a large number of images are impossible to be collected for training convolutional networks. Consequently, the proposed GAN framework for image data augmentation can significantly enhance the classification performance of DCNNs, which is beneficial in real applications where original training data is small and limited.



# Chapter 5

## Facial Image Synthesis from Small Training Data and Sparse Edge Features Using a GAN Framework based on One-to-one Image Mapping

### 5.1 Introduction

The motivation in this chapter is to augment a small image dataset by making use of conditional edge features extracted from the available training images, and it can be expected that the synthesised images are more diverse and less distorted than those obtained from traditional methods. As discussed in Section 2.3.1, traditional approaches to data augmentation include photometric transformations and geometric transformations, *i.e.*, translation, scaling, flip, rotation, noise adding, colour space shifting *etc.*, especially available for image data [188]. However, the data diversity introduced by traditional augmentation methods is limited and insufficient for many applications. To solve the limitations of traditional approaches, conditional inputs, such as edges, mark points, masks, semantic maps, labels and so on, can be used to make the synthetic images generated by GANs not only diverse but also controllable [189], [190]. Furthermore, one-to-one image translation methods using condition-based GANs can directly control the generated results by learning the pixel mapping relationship of paired images between conditions and real images [191].

Although one-to-one image translation methods based on conditional GANs have been developed for controllable image synthesis, there are still several problems that should be resolved when applying them to a small training dataset: 1) Compared with unconditional GANs, a limitation of condition-based GANs is that the output images must be generated from the corresponding conditional features, so a clear mapping relationship between inputs and outputs should be correctly established. Corresponding mapping relationships are hard to be discovered, especially when only a very small training dataset is available for deep neural networks to learn. 2) With a small training dataset, the training process for image-to-image translation is easy to converge but difficult to obtain high-quality results due to the overfitting and insufficient information about the underlying data distribution. Whether the used conditional features are of high

quality or not, the discriminator will overfit the training data [192], and the generator would produce unexpected distortions in the generated images in the validation or application phase [193], [194]. 3) Training GANs with a small dataset must deal with the inevitable problem of mode collapse [195], which implies that the GANs may learn the training data distribution only from a limited number of samples but overlook other useful training data [196]. Other training issues, such as non-convergence and instability, would also worsen the quality of the generated images [197]. Consequently, condition-based GANs are still difficult to synthesise photorealistic images merely relying on limited training data, such as incomplete conditional features and a small number of training images.

Edge-based image translation methods using condition-based GANs have the advantage of introducing diversity in image data augmentation. However, it is challenging in terms of generating high-quality photorealistic images because extracted edges cannot be regarded as the perfect conditional features to support various visual tasks and contain potential visual information of perceptual relevance [198]. Since edges, contrasted to other conditional features, generally contain incomplete information, such as unintegral geometry, simplified lines, discontinuous shapes, missing components, and undefined contours, it is hard for one-to-one translation methods to map edges into realistic images without clear conditional information.

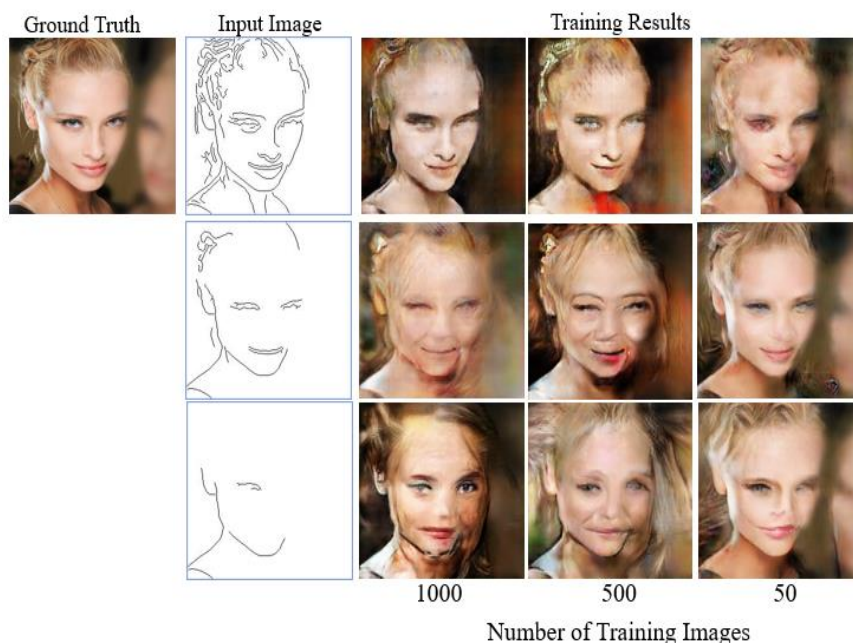


Figure 5.1: Examples of training results with a different number of training images and different conditional edge inputs by using the same parameter setting for the one-to-one translation method.

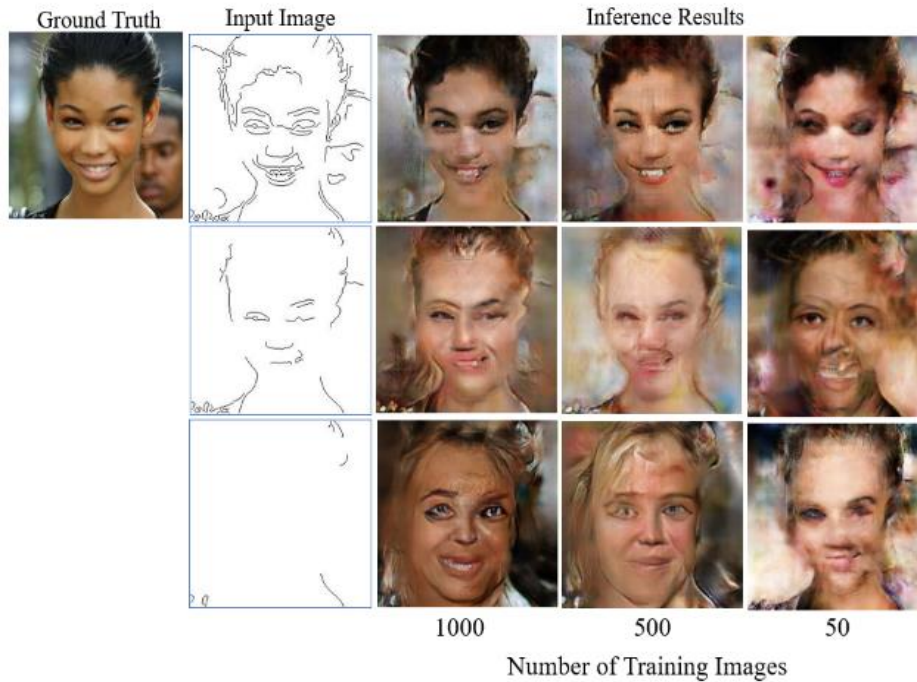


Figure 5.2: Examples of inference results with a different number of training images and different conditional edge inputs by using the same parameter setting for the one-to-one translation method.

To simply demonstrate the impact on the quality of the images generated by condition-based GANs, some preliminary experiments of the training results and inference results with a different number of training images and incomplete conditional edges are shown in Figure 5.1 and Figure 5.2 respectively. It can be observed that two important factors, the density of inputting conditional edges and the number of training images, have a considerable influence on the generative quality for both training and inference results, in which inference results are more critical for real applications because the edge features inputted in the inference process have not been learned during the training phase.

In this chapter, a new one-to-one image translation framework using condition-based GANs is proposed, which is expected to produce diverse and photorealistic images with fewer distortions when trained with a small number of training images. Instead of deepening the convolutional layers or increasing the number of parameters, the proposed condition-based GAN framework aims to learn additional relationships between incomplete edges and corresponding images; regional binarization and segmentation masks are used as new reference information, which can be automatically obtained by image processing. In particular, the proposed method can beneficially obtain extra mapping correlations between conditional edges and the corresponding ground truth images to mitigate negative impacts on inference results, such as synthetic

distortion, uncertainty and overfitting. If the condition-based GANs can efficiently learn from informative conditional inputs (*e.g.*, colour, texture, edges, labels, *etc.*), then it would be effortless to generate corresponding photorealistic outputs [199], [200]. A new GAN structure is proposed in this chapter, which divides the image synthesis task into two main stages: 1) The first stage transforms the conditional input of incomplete edges into refined images as the new conditional input. 2) In the second stage, the pixel values are processed by combining the information from segmentation masks and binarised images and then transforming the refined images into photorealistic image outputs.

The experimental results have demonstrated that the proposed GAN can generate diverse images of high quality even with a very small training dataset and the sparse unseen edge features as the conditional inputs. In addition, for data augmentation purposes, the proposed model efficiently mitigates large distortions easily caused by incomplete or untrained edge inputs. The contributions in Chapter 5 are as follows:

- To deal with the problem of distortions in one-to-one mapping images generated by GANs due to using limited training data, a network structure has been proposed for converting the original incomplete edges into new conditional features in a refined domain, in which distortions caused by small training data and incomplete conditional edges can be alleviated.
- For the first time, the proposed method uses the mixture of pixel values of both binary images and segmentation masks to enhance the conditional input in a new refined domain, which can integrate facial components, including eyes, nose, mouth, *etc.*, to introduce diversity and enhance the quality of the images generated by conditional GANs trained with a very small training dataset.
- A facial image augmentation method using conditional GANs has been proposed, which can generate photorealistic facial images of diversity from incomplete edges or hand-drawn sketches. Compared with traditional edge-to-image translation methods without ideal conditional inputs, the proposed method is tolerant to various incomplete edges as conditional inputs and able to generate diverse images of high quality in terms of Fréchet inception distance (FID) and kernel inception distance (KID).

## 5.2 Methods

One-to-one image translation methods are supposed to find specific mapping relationships between source distribution and target distribution. In general, a small

number of paired training features may not comprehensively align the data distribution with imperfect conditional inputs, such as incomplete edges and a small training dataset. Therefore, data refining between paired features can be processed to expand the mapping relationships based on unclear conditions and small training datasets. In this section, a new method with a very small training image dataset is proposed for facial image synthesis.

With incomplete conditional features in the source domain and small training data in the target domain, the mapping relationships between source and target domains cannot be clearly described by one-to-one image translation methods. The method proposed in this chapter transfers the source domain to an interim domain for refining images with extra annotated information, in which newly defined images in the interim domain need to be generated based on a small training dataset. This interim domain is designed able to provide extra reference information for discovering more precise mapping relationships between the source and target domain.

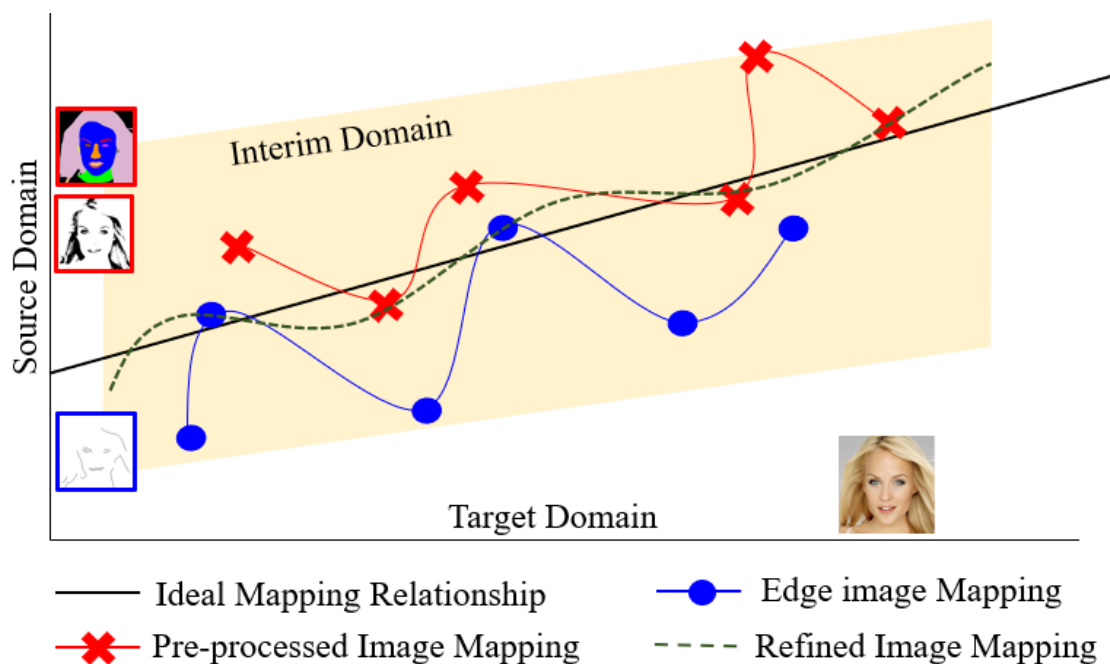


Figure 5.3: The proposed translation method by defining a refined domain based on a small training dataset. The GANs and image pre-processing are adopted to enrich the mapping relationships from the source domain to target domain.

Figure 5.3 shows the proposed translation method suitable for training with a small dataset. It is difficult for a small training dataset to contain a comprehensive view of correct mapping relationships between the source domain and target domain without sufficient representative training samples, as shown by the blue line in Figure 5.3. Even if changing the types of conditional inputs, a similar situation remains, and it is still

difficult to learn correct mapping relationships, as demonstrated by the red line in Figure 5.3. To comprehensively find correct mapping relationships, extending the mapping relationships with an interim domain and refined images, as shown by the green dotted line in Figure 5.3, can reduce uncertainty caused by using a small training dataset and incomplete edge features as conditional inputs to generate photorealistic as well as diverse results. More detail will be explained when introducing the proposed condition-based GAN framework later.

As training condition-based GANs with small training datasets, the following factors should be considered: 1) It is difficult to avoid distortions in the generated images and training imbalance with a small number of training images or insufficient samples. 2) Through convolutional neural network structures, such as convolution, normalisation and downsampling, it is easy to lose spatial information and impractical to completely preserve the conditional information from a small number of training images [201], [202]. If the conditional inputs contain sparse, unclear, limited, discontinuous, or incomplete features, fine-tuning model parameters without distortions becomes much more difficult. 3) The small training dataset and limited conditional features will make the training easy to overfit but hard to obtain realistic reference results. Since many parameters in a convolutional neural network need to be fine-tuned, it is impossible to optimise all the parameter values with a small number of training samples in terms of the generalisation capability of training a deep convolutional neural network. To tackle the above problems of using a small number of images to train a GAN, several training strategies are adopted in the proposed method, described as follows:

**Enlarging the diversity of source domain:** The GAN training using a small training dataset easily converges but frequently attains unrealistic inference results, mainly because of the overfitting. It is impossible to expect GANs to have a whole view of the target domain by merely training with a limited number of images. For the goal of achieving photorealistic results, both the discriminator and generator should stay in an equilibrium balance. Increasing the training data diversity and widening the mapping relationship between the source domain and target domain could help achieve the required balance between the generator and discriminator when a small number of training images are involved in training. In the proposed method, new reference information is created by image pre-processing, and the adoption of the interim domain for refining images can enlarge the data diversity in the source domain.

**Double translation:** Double translation aims to reduce the impact of the uncertainty due to using incomplete edges as conditional inputs. The double translation strategy not only decreases the chance of mode collapse compared to a single translation approach but also alleviates the distortions caused by training with a small number of

images. For generating additional reference information, the proposed method combines binary images and segmentation masks to produce refined images as conditional input in the second translation. Specifically, in the first translation, refined images with annotated facial components are generated from incomplete edge features. This translation is conducted between the source domain and the interim domain. On the other side, the second translation is conducted between the interim domain and target domain, which can successively learn from the possible distortions in the first stage to avoid or alleviate negative distortions in the final outputs.

**Reusing the conditional information:** Spatial information in conditional edges can be easily lost by training under convolutional neural layers, and the relationship between the source domain and target domain will become incomprehensive and unmapped. To reduce the spatial information vanishing, edge features in the source domain can be reused in each translation. Since the incomplete edge inputs contain useful conditional information for introducing the diversity of generated images, reusing the conditional information can keep the limited conditional inputs tight to reinforce the mapping relationships at each translation stage.

**Freezing weights:** Weight freezing is a strategy to overcome the gradient vanishing problem during training, which often happens when a small dataset is used as the training data. If the provided training data cannot give the discriminator enough information to progress the generator, the gradient will become smaller or close to zero when forwarding the gradient among the deep network layers. Incomplete conditions would worsen the gradient vanishing problem and make it impossible to fine-tune the model parameters well to obtain realistic results. Hence, freezing part of weights in separate training stages allows the discriminator to acquire useful gradients from each training stage rather than tuning all parameters at one time.

### 5.2.1 The Proposed Condition-based GAN Framework

To mitigate the output distortions led by using a small training dataset and incomplete edge as conditional input, additional paired segmentation masks and regional binary images are used as reference information in the proposed method, which can enrich the mapping relationships between the source domain and target domain. Consequently, the proposed method creates additional data distributions from the small training using image pre-processing, and the data in the interim domain provides more referable features than the original incomplete edges in the source domain.

Two U-nets [203], [204] are adopted in the proposed condition-based GAN framework for image-to-image translation. When training a condition-based GAN with

small training data, U-nets can achieve high performance for two reasons: 1) On one hand, during a training process, the U-net creates images based on the special concatenating structure, which is beneficial to retain the matched features from limited conditional features with an integral perception in convolutional layers. 2) On the other hand, the U-net structure is simple and advantageous to generate images without using very deep convolutional layers, which is critical for alleviating the gradient vanishing problem caused by training with small datasets and incomplete edges. The proposed framework also reuses the conditional input information to strengthen the input features at each training stage, and freezing weights for separate networks at each training stage can prevent gradient vanishing as well. To sum up, the proposed condition-based GAN framework can not only alleviate the negative influences on training with a small number of training images in the target domain but also intensify the meaningful conditional information in the source domain. An overview of the proposed condition-based GAN framework is shown in Figure 5.4 and described as follows.

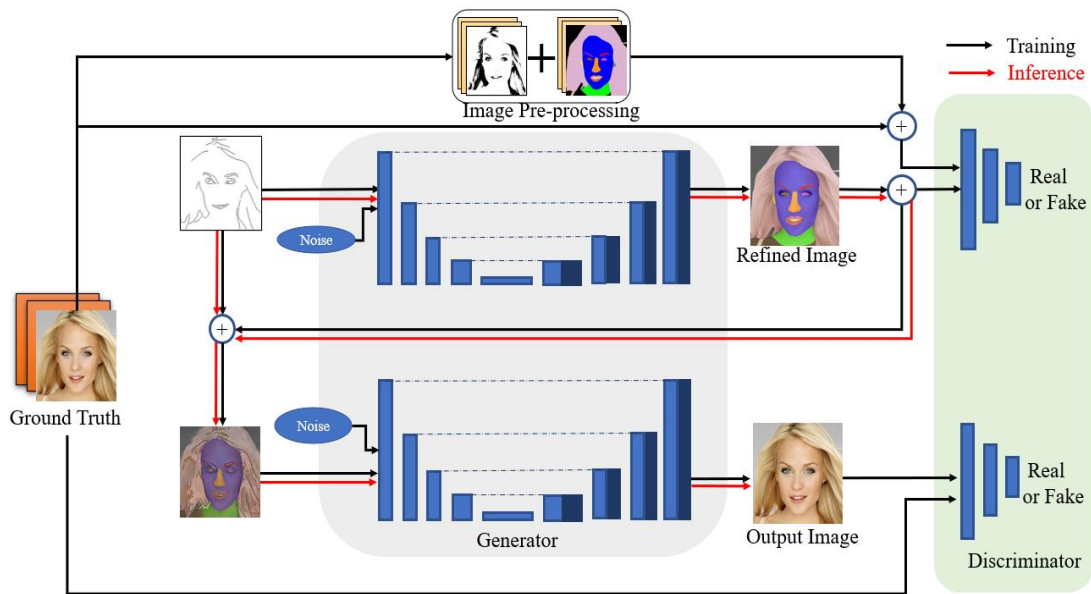


Figure 5.4: Overview of the proposed model for translating edges to photorealistic images using two U-nets.

The proposed model consists of three primary parts: 1) image pre-processing and refining, 2) generators and 3) discriminators. The two generators use the same convolutional structure of the U-net, both of which downsample and then upsample to the original size of input images [205], [206]. All convolutional layers use convolution kernels of size  $3 \times 3$ , and normalisation is applied to all convolutional layers except for the input and output layers. In the training phase, the first generator is used to create



refined images based on the original sparse edges and ground truth. The refined images are referred to the image pre-processing, which contains features related to texture, colour, and shape of different facial components. In addition, the second generator is designed to improve the synthetic process to generate photorealistic images from the interim domain. In the inference phase, the generators use fine-tuned parameters to generate photorealistic images from conditional edges that may have not been seen during training. The two discriminators have the same task of distinguishing between real and fake images; the first one is to identify generated images in terms of refined images, and the other is in terms of ground truth.

## 5.2.2 Image Pre-processing and Refining

Image refining is essential for providing informative conditional features since incomplete edges generally contain much unidentical information representing the same facial component. This uncertainty makes it difficult for condition-based GANs to comprehensively find pixel relevance between different domains. For instance, an unclear “black circle” with incomplete edges can represent either nose, ear or eye, even if using a powerful network, it is difficult to learn well with a rare sign of “circle” as a conditional input without any other crucial information (*e.g.*, colour, types, angles, positions, textures, sub-components, brightness, layouts, shapes, *etc.*). Moreover, there is no guarantee that ideal conditional inputs can be always obtained in real applications, especially if the conditional inputs are incomplete or sparse edges, in which these uncertainties commonly result in unexpected distortions. A refining process can be employed within an image-to-image translation method, which enhances the one-to-one mapping by providing close to ideal conditional inputs. If the refining images in the interim domain provide more specific mapping information, the synthetic quality will be correspondingly improved. Consequently, enhancing conditional information is one of the important goals for image pre-processing and refining.

## 5.2.3 Edge Extraction

Edges may contain incomplete features with many possible feature types, such as undefined density, shape, geometry and so on. However, to achieve high performance, one-to-one image translation methods need clear mapped conditions [207]. To generate photorealistic images from limited conditional information, extending translation relationships with proper reference information can make the mapping relationships

between the source domain and target domain more precise based on a small training dataset. As an example, the corresponding relationships among the ground truth, conditional features, and refined image are shown in Figure 5.5. Ground truth images are responsible for providing not only realistic features but also reference images to composite the refined images. The red boxes shown in Figure 5.5 indicate the eye mapping among different domains, and the new mapping relationships are expected to effectively reduce the mapping uncertainty in one-to-one image translation.

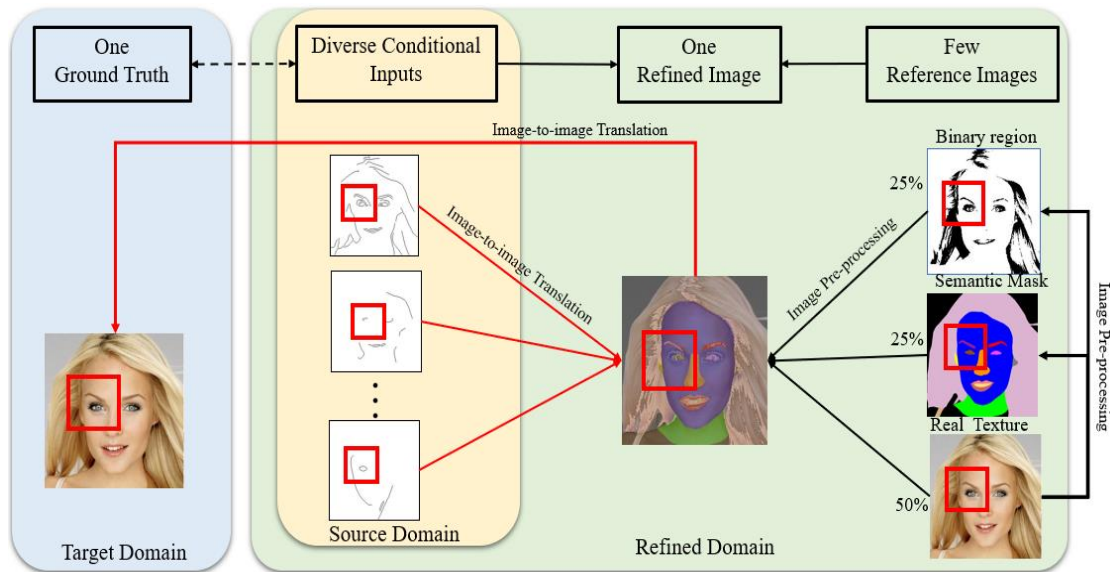


Figure 5.5: Corresponding mapping relationships among the conditional inputs, refined image and ground truth.

## 5.2.4 Adoption of Interim Domain

In contrast to directly transforming the source edges to target results, the proposed GAN framework first converts conditional edges to a refined interim domain. The interim domain reconstructs the incomplete conditional features using a U-net to get the possible missing information. Mode collapse and generative distortion problems may happen in the interim domain when incomplete edges are transferred to a refined image. Nevertheless, the translation at this stage is useful for facial image refining because the incomplete edges in the source domain are further processed. The refined images provide clearer accessorial information than the original incomplete edges, even if they are converted into simplified samples when mode collapse happens. By trial and error, regional features as reference images can efficiently reduce distortions and mismatches of generative features based on very sparse edges. Therefore, the refined

images are constructed by combining binarised images and segmentation masks, as shown in Figure 5.6. In short, the main function of the interim domain is to refine the original data distribution in the source domain for strengthening the incomplete conditional edge features.

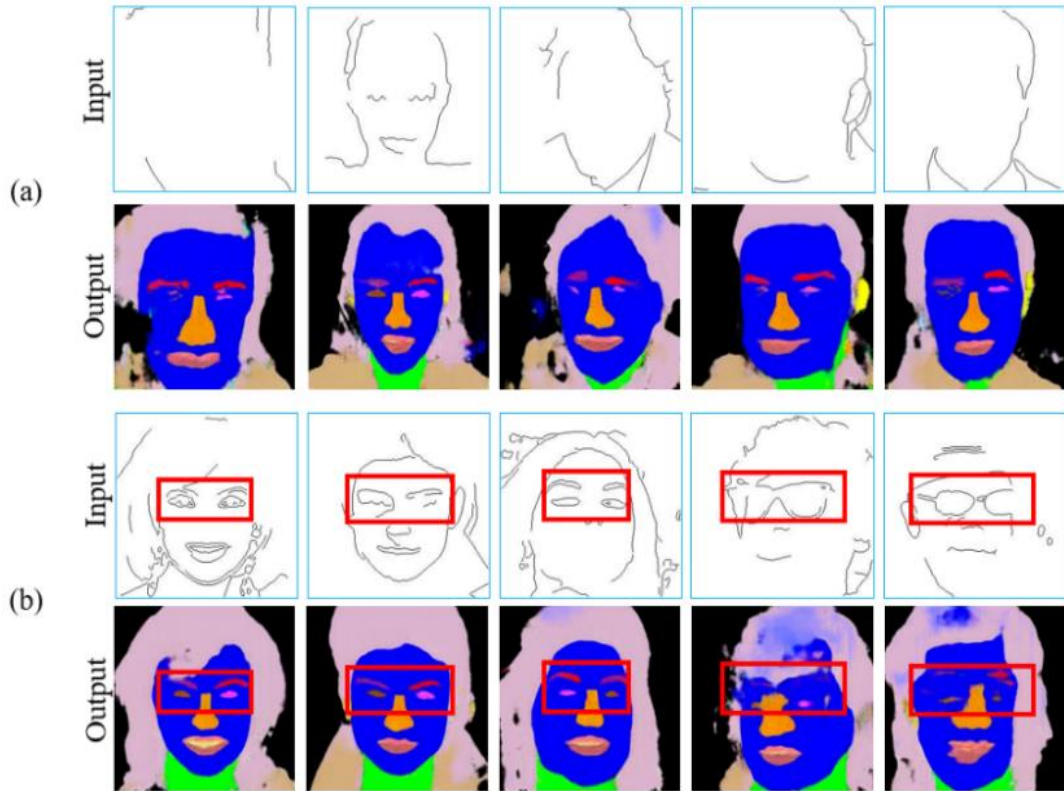


Figure 5.6: Inference results in translating sparse edges to labelled segmentation masks with 50 random training images. (a) The outputs can roughly resume the missing facial components from incomplete layouts when given abstract inputs. (b) The red boxes indicate the corresponding indefinite contours in the original inputs and generative masks.

Figure 5.6 (a) shows some inference results of using uncertain edges to generate segmentation masks from 50 paired training images. To handle the incomplete edges as the conditional inputs, facial components can be reconstructed by a U-net in an interim domain. The experimental results show that the proposed condition-based GAN can learn from only 50 segmentation masks to generate more integral face components, such as nose, eyebrow, hair and mouth. Figure 5.6 (b) illustrates examples where incorrect eye shapes are obtained, as shown in the red boxes, which would aggravate distortions in the target domain. What is worse, this situation is hard to be solved because it is difficult to increase the number of diverse samples based on a small dataset as GANs generally require more diverse data to be trained well. To resolve this problem,

additional binary images with clear regional information are additionally obtained through image pre-processing, which can enhance the contours and reduce uncertain distortions in facial components, as shown in Figure 5.5. In contrast to imprecisely depicting facial components in segmentation masks, binary images processed by appropriate thresholding can create more correct contours than segmentation masks and thus alleviate the problems caused by very limited training data.

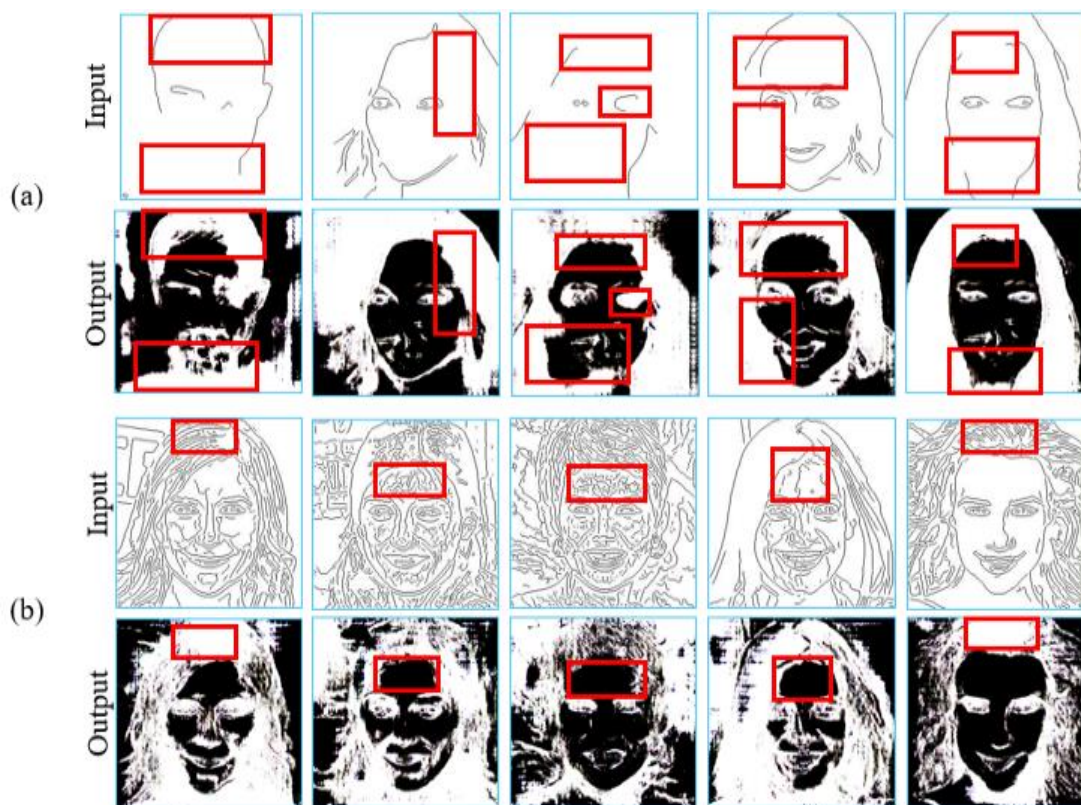


Figure 5.7: Inference results in translating sparse edges to binary regional images with 50 random training images. (a) The outputs integrate discontinued contours when given sparse inputs. (b) The outputs get rid of ‘bogus’ edges when given very dense inputs.

Figure 5.7 shows that binary images can handle the uncertain edge density in the inference phase to enhance crucial edge information with regional distributions. Binarised regional features can be extracted by the corresponding edge distribution from a small training dataset, which can not only integrate crucial contours, as shown in Figure 5.7 (a), but also get rid of meaningless noise if various untrained edges may be unrecognisable in the inference phase, as shown in Figure 5.7 (b). It is noteworthy that the results presented in Figure 5.6 and Figure 5.7 can be regarded as those from an ablation study, which shows that removing the component of combining binarised regional features in the proposed method will significantly deteriorate the performance

of the proposed condition-based GAN.

## 5.2.5 Model Training and Loss Functions

For training the proposed GAN framework, it is difficult to find a balance between the generator and discriminator, especially when there are very limited training data. Using an appropriate loss function is critical to ensure the good quality of the generated images. Firstly, to distinguish real images from fake ones, the following basic loss function is used between the two convolutional neural networks of generator and discriminator, which is treated as a conditional adversarial loss.

$$\begin{aligned} \mathcal{L}_{adv}(D, G) = & \mathbb{E}_{I,S}[\log D(S|I)] \\ & + \mathbb{E}_{I,I'}[\log(1 - D(I, G(I'|I)))] \end{aligned} \quad (5.1)$$

where the function employs the expected value  $\mathbb{E}$ , the generator  $G$ , the discriminator  $D$ , the source image  $S$ , the conditional edge feature input  $I$ , and the generated image  $I'$ . In the first U-net,  $S$  should contain a mixture of pixels of binary image, segmentation mask and ground truth to distinguish between real refined image and fake generated image. In the second U-net,  $S$  needs to be set as the ground truth only.

Secondly, inspired by the pix2pix GAN model, in which the  $L_1$  normalisation was used for achieving more realistic results than using  $L_2$  normalisation [166], the  $L_1$  normalisation is adopted as feature matching loss in the synthesised fake images. Since the paired images are used in the training phase, the  $L_1$  distance between the generated image ( $I'$ ) and source image ( $S$ ) can be defined as follows:

$$\mathcal{L}_{L_1}(G) = \mathbb{E}_{S,I,I'}[\|S - G(I')\|_1] \quad (5.2)$$

Finally, the main purpose of the loss function is to help the generator to synthesise photorealistic images by miniating the loss value with limited input conditional images. The overall loss function is defined as:

$$\min_G \max_D \mathcal{L}_{adv}(D, G) + \alpha \mathcal{L}_{L_1}(G) \quad (5.3)$$

where  $\alpha$  is the weight value of the loss function. A larger value of  $\alpha$  encourages the generator to synthesise images less blurry with  $L_1$  normalisation.

The second U-net uses the refined images and original sparse edges as inputs to generate photorealistic images with the same loss function but different training parameters and freezing weights. Another difference between these two networks is the source image  $S$ , which should be either the refined images or the ground truth images.

## 5.3 Experiments with the Proposed GAN Framework

### 5.3.1 Data Preparation

A small set of images, 50 training images randomly chosen from the dataset of CelebA-HD [208], formed the training samples in our experiments. CelebA-HD includes 30,000 high-resolution celebrity facial images. All the images were resized to  $256 \times 256$  in our proposed model. CelebA Mask-HQ [209] is a face image dataset consisting of 30,000 high-resolution face images of size  $512 \times 512$  and 19 mask classes, including skin, nose, eyes, eyebrows, ears, mouth, lip, hair, hat, eyeglass, earring, necklace, neck, cloth and so on. All the images in CelebA Mask-HQ were selected from the CelebA-HD dataset, and each image has segmentation masks of facial attributes corresponding to CelebA-HD.

Since different numbers of segmentation masks were used to compare the performance of different methods with different numbers of training samples, the CelebA Mask-HQ was used as the standard segmentation mask of reference images. If a very small training dataset is used, it would be fine to manually generate the segmentation masks by image pre-processing. In our experiments, the segmentation masks from CelebA Mask-HQ were used as the common reference images of the corresponding training images.

### 5.3.2 Implementation Details

The hyperparameter values were determined through trial and error because finding a balance between the generator and discriminator is still very challenging in GAN training, and optimisation with the hyperparameters needs to take plenty of time and computation capacities to affirm them. Therefore, the following parameters were the preliminary settings for training the proposed condition-based GAN framework: The Adam optimiser [147] was used to minimise the loss function with the initial learning rate set to 0.0002 and the momentum 0.5. The weight parameter  $\alpha$  in the loss

function was set to 100. All the experiments were conducted on a desktop computer with *NVIDIA GeForce RTX 2080 GPU*, *Intel Core i7-6700 (3.4 GHz)* processor, and 16G RAM.

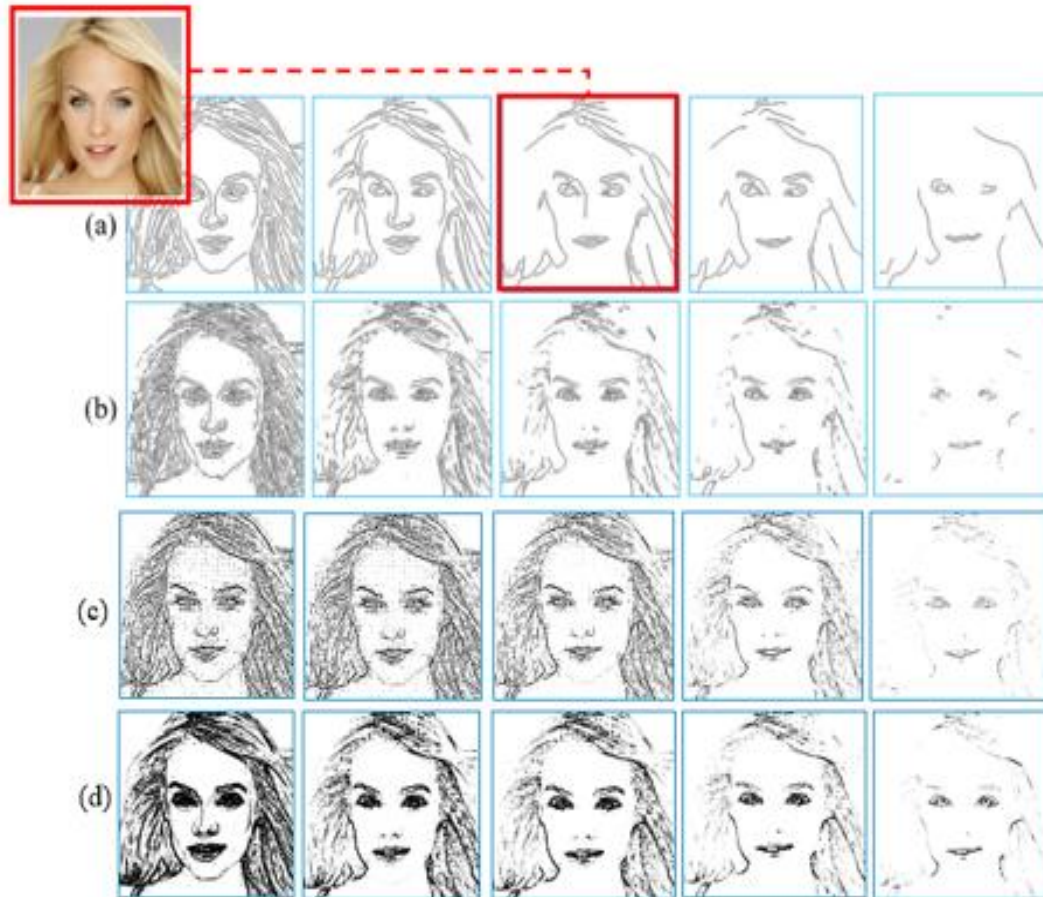


Figure 5.8: Comparison of different edge detectors: (a) results of Canny. (b) results of Sobel. (c) results of Laplace. (d) results of Gradient.

Incomplete edges or hand-drawn sketches usually represent abstract concepts of conditional inputs, which are beneficial for generating diverse results for data augmentation, but it is difficult for condition-based GANs to generate photorealistic images with limited conditional inputs based on small training data. In our experiments, edges extracted by the Canny edge detector [210] can produce simple and continuous lines from realistic images by setting intensity gradient values. The edges produced by the Canny edge detector are more similar to hand-drawn sketches than those by other commonly used edge detectors, as shown in Figure 5.8. Two intensity gradient magnitudes are used in the Canny edge detector as a parameter to control the edge density, which is determined by a threshold value in our experiments. The high-intensity gradient magnitude is set as a variable determined by the maximal threshold

value, and the low-intensity gradient magnitude of the low threshold value is 40% of the high threshold value. The threshold ratio in the Canny edge detector was appropriately chosen through trial and error in our experiment. The red box in Figure 5.8 shows the edges extracted with the threshold ratio setups, which contain clear information about facial components and meet the requirement of good conditional inputs without unexpected noise.

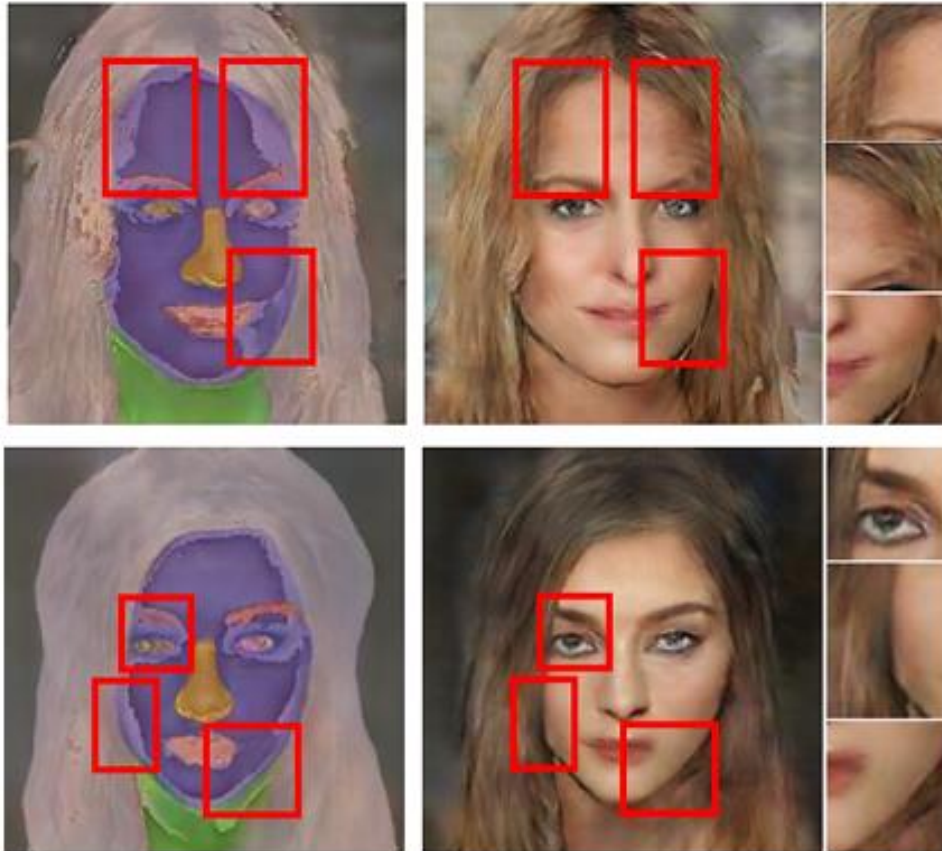


Figure 5.9: Inference results for refined images and final outputs. The red boxes represent blending areas in the refined region, which can be reflected by the brightness in the generated image outputs.

For the design of the interim domain, the pixel values of the refined image were set by the following mixture ratios: 25% from the binary image, 25% from the segmentation mask, and 50% from the original image. Figure 5.9 shows the inference results of the refined images and the corresponding generated image outputs. The red boxes represent blending areas in the masks, binary images and texture features between the refined images and output images, which reflect the brightness changes in the generated images. The overlapped regions are visually darker and gloomier compared to other regions. Therefore, these blending areas from different reference



images conduct transitions in brightness and lightness to synthesise realistic results. With the interim domain, the proposed condition-based GAN can efficiently deal with both overlapped and non-overlapped mappings between segmentation masks and binary regions, which lead to more photorealistic outputs.

Image blending with different styles is beneficial to diversely augment images. In the proposed condition-based GAN framework, generated images were controlled by conditional edge inputs. Exchanging or modifying edge features is an easy way to generate different images that increase data diversity and expand original facial features. Figure 5.10 shows an example to generate images with a small training dataset, in which the facial components are processed by exchanging edge features in conditional inputs. It can be seen that the generated images can preserve facial attributes with the swapped conditional edges and then reconstruct the incomplete or undefined input edges into clear as well as diverse results.

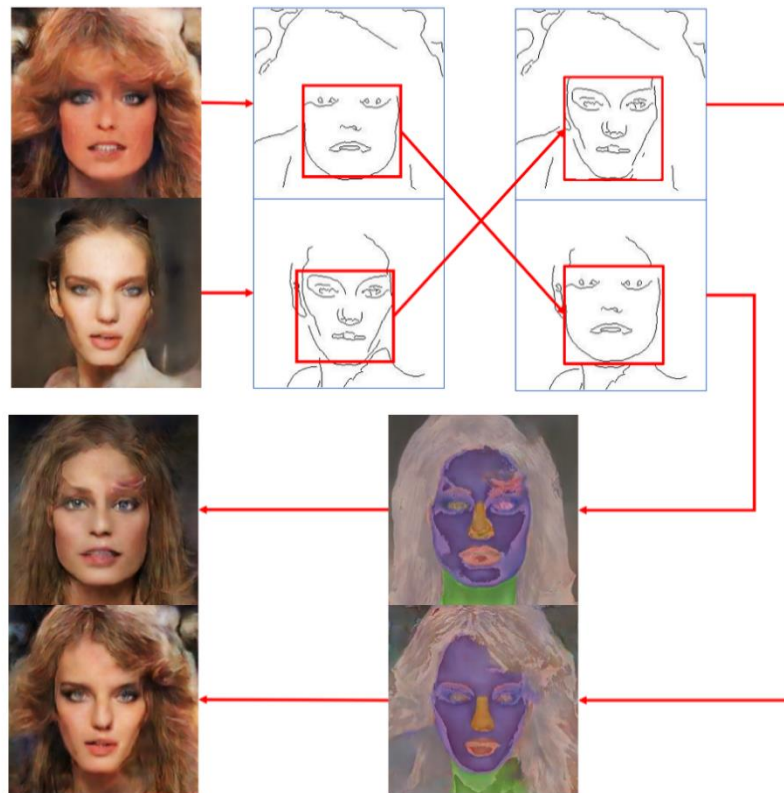


Figure 5.10: Synthesis results of exchanging conditional facial edges to generate diverse styles of facial images.

## 5.4 Results and Performance Evaluation

In this chapter, the proposed condition-based GAN framework was used to generate images with a different number of training images in the target domain and various settings of conditional edge inputs in the source domain. To demonstrate the performance of augmented data using the proposed method, some qualitative and quantitative approaches, including the visual inspection, FID score, KID score, human perception and image classification, are conducted as the metrics to evaluate the synthetic performance. In addition, state-of-the-art edge-to-image translation methods were used to compare the generative quality.

### 5.4.1 Diversity in Facial Image Augmentation Using the Proposed Condition-based GAN

It is clear that the threshold ratio values processed by the Canny edge detector design the density level of the extracted edges, in which the conditional inputs would affect the quality of images generated by the proposed condition-based GAN. It is desired that the condition-based GAN can generate diverse images with the change of edge density levels in the conditional input but be robust to the quality of the generated images. Figure 5.11 shows the inference results with different density levels in the conditional edges, which were not included in the training phase except for those in the red box. It can be seen that the generated images are slightly different with different density levels in the conditional edge inputs, and the distortions are small even when the GANs were trained using a small dataset of 50 training images. The generated images are more photorealistic if the conditional input contains less noise or unidentical edges, which correspond to those generated with the edge density level chosen in the training phase, as shown in the red box. Fortunately, with the change of density levels of the conditional edge inputs, the quality of the generated images is prevented from considerable deterioration because the refined images can integrally represent the facial features at an acceptable level based on a small dataset. Consequently, as the refined images are inputted to the second U-net in the second stage, the proposed interim domain plays an important role in reducing the distortions of the generative facial attributes.



Figure 5.11: Inference results in the source, interim, and target domains respectively. The various density levels in the conditional inputs are not in the training phase except the one in the red box generated by the Canny edge detector with the threshold value of 0.4. The results are from GANs trained using 50 images only.

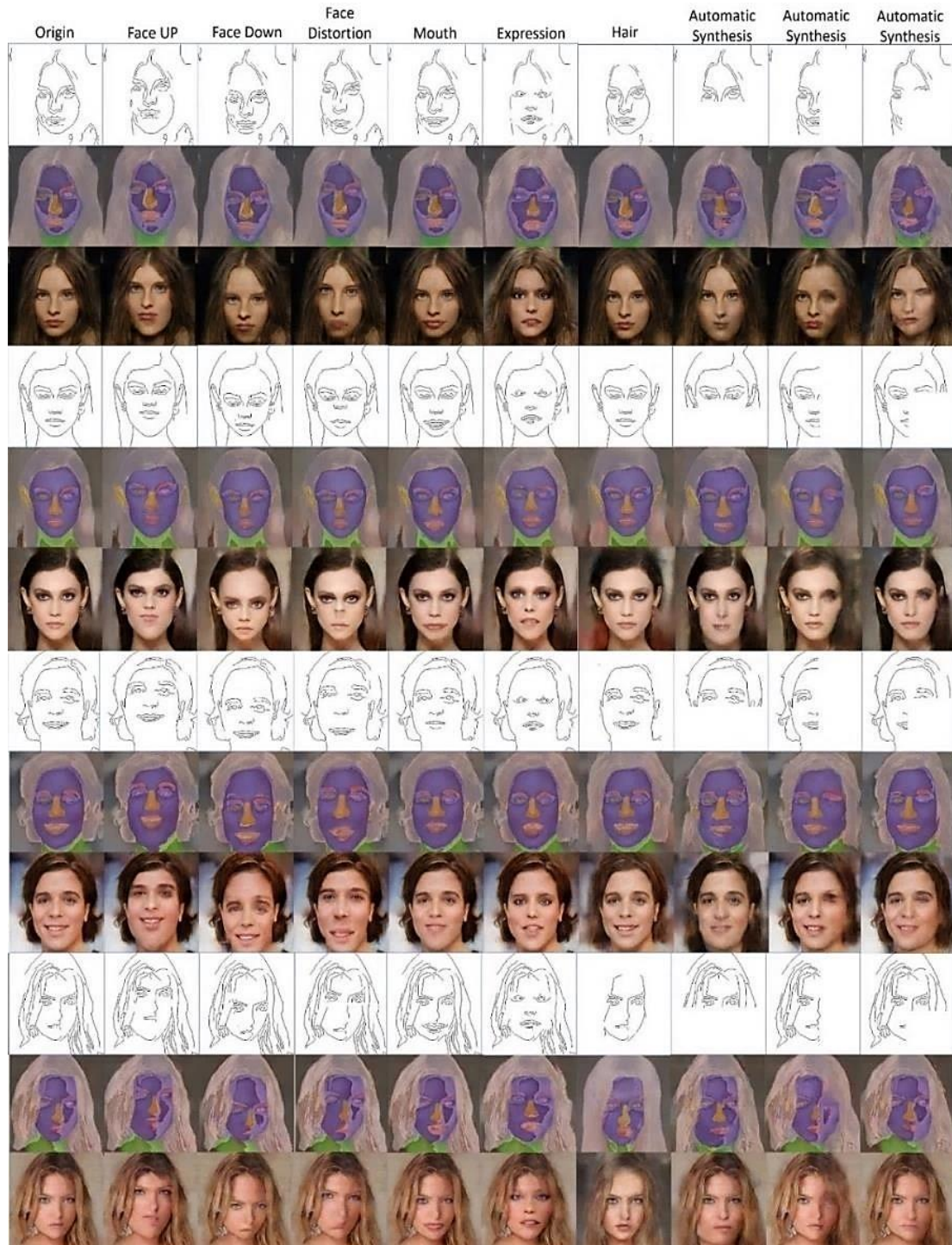


Figure 5.12: Examples of facial image augmentation results using 50 training images, with parts of input edges modified for introducing diversity to augment each training image with desirable facial features.

Figure 5.12 shows examples of facial image augmentation results using 50 training images to train the proposed condition-based GAN framework. Diverse facial images can be generated from each training image, in which the extracted edges are modified for desirable facial features as new conditional inputs. The modifications to the extracted edges include adding or deleting parts of the edges or changing facial expressions or directions, as shown in Figure 5.12. It can be seen that the image data augmentation results using the proposed condition-based GAN are more diverse than traditional augmentation methods, and the generated images are of good quality due to the use of the interim domain. For data augmentation purposes, deliberately modified edges as the conditional inputs make the proposed condition-based GAN framework able to boost the data diversity even with a small available set of training images.

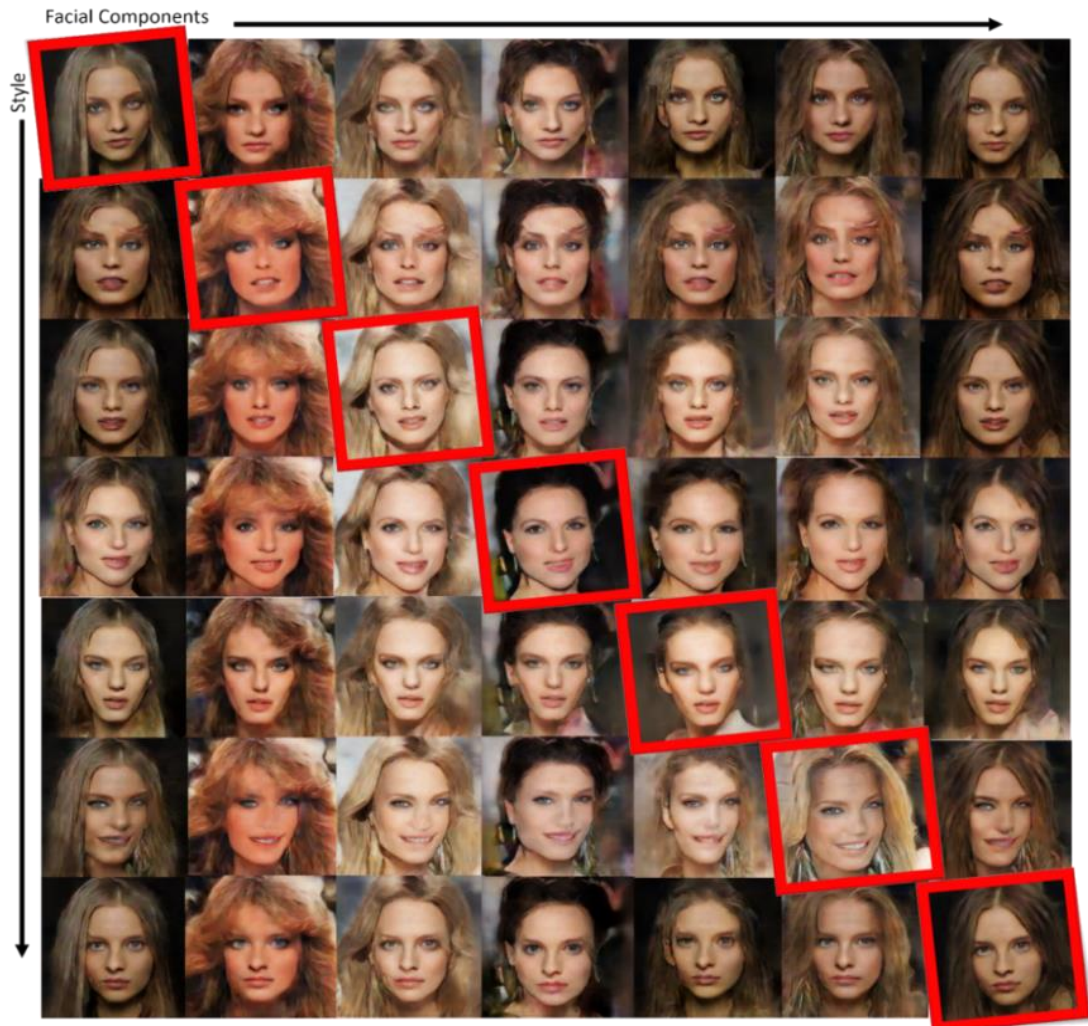


Figure 5.13: Examples of facial image augmentation results using 50 training images, with face components and hairstyles in different training images swapped in the edges as conditional inputs to generate diverse facial images.

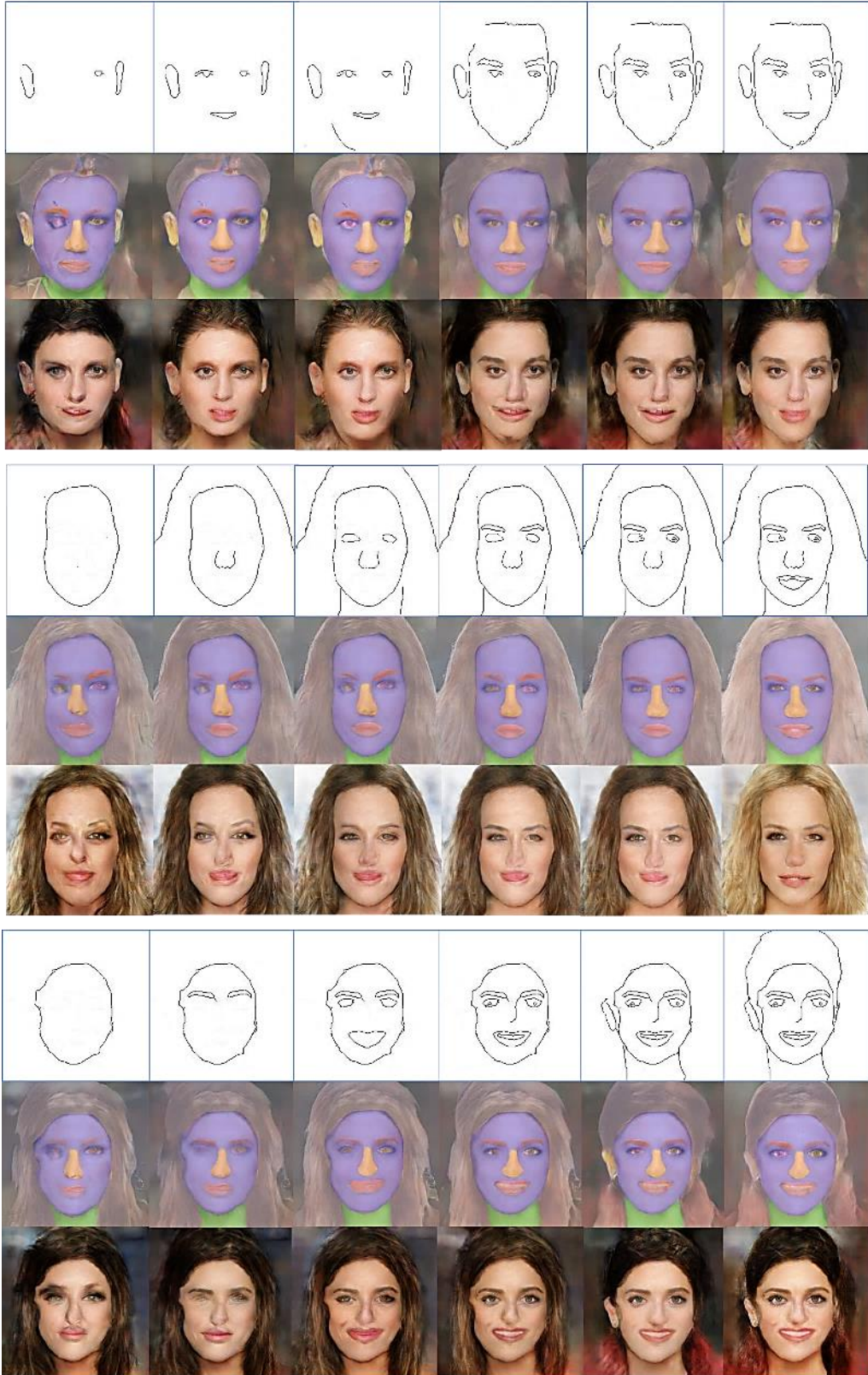


Figure 5.14: Inference results shown by images in the source, refined, and target domains respectively. The conditional inputs are hand-drawn sketches showing different facial expressions. The proposed GAN was trained using 50 training images.

Figure 5.13 shows some other augmentation examples of facial images using 50 training images to train the proposed condition-based GAN. The edge features are swapped from multiple training images as the new conditional inputs. The red boxes in the figure indicate the training images, and the other images in a row are generated by the proposed condition-based GAN, which shows the swapped facial features (including eyes, eyebrows, nose and mouth) and corresponding styles. It can be concluded that the proposed condition-based GAN, on the one hand, can efficiently keep the generated facial images of good quality. On the other hand, for data augmentation purposes, exchanging multiple edges as the new conditional images can improve the generative diversity with a small number of training images.

In general, it is difficult for one-to-one image translation methods to generate high-quality images if the conditional inputs do not directly correspond to features in the training images, such as untrained hand-drawn sketches. In previous experiments, it has been demonstrated that the interim domain is helpful to generate high-quality as well as diverse images with various edge density levels. In our experiments, hand-drawn sketches can be also used as the conditional inputs for the proposed condition-based GAN to generate photorealistic facial images with customer-designed edge features. Figure 5.14 shows the inference results with hand-drawn sketches as the conditional inputs, with the proposed condition-based GAN trained using a dataset of 50 training images. It is obvious when inputting unidentical or incomplete facial contours of the conditional inputs, the refined images generated by the first U-net in the proposed condition-based GAN structure are responsible for not only reducing the distortions of generated images but keeping the diverse facial attributes introduced by the hand-drawn sketches.

## 5.4.2 Qualitative Comparison

To evaluate the quality of synthetic results, the images generated by the proposed condition-based GAN were compared with those generated by the state-of-the-art edge-to-image translation methods, including pix2pix [166] and pix2pixHD [167], by inputting the same untrained edge conditions and in terms how the generated images are comparable to the ground truth images. Figure 5.15 shows a comparison of some representative images generated respectively by the three condition-based GANs, trained with the same small dataset of 50 training images. The sparse edges as conditional inputs were tested. The results in Figure 5.15 demonstrate that the proposed method can generate more photorealistic facial images with fewer distortions than pix2pix and pix2pixHD when the GANs were trained by a small number of training

images.

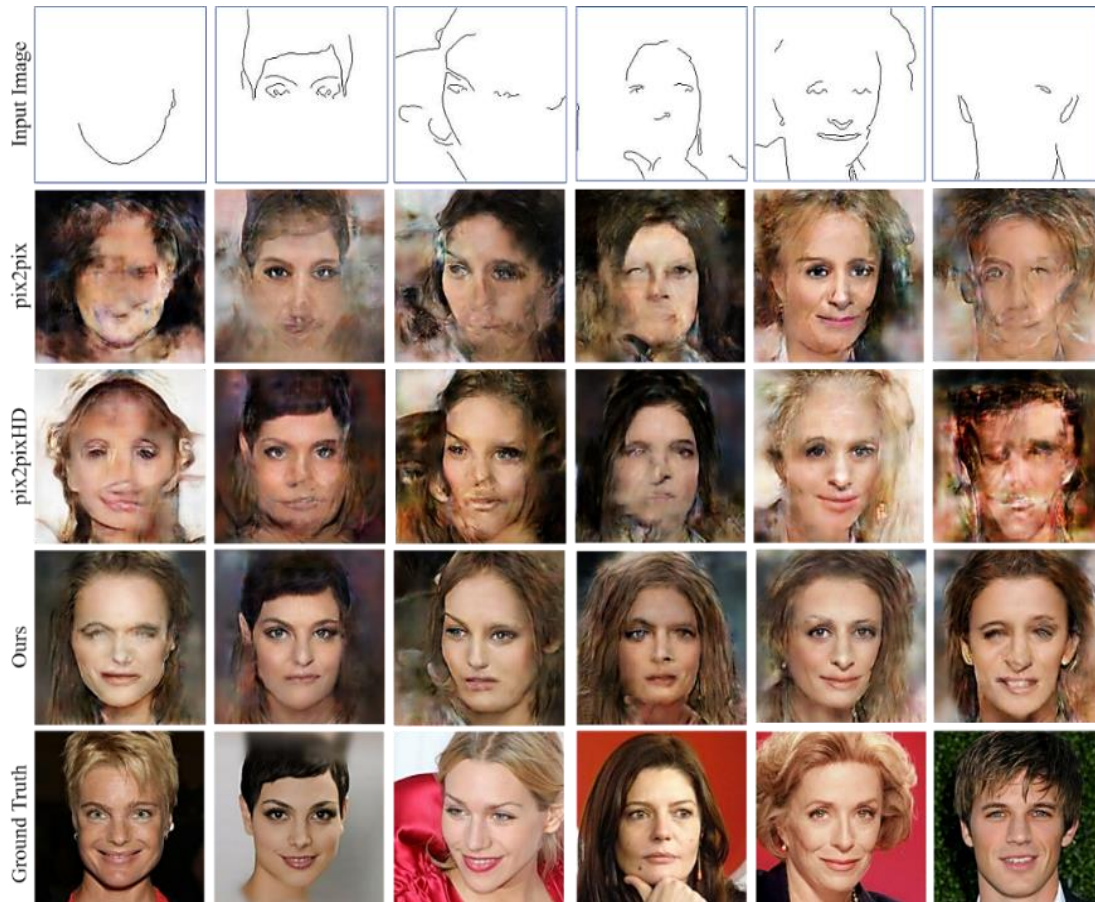


Figure 5.15: Inference results generated with sparse edge inputs (the first row), in comparison with those obtained from the state-of-the-art condition-based GANs. The images were generated respectively by the three GANs for comparison, trained using the same small dataset of 50 training images.

### 5.4.3 Quantitative Comparison

For data augmentation purposes, the proposed GAN model aims to generate photorealistic facial images and mitigate the distortions with very limited conditional features and a small set of training images. In this section, a series of evaluation experiments are designed to quantitatively evaluate the impacts on generated images, including the conditional edges, model architecture, number of training samples, human perception, and image classification performance. The setting detail and experimental result are described as follows.



### 5.4.3.1 Evaluation of the Influence of Conditional Edges

The quality of conditional edges make a huge influence on the generative results. Since features generated by the proposed GAN model should refer to the inputting conditions correspondingly, having clear and easily distinguished conditional edges can strengthen the generative quality. For influence evaluation on various conditional edges, the same 50 training images as the training data were respectively conducted with two setting types of edge density levels: one threshold value was 0.4, and the other threshold values were 0.2, 0.4 and 0.6. A set of 1,000 validation images with different types of edge density were used as the validation data for each conditional edge density. To comprehensively compare the impacts on different edge densities, the 11 types of edges density in the source domain, including the threshold value of 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9, were used to analyse the effects between the edge density in the source domain and synthetic quality in the target domain. Figure 5.16 illustrates samples of the conditional edges produced by the Canny edge extractor with different used threshold values.

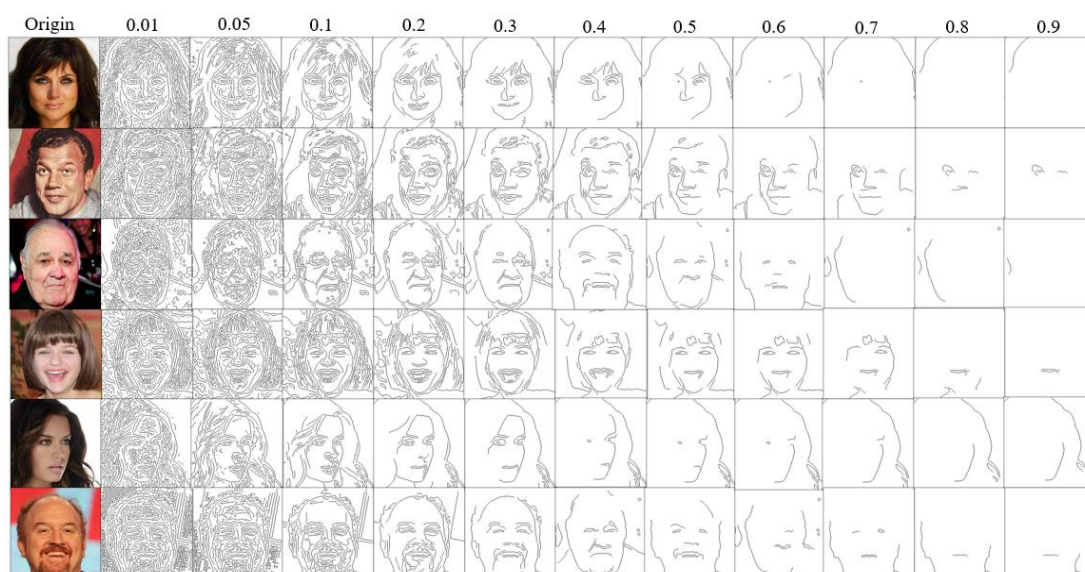


Figure 5.16: Samples of different threshold values using the Canny edge extractor.

For quantitatively comparing the performance of images generated by different conditional edges, FID and KID scores were adopted to evaluate the realistic scales of the generated images in this experiment. FID is widely used to evaluate the visual quality of generated images, which calculates the Wasserstein distance between the generated images and the corresponding ground truth images. Similar to FID, KID scores are based on an unbiased estimator with a cubic kernel. Lower FID and KID scores represent a better match between the generated images and the corresponding

ground truth images.

To evaluate the effectiveness of the generated quality produced by the proposed condition-based GAN structure, the validation data conducts the comprehensive conditional edges, which are extracted by the Canny edge detector with various threshold values. Each threshold value contains the same 1,000 validation images as the ground truth to create diverse edge conditions, and the generated images are compared with the ground truth for calculating the FID and KID scores. On one side, in the training phase, two types of edge thresholds were used as the training data: 1) Firstly, the training edge type was threshold value = 0.4 in the source domain along with 50 training images in the target domain. Secondly, three input settings with threshold values = 0.2, 0.4 and 0.6 were conducted with the same number of 50 training samples. 2) On the other side, in the inference phase, the comprehensive data of 11 different threshold values (0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9) were set to validate the performance of synthetic reality. Figure 5.17 separately shows the FID and KID scores of inference results at each threshold value.

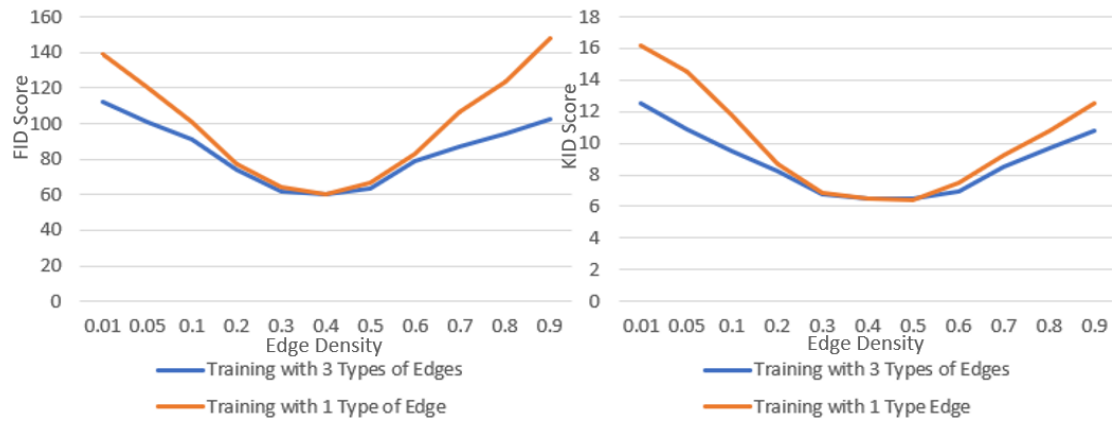


Figure 5.17: FID and KID scores with different levels of input edge density using the proposed model. One input type (high threshold = 0.4) and three input types (high threshold = 0.2, 0.4, 0.6) in the source domain were used during training with a small training dataset of 50 images. The FID and KID scores were evaluated based on the same 1,000 inference images for each edge density level from 0.01 to 0.9.

According to the experimental results shown in Figure 5.17, three conclusions can be made: 1) Firstly, compared to training with only one conditional type, training with three types of conditional edge density levels can achieve more robust performance than one type for reducing the negative effects on the uncertainty of inputting edges when various edge density levels are used as the validation data. 2) Secondly, in terms of the lowest FID and KID values of the two different training conditional types, both of the best generative qualities almost overlap at the threshold value of 0.4, which also

correspond to the training threshold value of conditional edge density. This overlapping phenomenon means that the edge densities for training the proposed model are a critical parameter to achieving good inference performance. In other words, good realistic results can be acquired when inputting edges are close to training conditional density. 3) Finally, the best performance of inference results among the edge inputs is between the threshold value of 0.3 to 0.5, and the ideal conditional edge setting is very close to the threshold value of 0.4, which value is also similar to the density used in the training. Compared with the other two threshold values of 0.2 and 0.6 also used to train the proposed model, the best realistic results appears at the training value of 0.4 rather than 0.2 or 0.6. It indicates choosing clear edges and appropriate density levels to form the conditional data in the source domain are key factors for the proposed model to generate photorealistic synthetic results.

### **5.4.3.2 Evaluation of the Usefulness of Interim Domain**

In this experiment, the use of the interim domain in our proposed double translation method with two U-nets was compared to a directive translation method with a single U-net. Figure 5.18 and Figure 5.19 present the comparative results between the single translation method and our proposed GAN model by using the interim domain. Followed with previous experiments, the same parameters of the edge densities in the source domain were continuously conducted as the default training data and validation data to evaluate the inference performance. The proposed model and the single translation model were compared under the same training parameters and validation data. Figure 5.18 shows the FID and KID values with 1 type of edge setting of the threshold = 0.4 between the single translation method and ours, and Figure 5.19 demonstrates the FID and KID values with 3 types of edge settings of threshold = 0.2, 0.4 and 0.6.

In Figure 5.18 and Figure 5.19, it can be found from the experimental results that the proposed GAN framework with the interim domain has a significant influence on generative reality, and two major points can be concluded: 1) Firstly, our proposed model with the interim domain can improve the synthetic quality with a flatter curve and lower values on KID and FID scores, which significantly mitigate the generative distortions compared to the single translation method. 2) In addition, according to the generative performance, the proposed model with double translation is more robust than the U-net to deal with the deficiency of the untrained edge densities when extremely sparse and dense edges away from the training densities are employed. For instance, referring to the generative quality in the threshold value of 0.01 and 0.9, the inference results generated by the proposed model outperform the directive translation method.

Consequently, based on the overall experimental results, the proposed GAN model achieves lower FID and KID values than the single translation model of U-net, which proves the use of an interim domain in the proposed condition-based GAN can not only reduce distortions caused by incomplete conditional edges but also improve the realistic quality of the generated images when a small number of images are involved in training.

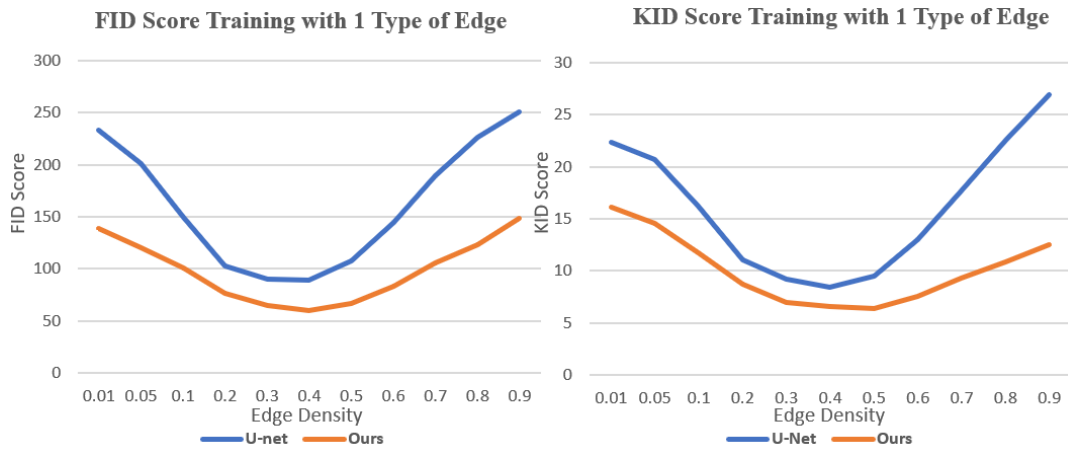


Figure 5.18: FID and KID scores of double U-nets with refined domain and single U-net with one input type (high threshold = 0.4) in the source domain, where a small training dataset of 50 images was used during training. The FID and KID scores were calculated based on the same 1,000 inference images at different edge density levels.



Figure 5.19: FID and KID scores of double U-nets with refined domain and single U-net with three input types (high threshold = 0.2, 0.4, 0.6) in the source domain, where a small training dataset of 50 images was used during training. The FID and KID scores were calculated based on the same 1,000 inference images at different edge density levels.

### 5.4.3.3 Evaluation of the Impact of the Number of Training Samples

The proposed model aims to generate photorealistic facial images using condition-based GANs trained with a small set of training images for data augmentation. To evaluate the effects on the number of training images and compare the generative quality of our proposed condition-based GAN with pix2pix and pix2pixHD, different numbers of training images (25, 50, 100, and 500) were used to train each of the three condition-based GANs separately. Moreover, to demonstrate the effects of different conditional edge density levels, both sparse edges (threshold ratio = 0.4) and dense edges (threshold ratio = 0.2) were used to generate 1,000 images by each trained condition-based GAN. The FID and KID scores of the images generated by the three condition-based GANs were calculated respectively.

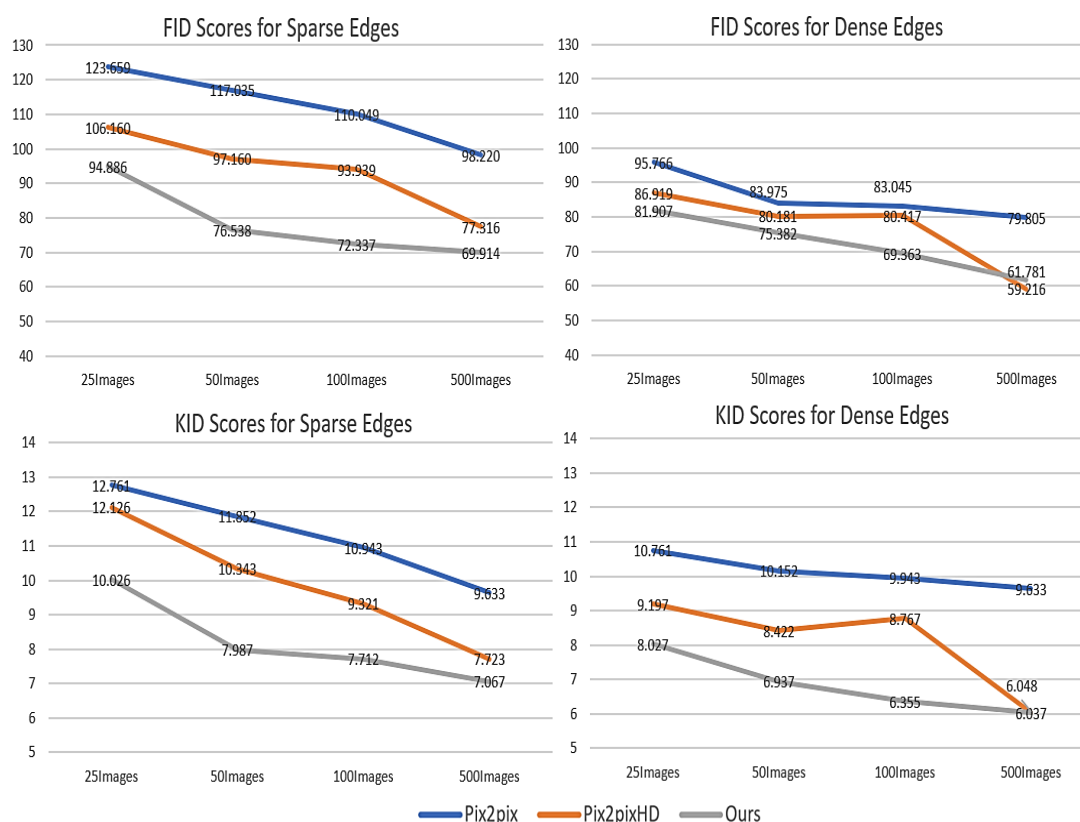


Figure 5.20: Changes in FID scores (first row) and KID scores (second row) with a different number of training images. Comparison among three edge-to-image translation methods with sparse and dense edge inputs respectively: pix2pix, pix2pixHD and ours.

Figure 5.20 shows the changes in FID and KID scores with the different numbers of training images, from which the following three points can be made: 1) The proposed condition-based GAN achieves lower FID and KID scores than pix2pix and pix2pixHD when trained with the same number of training images. 2) Dense conditional edges achieve lower KID and FID scores than sparse edges, but the diversity in the generated images may be constrained. 3) With the increase in the number of training images, the advantage of the proposed method over the existing methods becomes less obvious. This tendency indicates that the proposed condition-based GAN framework is very effective when it is trained with a small number of training samples, and its performance would approach that of the existing methods when the number of training samples becomes relatively large.

#### **5.4.3.4 Evaluation by Human Perception**

To further evaluate the proposed method, randomly selected sparse edges were used to generate images in the inference phase by the three methods: pix2pix, pix2pixHD and ours. In our analysis, the method of pairwise comparison (also known as paired comparison analysis) is adopted to evaluate the generative quality between ours and different competing models, in which the voters submit their preference by choosing from many paired options, and the total percentage is calculated to represent the relative importance between the compared entities. In this experiment, the generated images were arranged randomly in pairs, ours vs. pix2pix, or ours vs. pix2pixHD, and presented together with the corresponding ground truth images to human participants. The participants were asked to choose which image in each pair is more photorealistic. Google Forms were used for this evaluation with 100 pairs of generated images, where the participants were required to select a better one between the paired images visually and complete the task with 100 image pairs within 15 minutes. Both postgraduate and undergraduate students, aged from 18 to 28, in the School of Computer Science and Electronic Engineering (CSEE) at Essex University, were invited to take part in this evaluation, and 112 effective responses were received. Based on these received responses, the percentage of preference for the generated images was calculated in terms of their photorealistic quality. Table 5.1 shows the results of the evaluation by human perception, which indicate that 86% of the participants preferred the images generated by our method over those generated by pix2pix, and 78% preferred images generated by our method over those generated by pix2pixHD.

Table 5.1: Results from user preference study. The percentage indicates the users who favour the results of our proposed method over the competing method.

	<i>Ours vs Pix2pix</i>	<i>Ours vs Pix2pixHD</i>
<i>Preference</i>	86 %	78 %

### 5.4.3.5 Evaluation by Balanced Image Classification

According to the results shown in the previous section, the images generated by the proposed model achieve high preference in human perception. To further evaluate the augmentation performance of adding these synthetic images into original datasets as the newly augmented dataset, an image classification task as the evaluation metric was designed in this experiment, where gender recognition based on two classes (male and female) was conducted for the image classification task. A different number of training images per class were randomly chosen as the small training dataset and these chosen images were also used as the training samples in the proposed model. For instance, if there are 15 images in each class are used for image classification tasks, the same 15 images in each class are the training samples for the proposed model to generate many diverse images, and the generated images will be added back to the original 15 images as the new augmented datasets. All the validation data were taken from the CelebA dataset, and each class used 2,000 images as the validation data to calculate the validation accuracy in image classification. The methods of transfer learning were employed to access whether the augmented images can improve the performance of image classification. Four convolutional neural networks (CNNs), including AlexNet, GoogLeNet, VGGNet and ResNet, were taken as the classifier of transfer learning methods. In particular, the same input edges of conditional features were commonly used in both male and female classes, and different numbers of augmented images, including 50, 100, 500 and 1,000 synthetic images were added to the original training data for evaluating the validation accuracies of image classification.

Figure 5.21 illustrates some inference samples from the same input conditional data and the corresponding synthetic results in the male and female classes separately. All the results were generated from 15 training images used for data augmentation. Additionally, Table 5.2 shows the validation accuracies of CNNs, where a different number of original images and augmented images per class are used. The same number of augmented images, including 50, 100, 500 and 1,000 images, were added to both female and male classes.

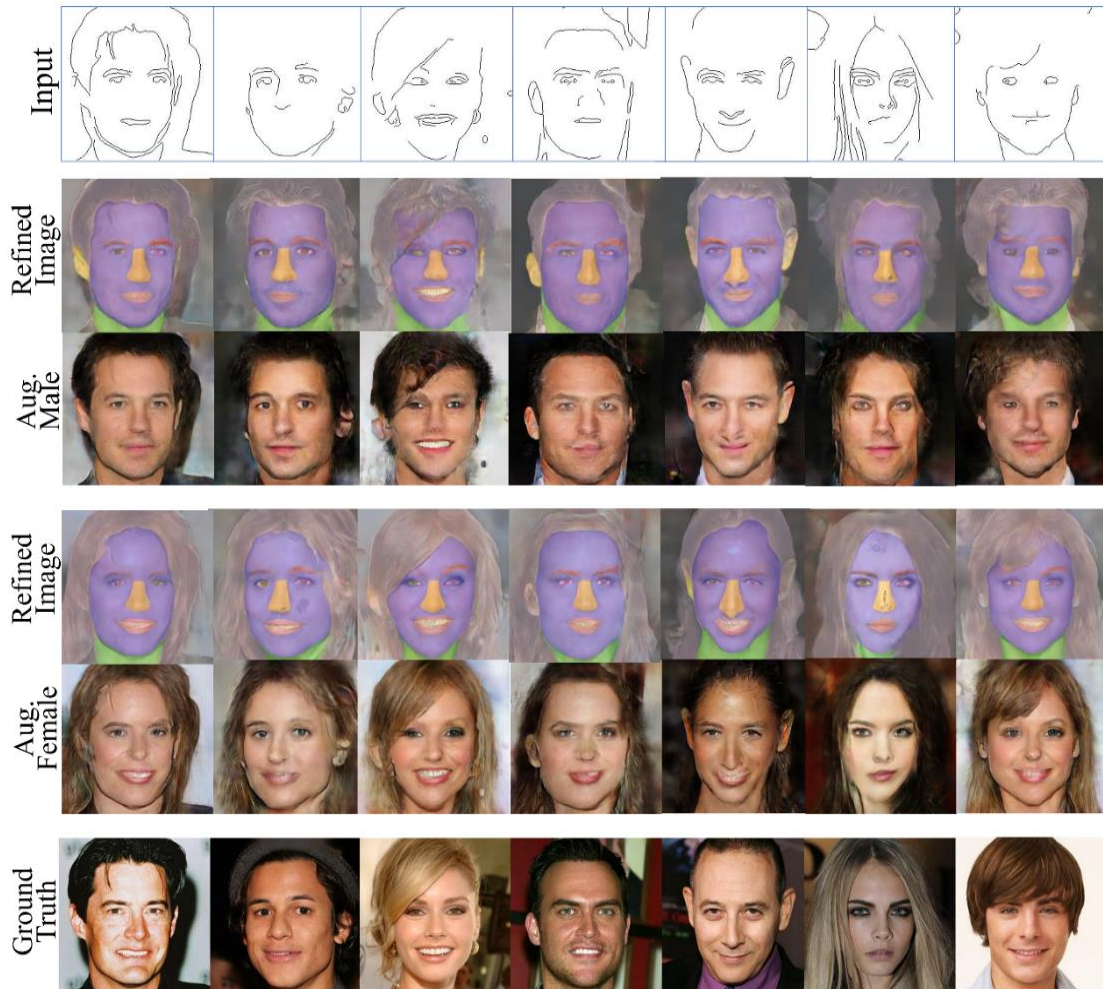


Figure 5.21: Samples of inference results (second and third row) generated from the classes of male and female separately (bottom row), where the results are generated with 15 training images by inputting the same sparse edges (top row).

As seen in Table 5.2, besides the conclusion that the augmented data generated by the proposed model can efficiently promote the validation accuracies in a gender classification task, additional conclusions can be obtained from the experimental results: Firstly, as the number of augmented images increased by 50 to 100 per class, the validation accuracies are promoted compared to those without data augmentation. Secondly, when the number of augmented images sets from 100 to 500, it is the preferred data amount of synthetic images to efficiently promote classification performance. Based on the experimental results of adding 100 to 500 augmented images, the validation accuracies, shown in Table 5.2, generally stays at a relatively stable level. Finally, the enhancement of validation accuracies becomes unobvious in most cases when the number of augmented images is added by more than 500 synthetic images. The main reason for this situation is that the CNNs have already learned sufficient information from the augmented features. If no extra useful representations or new data



can be discovered, no significant improvement will be obtained despite more augmented images being added to the original data. Consequently, to find a balance between performance enhancement and computing efficiency, having the appropriate number of augmented images generated by the proposed model can not only boost the classification accuracies but also reduce the training cost to discover an optimisation.

Table 5.2: Validation accuracies of CNNs trained with different numbers of original images and augmented images per class.

<i>No. of Training Images per Class</i>	<i>No. of Augmented Images per Class</i>	<i>GoogLeNet</i>	<i>AlexNet</i>	<i>VGGNet</i>	<i>ResNet</i>
5	0	76.05%	71.68%	74.61%	73.35%
	50	88.07%	78.03%	83.63%	80.97%
	100	90.16%	81.35%	85.35%	89.63%
	500	91.52%	87.47%	83.99%	87.72%
	1,000	91.40%	87.58%	90.94%	80.49%
10	0	86.18%	78.69%	74.36%	80.46%
	50	87.59%	85.49%	90.64%	89.64%
	100	91.45%	87.61%	74.95%	91.27%
	500	91.24%	86.60%	91.49%	90.10%
	1,000	93.07%	89.96%	82.08%	91.38%
15	0	88.09%	78.87%	84.82%	85.90%
	50	90.65%	79.96%	87.14%	87.75%
	100	91.76%	82.63%	86.47%	91.87%
	500	91.86%	88.84%	89.97%	91.36%
	1,000	92.97%	88.60%	90.16%	92.35%

#### 5.4.3.6 Evaluation by Imbalanced Image Classification

In most medical diagnosis cases, the class imbalance is a common challenge for deep networks to achieve good classification performance. In this section, to evaluate

the influence of the proposed GAN model on imbalanced image classification, a reduced magnetic resonance imaging (MRI) dataset [211], [212] was used as an imbalanced dataset, and the proposed GAN model was applied to generate augmented images from a small number of training images in the minority class for comparing the classification performance with and without using augmented image data for training deep neural networks.

In the reduced MRI dataset, two classes, normal and abnormal, were adopted in our experiment. To create training datasets with different imbalance ratios, 20 images were randomly picked from the abnormal class whilst different numbers of normal images were randomly picked, with 20, 100, 300, and 500 tested separately. The validation dataset consists of 1500 images per class, which are different from any picked training samples. Transfer learning was conducted with training datasets of different imbalance ratios respectively to train four CNNs: AlexNet, GoogLeNet, VGGNet and ResNet. The validation accuracies of the 4 CNNs trained by datasets of different imbalance ratios are shown in Table 5.3, where the validation accuracies are values averaged over the 10 best classification results to make sure that the obtained validation accuracies were close to the ideal fitting in real data distribution for mitigating the overfitting problem. It can be seen from the experimental results that classification performance drops as the imbalance ratio increases.

Table 5.3: Validation accuracies of CNNs trained with different numbers of normal images in the imbalanced training dataset. (Unit: %)

<i>No. of training images (abnormal/ normal)</i>	<i>AlexNet</i>	<i>GoogLeNet</i>	<i>VGGNet</i>	<i>ResNet</i>	<i>Avg.</i>
20/20	67.39	69.42	72.51	71.95	70.32
20/100	55.27	57.96	57.60	61.31	58.04
20/300	52.93	56.13	53.87	54.22	54.29
20/500	50.85	54.61	51.64	53.26	<b><u>52.59</u></b>

A further experiment using the proposed GAN models for image augmentation to enhance the classification performance of CNNs was conducted using the imbalanced training dataset with 20 abnormal and 500 normal images, based on which the CNNs achieved an average accuracy of 52.59% only without using augmented data.

The training dataset for the proposed GAN model consists of 20 images in the abnormal class containing brain tumour features, and additional 20 mask images in the interim domain were generated from the training data to specify the tumour sizes and locations. The edges as conditional input were extracted from normal images. The proposed GAN model can learn the mapping relationship between the extracted edges and the real images with tumour features, which makes it possible to transfer the edges extracted from the normal class into many augmented images containing tumour features and thus increases the number of images in the minority class. This is similar to oversampling an imbalanced dataset, leading to a balanced dataset for training CNNs for brain tumour detection. Figure 5.22 shows some examples of the inference results from the trained GAN, with normal images, extracted edges, refined images and augmented images in comparison.

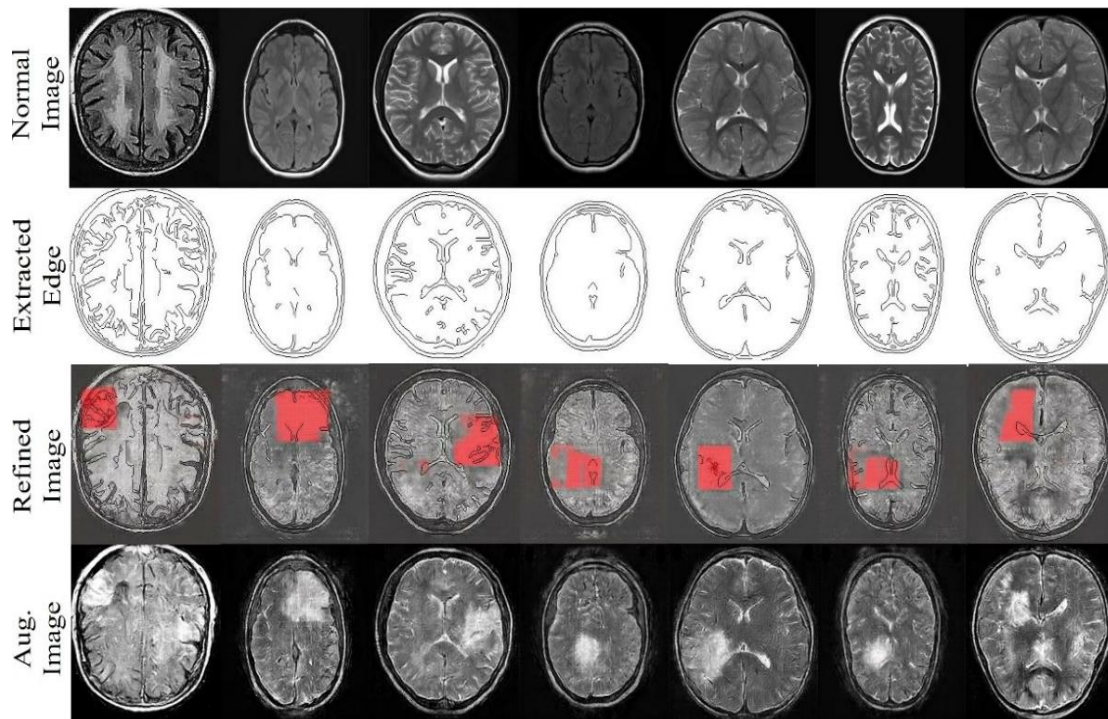


Figure 5.22: Samples of inference results (bottom row) from the proposed GAN, trained by 20 images in the abnormal class, where the input edges (second row) are extracted from the normal images (top row).

The validation accuracies of CNNs trained with and without using augmented images are demonstrated in Table 5.4. Similarly, as in Table 5.3, the validation accuracies are values averaged over the 10 best classification results of each CNN. It can be observed that the validation accuracies can be significantly improved when the augmented images generated by the proposed GAN model are used to enlarge the

amount of training data and the diversity in the minority class.

Table 5.4: Comparison of validation accuracies of CNNs trained with and without using augmented images. (Unit: %)

<i>No. of training images (abnormal/ normal)</i>	<i>AlexNet</i>	<i>GoogLeNet</i>	<i>VGGNet</i>	<i>ResNet</i>	<i>Avg.</i>
20/500	50.85	54.61	51.64	53.26	52.59
520 (20 & GAN) /500	62.32	<b><u>61.77</u></b>	65.81	69.26	64.79

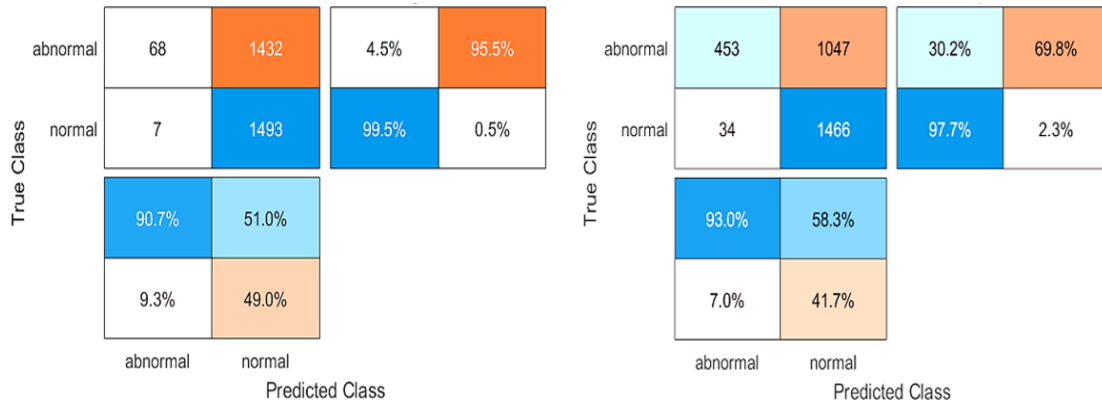


Figure 5.23: Comparison of confusion matrices of the GoogLeNet trained with and without using augmented images: The left column shows the results without using augmented images and the right column shows the results using augmented images.

For imbalanced image classification, accuracy may not be a good performance metric. As GoogLeNet performed the worst in this experiment, it was further analysed by the confusion matrix and other performance metrics. Figure 5.23 respectively illustrates the confusion matrices of the GoogLeNet trained with and without using augmented images. Although the performance improvement by using augmented images for training is significant, it can be seen that the false positive rates are quite high, which could be due to overfitting as a result of using a very small training dataset. To further evaluate the effectiveness of using image augmentation to improve classification performance with imbalanced original datasets, accuracy, precision, recall and  $F_1$  were calculated based on the confusion matrix as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5.5)$$

$$Recall = \frac{TP}{TP + FN} \quad (5.6)$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5.7)$$

where TP, FP, TN and FN represent true positive, false positive, true negative and false negative.

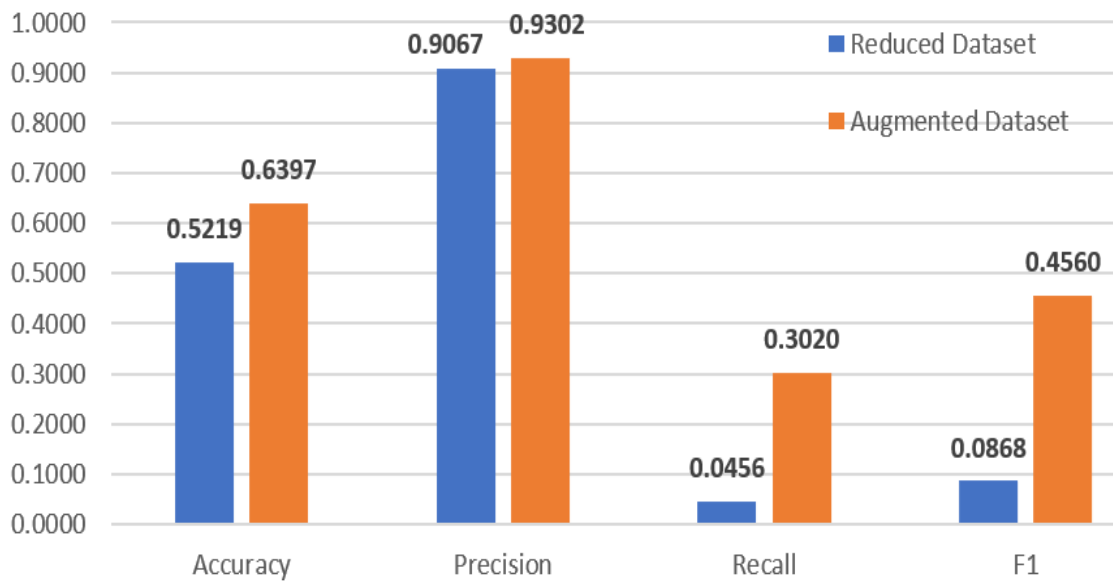


Figure 5.24: Comparison of the accuracy, precision, recall and  $F_1$  of GoogLeNet trained with and without using augmented images.

Figure 5.24 shows the accuracy, precision, recall and  $F_1$  values of the GoogLeNet trained with and without using augmented images for comparison. As shown in the figure, the use of augmented images generated by the proposed GAN model consistently improved the accuracy, precision, recall and  $F_1$  of the GoogLeNet for imbalanced image classification.

It can be seen from the above preliminary results that the classification performance of CNNs can be significantly improved by using augmented images generated by the proposed GAN model. However, the classification performance is still quite poor with a high false negative rate. For practical applications, further investigation should be conducted with larger original training image datasets.

## 5.5 Conclusion

In this chapter, a novel GAN model using one-to-one mapping methods is proposed. The proposed model is designed to generate diverse as well as photorealistic augmented images from limited input features, including sparse edges in the source domain and a small number of training images in the target domain. Refined images in the interim domain, presented in Section 5.2, are used to improve the synthetic reality and reduce the distortions caused by training with sparse edges and limited samples. The experimental results, shown in Section 5.4, demonstrate that the proposed GAN model can not only outperform the state-of-the-art image-to-image translation methods but also accordingly improve the validation accuracies of CNNs for image classification. The proposed one-to-one image translation method makes it feasible for a learning-based generative model to generate unblurring images from a small number of training samples, which would be beneficial to deal with problems in deep learning, such as scarcity of labelled data and imbalanced training data in real applications. To sum up, the proposed GAN model is advantageous in synthesising controllable, photorealistic and diverse augmented images from a small dataset, with promising applications in deep learning when a large number of training images are difficult to be collected.

## Chapter 6

# Augmenting Small Facial Expression Training Dataset Using a Novel GAN Model Based on Many-to-many Image Mapping

### 6.1 Introduction

Facial expressions provide critical information about an individual's physical and psychological status [213]. With the developments of deep learning and computer vision in recent years, facial expression recognition (FER) has played an important role in human-machine interfaces, especially in many realistic applications, such as mental detection, pain feeling, emotional understanding, human-machine communication, psychological analysis, and so on. However, it is a difficult task to collect a large number of expressional samples for training deep learning models with high performance [214]. Most publicly available facial expression datasets are constructed with limited facial expression sets, which generally contain insufficient representations for a deep learning model to learn effectively.

It is nearly impossible to acquire perfect and comprehensive facial expression data since each person can have diverse emotional expressions with 44 action units (facial muscles) [215]. It is challenging for machine learning models to recognise facial expressions based on small training datasets [216]. For instance, "happiness" may be due to various levels of facial actions, such as smiling, laughing, yelling, pouting and so on. Data augmentation is one of the efficient ways to mitigate the problem of lacking labelled facial expression data. By increasing the diversity and amount of training data from a small set of well-defined facial expression images, data augmentation can improve the FER performance in real applications.

GANs have been proven to have powerful capabilities for generating complex and high-quality synthetic images [217]. Nevertheless, it is hard for traditional GANs to generate photorealistic expressional images without sufficient labelled training data [218]. Generating realistic expressional images from a small number of training samples is not an easy task because with limited training samples there exists the overfitting problem in the discriminator, and thus the generator receives inadequate feedback on the quality of the generated images, which may also cause the problem of training collapse [219].

The motivation of the research in this chapter is to enhance the FER performance by developing a new GAN model capable of generating good-quality images with diverse facial expressions from a small number of training samples to enlarge the diversity and amount of facial expression images as an augmented training dataset for deep learning models to learn more meaningful representations in FER. The proposed model is based on a many-to-many image translation method [220], which is empowered by discovering a wider mapping relationship between two different labelled image domains especially when the unpaired expressional attributes are difficult to be found using traditional CNN structures. Specifically, the proposed GAN model can learn representations from a very small number of training samples and identify the spatial difference between expressions and facial attributes from two domains. The spatial information is provided by a feature map mechanism, which is proposed to assist the proposed model in transferring expressions correctly.

To demonstrate the effectiveness of the proposed GAN model for augmenting facial expression images, the augmented images are firstly evaluated with visual analysis and then applied to enhance the FER performance of convolutional neural networks (CNNs), including AlexNet, GoogLeNet, ResNet and VGGNet. The CNNs will be initially trained with a small number of original facial expression images in each emotional class, which are also used as the training data for the proposed GAN model to transfer neutral face images into different facial expression images. Due to the training restrictions, CNNs are the merely deep learning models to evaluate the performance of the generative models, even though many deep learning models were proposed in recent years, *e.g.*, active appearance model (AAM), active shape model (ASM), manifold-based models, *etc.*, which have been proved to reach remarkable results in FER applications [221]. For performance evaluation, classification accuracies of CNNs are preliminarily compared to identify the difference between performances with and without using the augmented images generated by the proposed model. Experimental results show that the proposed GAN model can effectively transfer neutral face images to images with different facial expressions and of high quality in terms of Fréchet inception distance (FID) [152] and kernel inception distance (KID) [153], and using these synthetic images as an augmented training dataset can significantly improve the FER accuracy of the CNNs even though a very small amount of original facial expression data is involved in training the proposed GAN for image data augmentation. The contributions of this chapter can be summarised as follows:

- To the best of our knowledge, this is the first study on GANs based on many-to-many image translation for facial expression transfer by using a very small set of training samples. The proposed GAN model not only increases data diversity in generating images with different facial



expressions from neutral face images but also maintains critical characteristics of the neutral face images, which is particularly desirable for meeting the data augmentation requirements based on learning labelled image mapping relationships from a small number of training samples.

- A novel feature map mechanism is introduced to enlarge the spatial view on expressional attributes, and a feature extractor is adopted in the proposed GAN model, which forwards the feature map information to the generator for synthetic diversity. Relying on the designed loss functions, the embedded feature map mechanism with an appropriate balance between the feature extractor and the discriminators is developed to generate desired facial expression images of high quality and good diversity from a very small set of training samples.
- Experimental results have demonstrated the effectiveness of the proposed methods in enhancing FER performance. According to the FER results from four different CNNs and student's t-test values, the use of augmented images can significantly improve the validation accuracy by more than 10% when only a small number of real images in each expressional class were used as training samples.

## 6.2 Methods

A novel many-to-many image translation method is introduced in this section. The image synthesis process relies on a cycle structure in the proposed GAN model. In contrast to the initial cycle structure in CycleGAN, the proposed model employs different network structures, mechanisms and loss functions to improve the learning efficiency from small FER datasets. The proposed model has the advantage of working with a small number of training samples due to the following reasons: 1) Firstly, for two different labelled domains, known as the source domain and target domain, the generator synthesises facial expressions without needing to use paired images. 2) The decoder specifically replaces expressional features with the extra region information provided by the proposed feature map mechanism. 3) The image synthesis process can recognise a larger range of mapping vision than traditional convolutional layers, which mitigates the generative uncertainty caused by a small training dataset.

An overview of the proposed GAN model is shown in Figure 6.1. Two differently labelled real facial expression images (represented as  $X$  and  $Y$ ) are allocated in the source domain and target domain respectively. The model is composed of five subnetworks, including two generators with the encoder and decoder structure, a feature

extractor and two discriminators. Four loss functions, including adversarial loss, cycle loss, perceptual loss and feature loss, are designed to separately work on different network components, as shown with the red dashed lines in Figure 6.1, and all are combined as the total loss function. The detail of each adopted loss function will be further discussed in Section 6.2.3. In the application phase, facial expression attributes should be transferred from the neutral face images in the source domain to new facial images with desired labelled attributes (or expressions) in the target domain. Contrasted to condition-based GANs, the images in the target domain here are equivalent to the conditional inputs.

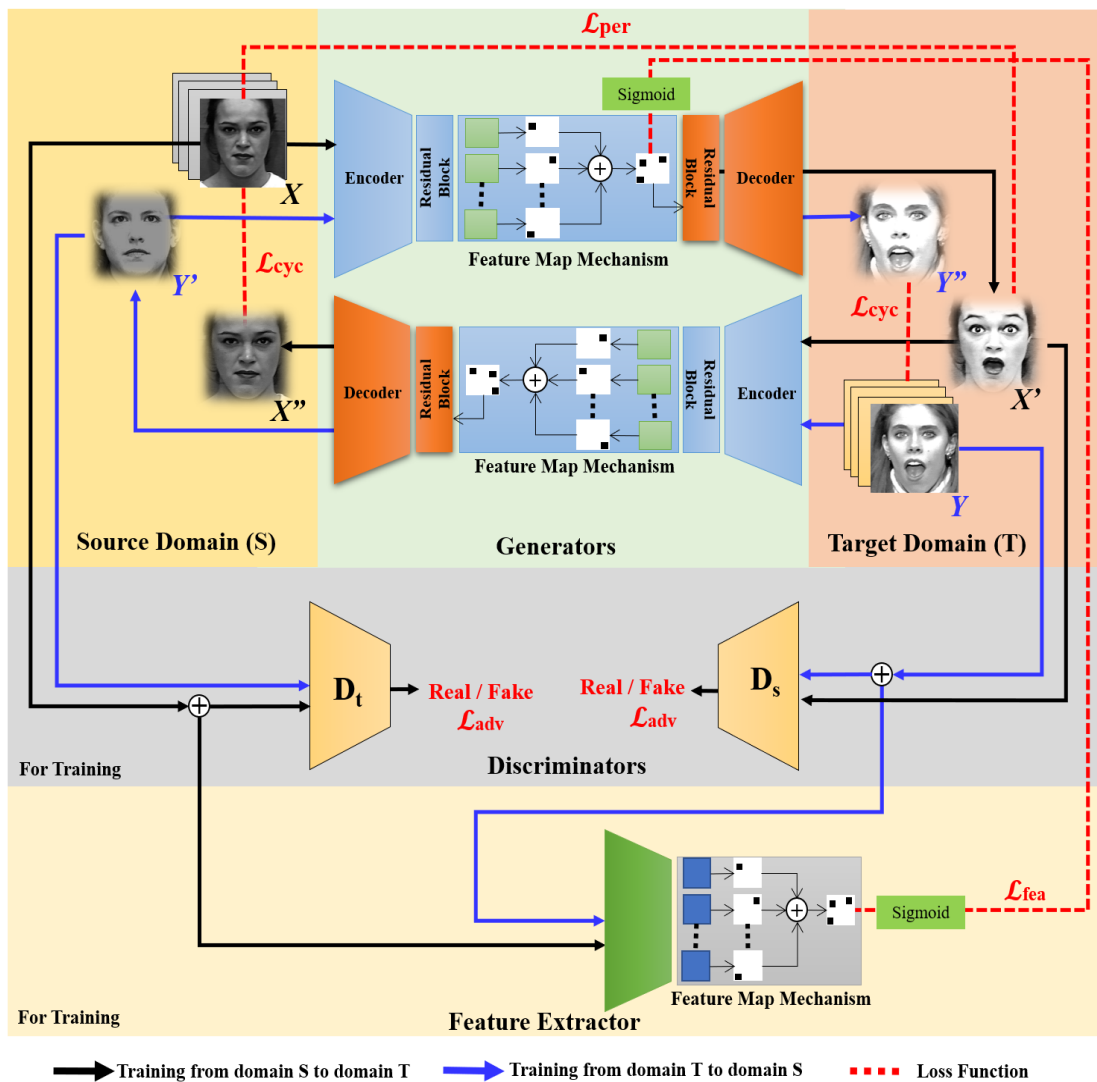


Figure 6.1: An overview of the proposed GAN model, which contains five subnetworks, including two generators with the encoder and decoder structure, a feature extractor and two discriminators. The proposed feature map mechanism is involved in both the encoder and feature extractor.

## 6.2.1 Subnetworks

Since a GAN framework hardly generates new contents of facial expressions without new identities, transferring additional neutral face images into images with various facial expressions as a new augmented dataset is a beneficial way to increase the data diversity as well as data amount from a small facial expression dataset. The proposed GAN model aims to regionally enlarge the mapping relationships between the source domain and target domain to mitigate the uncertainty of facial expression attributes resulting from using a very small set of training samples. Two discriminators are involved to recognise the quality of synthetic images in two different domains, and generators transfer images from one domain to another. In addition, since training GANs with a small number of training samples easily leads to overfitting, the proposed GAN model utilises additional subnetworks, consisting of the encoders, decoders and a feature extractor, which provides additional perspectives in the image synthesis process, to mitigate the overfitting or over-optimisation problem caused by limited training features. The details of the subnetworks, feature map mechanism, and training formulations are described as follows.

### 6.2.1.1 Generators

The generators in the proposed GAN model are made up of two encoders and two decoders to create facial expression attributes by reconstructing expressions from the two domains. The generator network and related parameters are shown in Table 6.1. Due to the generator taking responsibility to translate facial features across two different domains, the encoder is trained to identify the attribute difference, and feature map information is used by the decoder to regionally reconstruct facial attributes. The encoder is composed of convolutional layers, residual layers and a feature map mechanism, and the decoder has corresponding but contrasted components of convolutional layers and residual layers as in the encoder. The proposed feature map mechanism is embedded in the encoder and provides the generator with more capacity to process data, which enables the model to modify facial expressions with semantic regions so that the decoder can synthesise new images with desired facial expressions. The detail of the proposed feature map mechanism will be described in Section 6.2.2.

Instance normalisation is conducted in the normalisation layers of the generator for improving the performance of expression transferring. Instance normalisation was proposed by Ulyanov *et al.* in 2017 [222] and has been proven to significantly boost the normalisation in domain transfer compared to batch normalisation. Different from

batch normalisation, instance normalisation computes the spatial dimension independently with each channel, which is applied by unchanged test time. Instance normalisation has been widely used in image generation tasks, such as style transformation and image-to-image translation. Due to the powerful capacities of instance normalisation in domain transferring, the proposed generator adopts it to statistically exploit instance features whilst batch normalisation is usually affected by the parameter of a minimal batch value for statistical computations.

Additionally, residual blocks are embedded in the generators and provide a strong capacity to recognise expressional data from different domains. The residual blocks are expected to help the generators to modify the expressional attributes as well as recognise facial features. Furthermore, since the proposed feature map mechanism provides important semantical information in image synthesis, the residual blocks can process the feature maps fed into the mechanism and then assist to decode the outputs from the feature map mechanism. To well recognise the expression attributes, four residual blocks are used as parts of the encoder-decoder structure in the image synthesis process. The residual block used in the generator network is shown in Figure 6.2.

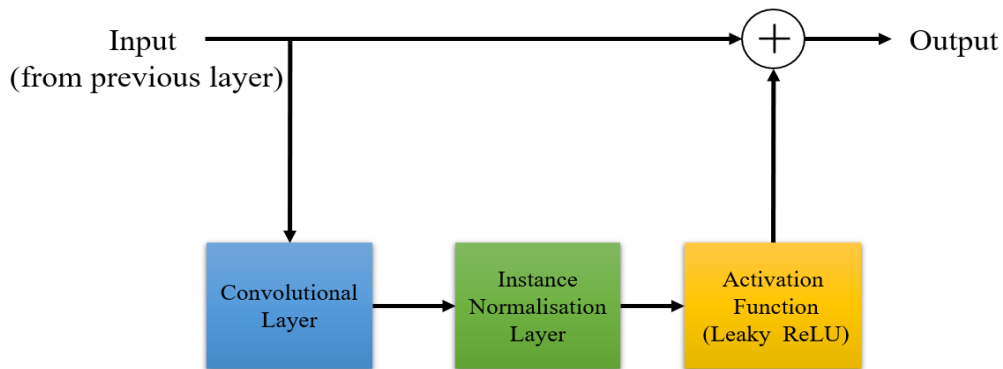


Figure 6.2: The residual block used in the proposed GAN model.

### 6.2.1.2 Discriminators

The two discriminators are designed to distinguish between real and fake images in separated domains during the training phase. New facial expression attributes can be functionally created to transfer facial expressions between two domains through a good balance between training the generator and discriminator. The feature map mechanism is also used in the discriminators. However, similar to traditional GANs, the two discriminators recognise the similarity between fake and real images in separate domains; the discriminator network and related parameters are shown in Table 6.2.

Table 6.1: The generator network and related parameters.

<i>Part</i>	<i>Name</i>	<i>Type</i>	<i>Input Size</i>	<i>Output Size</i>
Encoder (Convolutional Block)	Conv. Layer 1	Conv. Transposition	$3 \times 256 \times 256$	$64 \times 256 \times 256$
	Instance Normalisation 1	Normalisation	$64 \times 256 \times 256$	$64 \times 256 \times 256$
	Leaky ReLU 1	Activation	$64 \times 256 \times 256$	$64 \times 256 \times 256$
	Conv. Layer 2	Conv. Downsampling	$64 \times 256 \times 256$	$128 \times 128 \times 128$
	Instance Normalisation 2	Normalisation	$128 \times 128 \times 128$	$128 \times 128 \times 128$
	Leaky ReLU 2	Activation	$128 \times 128 \times 128$	$128 \times 128 \times 128$
	Conv. Layer 3	Conv. Downsampling	$128 \times 128 \times 128$	$256 \times 64 \times 64$
	Instance Normalisation 3	Normalisation	$256 \times 64 \times 64$	$256 \times 64 \times 64$
	Leaky ReLU 3	Activation	$256 \times 64 \times 64$	$256 \times 64 \times 64$
Encoder (Residual Block $\times$ 4)	Residual Block 1-4	Residual Block	$256 \times 64 \times 64$	$256 \times 64 \times 64$
	Instance Normalisation 1-4	Normalisation	$256 \times 64 \times 64$	$256 \times 64 \times 64$
	Leaky ReLU 1-4	Activation	$256 \times 64 \times 64$	$256 \times 64 \times 64$
Encoder (Feature Map Mechanism)	Max Pooling 1	Normalisation	$256 \times 64 \times 64$	$256 \times 32 \times 32$
	Fully-connected Layer 1	Dense	$256 \times 32 \times 32$	$256 \times 1 \times 1$
	Average Pooling 1	Normalisation	$256 \times 64 \times 64$	$256 \times 32 \times 32$
	Fully-connected Layer 2	Dense	$256 \times 32 \times 32$	$256 \times 1 \times 1$
	Concatenate 1	Inception	$512 \times 64 \times 64$	$512 \times 64 \times 64$
Decoder (Residual Block $\times$ 4)	Conv. Layer 1	Conv. Transposition	$512 \times 64 \times 64$	$256 \times 64 \times 64$
	Residual Block 1-4	Residual Block	$256 \times 64 \times 64$	$256 \times 64 \times 64$
	Instance Normalisation 1-4	Normalisation	$256 \times 64 \times 64$	$256 \times 64 \times 64$
	Leaky ReLU 1-4	Activation	$256 \times 64 \times 64$	$256 \times 64 \times 64$
Decoder (Convolutional Block)	Conv. Layer 1	Conv. Upsampling	$256 \times 64 \times 64$	$128 \times 128 \times 128$
	Instance Normalisation 1	Normalisation	$128 \times 128 \times 128$	$128 \times 128 \times 128$
	Leaky ReLU 1	Activation	$128 \times 128 \times 128$	$128 \times 128 \times 128$
	Conv. Layer 2	Conv. Upsampling	$128 \times 128 \times 128$	$64 \times 256 \times 256$
	Instance Normalisation 2	Normalisation	$64 \times 256 \times 256$	$64 \times 256 \times 256$
	Leaky ReLU 2	Activation	$64 \times 256 \times 256$	$64 \times 256 \times 256$
	Conv. Layer 3	Conv. Transposition	$64 \times 256 \times 256$	$3 \times 256 \times 256$
	Tanh 1	Activation	$3 \times 256 \times 256$	$3 \times 256 \times 256$

Table 6.2: The discriminator network and related parameters.

<i>Name</i>	<i>Type</i>	<i>Input Size</i>	<i>Output Size</i>
Conv. Layer 0	Conv. Transposition	$3 \times 256 \times 256$	$64 \times 128 \times 128$
Leaky ReLU 0	Activation	$64 \times 128 \times 128$	$64 \times 128 \times 128$
Conv. Layer 1	Conv. Downsampling	$64 \times 128 \times 128$	$128 \times 64 \times 64$
Leaky ReLU 1	Activation	$128 \times 64 \times 64$	$128 \times 64 \times 64$
Conv. Layer 2	Conv. Downsampling	$128 \times 64 \times 64$	$256 \times 32 \times 32$
Leaky ReLU 2	Activation	$256 \times 32 \times 32$	$256 \times 32 \times 32$
Conv. Layer 3	Conv. Downsampling	$256 \times 32 \times 32$	$512 \times 16 \times 16$
Leaky ReLU 3	Activation	$512 \times 16 \times 16$	$512 \times 16 \times 16$
Conv. Layer 4	Conv. Downsampling	$512 \times 16 \times 16$	$1024 \times 8 \times 8$
Sigmoid 1	Classifier	1	1

### 6.2.1.3 Feature Extractor

The feature extractor, which is based on a convolutional network, is supposed to semantically extract specific expressional features from both the source domain and target domain. The feature extractor network and related parameters are shown in Table 6.3. To promote the mapping relationships between two domains, a feature map mechanism is embedded in the feature extractor to enlarge the mapping vision. To control the synthetic process of expressional transfer, the feature extractor is trained to extract regional information from real expressional data. Due to unpaired images used as the translation conditions, it is very difficult to acquire precise mapping relationships from a small number of training samples. The feature extractor is designed to acquire additional expressional representations with attention maps, which can roughly identify the activation regions of expressional attributes from feature maps. In addition, a feature map mechanism is embedded in the feature extractor and proposed to enlarge the mapping relationships between two domains by providing critical spatial region information in the image synthesis process. Feature maps can be produced by the feature extractor, which also forwards the progressing information to the encoder during the training phase. Compared with convolutional layers, this proposed feature map mechanism in the feature extractor is to reduce the expression uncertainty in facial

image synthesis and successfully transfer facial expressions based on a small set of facial expression image samples, which will be further discussed in Section 6.2.2.

Table 6.3: The feature extractor network and related parameters.

<i>Part</i>	<i>Name</i>	<i>Type</i>	<i>Input Size</i>	<i>Output Size</i>
Conv. Block	Conv. Layer 0	Conv. Transposition	$3 \times 256 \times 256$	$64 \times 256 \times 256$
	Leaky ReLU 0	Activation	$64 \times 256 \times 256$	$64 \times 256 \times 256$
	Conv. Layer 1	Conv. Downsampling	$64 \times 256 \times 256$	$128 \times 128 \times 128$
	Leaky ReLU 1	Activation	$128 \times 128 \times 128$	$128 \times 128 \times 128$
	Conv. Layer 2	Conv. Downsampling	$256 \times 128 \times 128$	$256 \times 64 \times 64$
	Leaky ReLU 2	Activation	$256 \times 64 \times 64$	$256 \times 64 \times 64$
Feature Map Mechanism	Max Pooling 1	Normalisation	$256 \times 64 \times 64$	$256 \times 32 \times 32$
	Fully-connected Layer 1	Dense	$256 \times 32 \times 32$	$256 \times 1 \times 1$
	Average Pooling 1	Normalisation	$256 \times 64 \times 64$	$256 \times 32 \times 32$
	Fully-connected Layer 2	Dense	$256 \times 32 \times 32$	$256 \times 1 \times 1$
	Concatenate 1	Inception	$512 \times 64 \times 64$	$512 \times 64 \times 64$
	Sigmoid 1	Classifier	1	1

In contrast to the discriminator network for recognising real or fake data, the feature extractor is responsible for recognising differences in regional attributes to acquire geometric information of attention maps with a larger view. By an appropriate design of loss functions, a balance between the discriminators and feature extractor can be achieved, and the generator can synthesise photorealistic and desirable facial expression images in terms of data augmentation requirements.

## 6.2.2 Feature Map Mechanism

Feature maps are vectors of latent variables, which need to be extracted from the last convolutional layers of the encoder and feature extractor. Convolutionally acquired from input data consisting of images with different facial expressions of two domains, feature maps represent different attributes between two differently labelled expressions. In theory, convolutional features as feature maps are calculated based on small local neighbourhoods using filters (or kernels), which are difficult to receive a

comprehensive view through a small amount of training data, as the positions of facial attributes are always within a larger region than the perceptive fields of kernels.

Based on the above concerns, a feature map mechanism is proposed, which is implemented by two subnetworks, including the feature extractor and encoder. The feature extractor is designed to automatically acquire the active regions from the real images of two domains and further pass the extracted information is passed to the encoder via adversarial learning, formulated as  $E_{x \sim P_{data}} [G_{enc}(X)] = E_{y \sim P_{data}} [G_{enc}(Y)]$ , where  $Y$  is the real data in the target domain;  $X$  is the real data in the source domain;  $G_{enc}$  is the output of the encoder. The outputs from the feature map mechanism are also the inputs of the decoder.

In the proposed mechanism, the use of pooling and fully connected layers can semantically consider a larger expressional range than the convolutional layers to assist regional expression discovery. Each feature map, denoted as  $F_i$ , is weighted by a weight value  $w_i$ , which automatically calculates the importance of each feature map from the real data. The weighted sum of absolute values of the feature maps, also known as attention maps, can be treated as a new map denoted as  $A_0$ , which is formulated as follows:

$$\sum_{i=1}^n w_i |F_i| = \sum_{i=1}^n A_i = A_0 \quad (6.1)$$

where  $n$  is the number of channels or filters.

Figure 6.3 shows the feature map mechanism in the proposed model. Whether the input features are from residual layers in the encoder or convolutional layers in the feature extractor, they need to be normalised by maximum pooling and average pooling, both of which aim to detect a larger relationship. Furthermore, for discovering a closer mapping relationship among feature maps, fully-connected layers are used to compute the weight values from max pooling and average pooling separately. The final attention map obtained by a weighted sum of all the feature maps with semantical regularisations can be regarded as covering a clearer and wider view than the original feature maps.

As a whole, the adoption of the pooling in the feature map mechanism is to enlarge the mapping relationships by reducing the feature size of the convolutional layers, and the fully connected layers are to evaluate the importance of each feature map derived from the results of separate pooling with weight values. In the proposed mechanism, the outputs of attention maps are trained to highlight the critical regional information learnt from the real expressional images in both the source domain and target domain. Consequently, the final attention map provides a comprehensive view and significant



regional information in the sense that the regions with higher weight values should be paid more attention in the image synthesis process.

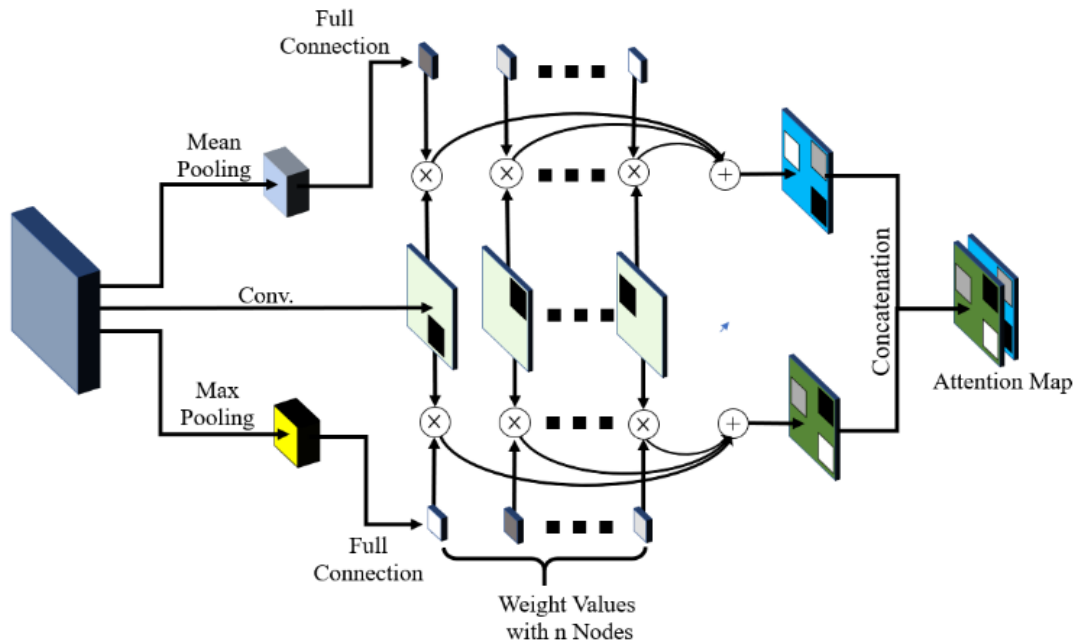


Figure 6.3: The feature map mechanism in the proposed model.

### 6.2.3 Model Learning

For generating desired facial attributes, several loss functions are adopted, and they work together as the objective function to train the proposed GAN model: 1) Firstly, an adversarial loss function should be designed with the fundamental principle of adversarial learning in a GAN-based model. 2) Secondly, regarding the cycle consistency structure, the image translation is designed to be constrained by a cycle loss function during training. 3) Furthermore, since the generated images have to maintain similar characteristics of the input neutral face images for data augmentation requirements, a perceptual loss function is involved in the learning. 4) Finally, the proposed feature map mechanism is expected to encourage the model to transfer photorealistic expressional representations, and a feature loss function should be included to effectively augment neutral face images based on a small training dataset. Consequently, the above-mentioned loss functions are combined as the overall objective function to simultaneously train the five subnetworks of the proposed GAN model. The details of the loss functions are described as follows.

### 6.2.3.1 Adversarial Loss

Let  $X$  be real images in the source domain and  $Y$  real images in the target domain. To find the expectation values  $\mathbb{E}$  of the data distribution  $P_{data}$ , the generator  $G$  has to capture the data distribution from both domains with an adversarial learning process, whilst the discriminators,  $D_s$  and  $D_t$ , aim to distinguish the fake and real data from the source domain and target domain respectively. The adversarial loss is designed to make the generated images visually photorealistic under a balance between the generator and discriminator [15], where the adversarial losses in the source domain and target domain are separately formulated as follows:

$$\begin{aligned}\mathcal{L}_{adv}(G, D_s, X, Y) &= \mathbb{E}_{x \sim P_{data}}[\log D_s(X)] \\ &+ \mathbb{E}_{y \sim P_{data}}[\log (1 - D_s(G(Y)))]\end{aligned}\quad (6.2)$$

$$\begin{aligned}\mathcal{L}_{adv}(G, D_t, Y, X) &= \mathbb{E}_{y \sim P_{data}}[\log D_t(Y)] \\ &+ \mathbb{E}_{x \sim P_{data}}[\log (1 - D_t(G(X)))]\end{aligned}\quad (6.3)$$

where  $G(\cdot)$  is defined such that  $\begin{cases} X \xrightarrow{G(X)} X' \approx Y \\ Y \xrightarrow{G(Y)} Y' \approx X \end{cases}$

Ideally, with an adversarial loss, the generated images  $G(X)$  should follow a data distribution of the real data  $Y$  whilst  $G(Y)$  follows that of the real data  $X$ . To evaluate the generative quality, two discriminators are designed to guide the generators to create data with expectation values  $\mathbb{E}$ . When the two discriminators are used to respectively distinguish fake images from real ones in different domains, the generated images will be labelled as the data transferred to another domain. However, only with adversarial loss, it is easy to generate visually meaningless results with unexpected expression distortions, especially when trained with a small dataset. Even though the proposed model can learn the basic representations to transfer expressions, distortions and unreal generative results usually happen with the adversarial loss function when the generators merely receive the real or fake information from discriminators. Furthermore, the generators are expected to map a set of input images and generate desired expressional data in the target domain, but the adversarial loss only indicates the similarity or difference between real and fake data distributions, which cannot guarantee to map neutral face images to desired expressional features. Therefore, besides an adversarial loss function, additional attribute constraints need to be used to train the proposed GAN model.

### 6.2.3.2 Cycle Loss

To enhance the mapping capacity between two different domains, the adoption of cycle loss is a beneficial method in the sense that images transferred to another domain can be brought back to the original one. Thus, the generators are forced to learn more specific mapping relationships instead of merely learning the generative similarity with the adversarial loss.

To enlarge the mapping relationships among unpaired images, the loss defined in CycleGAN [168] is adopted as the cycle loss function that aims to learn the mapping consistency between the source domain and target domain to reduce the problem of mode collapse, which often happens with all input images mapped to a few output images. For real images  $X$  in the source domain, the synthetic images  $X'$  should satisfy the cycle consistency; that is when  $X$  is transferred to  $X'$  in the target domain, and then  $X'$  is transferred back to  $X''$  in the source domain,  $X$  and  $X''$  should be visually similar, represented as  $X \approx X''$ . The same situation should be satisfied with real images  $Y$  in the target domain, *i.e.*,  $Y \approx Y''$ . The  $L_1$  norm is used to formulate the cycle loss [223], which is widely applied to measure the similarity between two different pictures. The cycle loss is defined as follows:

$$\mathcal{L}_{cyc}(G) = \mathbb{E}_{x \sim P_{data}} \|X'' - X\|_1 + \mathbb{E}_{y \sim P_{data}} \|Y'' - Y\|_1 \quad (6.4)$$

where  $G(\cdot)$  is defined such that  $\begin{cases} X \xrightarrow{G(X)} X' \xrightarrow{G(X')} X'' \approx X \\ Y \xrightarrow{G(Y)} Y' \xrightarrow{G(Y')} Y'' \approx Y \end{cases}$

### 6.2.3.3 Perceptual Loss

For facial image reconstruction and data augmentation, the generated expressional face images are expected to look similar to the input neutral face images with desirable facial expressions but without a large scale of differences and distortions. For this purpose, a loss function is additionally needed for comparing the similarity between the generated images and original images, where the  $L_1$  norm is used as well to represent the perceptual loss for reducing the generative blurriness and distortions. The perceptual loss is formulated as follows:

$$\mathcal{L}_{per}(G) = \mathbb{E}_{x \sim P_{data}} \|X' - X\|_1 \quad (6.5)$$

It is noted that the perceptual loss only evaluates the generated data with one

transformation direction from real images  $X$  to fake images  $X'$  instead of from  $Y$  to  $Y'$  mainly because the proposed model is only designed to augment the neutral face images to expressional ones. The generated contents are expected to augment images similar to the original input of neutral faces in the source domain, but an adversarial loss is not able to verify whether the generated images look similar or not. Therefore, the generated data  $X'$  is further constrained by the perceptual loss for learning the data similarity between two labelled domains. Consequently, the perceptual loss concentrates on the feature similarity between input images and synthetic results, which are designed to transfer facial attributes, except for expressions, for meeting the data augmentation requirements.

### 6.2.3.4 Feature Loss

To generate images with expected facial expressions using a small training dataset, the feature extractor in the proposed GAN model needs to identify the feature representations between two different classes of facial expressions. For this purpose, a feature loss is designed for training the proposed GAN model, which aims to enhance the quality of the final attention map generated by the encoder  $G_{enc}$  and feature extractor  $E$  for correctly translating expressional attributes. The feature loss function is defined as follows:

$$\begin{aligned} \mathcal{L}_{fea}(G_{enc}, E, X, Y) &= \mathbb{E}_{x,y \sim P_{data}} [\log E(X, Y)] \\ &+ \mathbb{E}_{x \sim P_{data}} [\log (1 - G_{enc}(X))] \end{aligned} \quad (6.6)$$

where  $E(X, Y) \approx G_{enc}(X) = G_{enc}(Y)$

Different from the previously proposed loss functions for evaluating the quality of generative results, the feature loss is designed to assess the quality of the regional information in the feature map mechanism. Due to the facial reconstruction process that significantly relies on the feature map mechanism, improving the learning process of the proposed model to control the synthetic quality in the feature map mechanism is a critical factor in facial reconstruction. The activation regions of the final attention map in the feature map mechanism are constrained by the feature loss function. With the use of feature loss, the model can be more sensitive to the expression attributes, and the attention map information can be correctly forwarded to the encoder during the training process. The adoption of the feature loss constrains the encoders to learn knowledge from the final attention map. Consequently, the expression difference and regional information are restricted by the feature loss, and photorealistic results can be generated

by regionally mitigating the unexpected distortions caused by training the proposed GAN model with a small number of labelled facial image samples.

### 6.2.3.5 Overall Loss

By combining the adversarial loss, cycle loss, perceptual loss, and feature loss, the objective of training the proposed GAN model is as follows:

$$\begin{aligned} \min_{D_s, D_t, E} \max_{G, G_{enc}} \mathcal{L}_{adv}(G, D_s, X, Y) + \mathcal{L}_{adv}(G, D_t, Y, X) + \lambda_1 \mathcal{L}_{cyc}(G) \\ + \lambda_2 \mathcal{L}_{per}(G) + \lambda_3 \mathcal{L}_{fea}(G_{enc}, E, X, Y) \end{aligned} \quad (6.7)$$

where  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are weighting parameters controlling the contribution of the gradient penalty to the overall loss.

## 6.3 Experiments with the Proposed GAN Framework

In this section, datasets, experimental setup and ablation studies for the proposed method are described as follows.

### 6.3.1 Datasets

Three facial expression datasets, the extended Cohn-Kanade (CK+) [224], Karolinska directed emotional faces (KDEF) [225] and Taiwanese facial expression image database (TFEID) [226], were partially used as the small labelled datasets to evaluate the performance of the proposed GAN model.

#### 6.3.1.1 Extended Cohn-Kanade Dataset

The extended Cohn-Kanade dataset (CK+) is composed of 593 videos from 123 subjects of different genders and heritage. The participants consist of 100 university students aged from 18 to 30 years old, where 65% are female; 15% are African-American; 3% are Asian or Latino. Images in the database were extracted from videos and each subject was instructed to perform expressions. The videos illustrate facial shifts from neutral expression to the targeted expression with a resolution of  $640 \times 480$  pixels.

### 6.3.1.2 Karolinska Directed Emotional Faces Dataset

The Karolinska directed emotional faces dataset (KDEF) provides images with  $762 \times 562$  pixels, which were taken from 70 individuals (35 females and 35 males) with a set of 5 different angles. Seven fundamental emotional expressions are used, including neutral, happiness, anger, sadness, disgust, surprise, and fear. In our experiments, only the frontal view of each individual was randomly selected as the small training dataset.

### 6.3.1.3 Taiwanese Facial Expression Image Dataset

The Taiwanese facial expression image database (TFEID) contains 7,200 images, and 40 participants are involved with one neural class and seven facial expressions, which are anger, contempt, disgust, happiness, surprise, fear and sadness. The images are based on two different angles of  $0^\circ$  and  $45^\circ$ . Only frontal face images with angles of  $0^\circ$  were adopted in our experiments.

Table 6.4: Description of facial expression datasets.

<i>Name</i>	<i>Expressions</i>	<i>No. of Images</i>	<i>Resolution</i>
<i>CK+</i>	Neutral, sadness, surprise, happiness, fear, anger, contempt and disgust	486	$640 \times 480$
<i>KDEF</i>	Neutral, anger, disgust, fear, happiness, sadness and surprise	490	$762 \times 562$
<i>TFEID</i>	Angry, Fearful, Disgusted, Sad, Happy, Surprised and Neutral	7,200	$600 \times 480$

Table 6.4 shows the FER information of used datasets, including expression types, number of released images and resolution details. The datasets are labelled with 6 or 7 different facial expressions (*e.g.*, happy, sad, surprise, angry, neutral, fear, disgust, *etc.*) to train the proposed model. A detailed description of the dataset along with facial expressions as well as the number of images is presented in Table 6.4. To form a small training dataset, only a limited set of images was chosen from the facial expression datasets, which was partially adopted to train the proposed GAN model, and the remaining images were employed as the validation data for performance evaluation. The numbers of chosen images used as the small training dataset are shown in Table 6.5.

Table 6.5: Number of images used as the small datasets.

	<i>CK+</i>	<i>KDEF</i>	<i>TFEID</i>	<i>Domain</i>
<i>Neutral</i>	50	70	40	Source Domain
<i>Sadness</i>	28	70	40	
<i>Surprise</i>	62	70	36	
<i>Happiness</i>	59	70	40	
<i>Fear</i>	24	70	40	Target Domain
<i>Anger</i>	45	70	34	
<i>Disgust</i>	59	70	40	
<i>Contempt</i>	45	/	40	
<b><i>Total</i></b>	<b>372</b>	<b>490</b>	<b>310</b>	/

### 6.3.2 Experimental Setup

The Adam optimiser [147] was used for model training, where the values of  $\beta_1$  and  $\beta_2$  were set to 0.5 and 0.999 respectively. The learning rates for both the discriminator and generator were set to 0.0001, and the batch size was chosen appropriately for small training datasets. For evaluating the performance without over-training and preventing the overfitting problem, for all cases, the number of training epochs was less than 4,000. The training was conducted with two *Nvidia RTX 2080 GPUs*. The input and output images were cropped to keep the centre part with faces only, which is particularly useful in the case of training with small datasets. The resolution was initialised to  $256 \times 256$  pixels, and the size of attention maps is  $64 \times 64$  pixels. The values of the weighting parameters  $\lambda_1, \lambda_2, \lambda_3$  were set to  $10^1, 10^2$  and  $10^3$  separately, which were determined by the experimental results of an ablation study, shown in Section 6.3.3.1.

Since the adopted datasets include basic facial expression categories (*e.g.*, sadness, happiness, surprise, sadness, *etc.*), the expressional images in each class can be directly used as the well-defined training data in the target domain, which provides essential information for augmenting neutral faces to new expressional faces. The chosen expressional face images are not necessarily paired with neutral face images, *i.e.*, they can be from different subjects.

### 6.3.3 Ablation Studies

To evaluate the effectiveness of the proposed GAN model, three ablation studies are presented in this section: 1) Firstly, the impacts of the loss functions with different

weight values are discussed in Section 6.3.3.1. 2) Secondly, in Section 6.3.3.2, different adversarial loss functions are compared in terms of the generative performance. 3) Finally, the synthetic results with or without using the proposed feature map mechanism are demonstrated in Section 6.3.3.3. The details and experimental results of the above ablation studies are described as follows.

### 6.3.3.1 Weighting Values in the Overall Loss Function

Finding out the optimal weight values in the overall loss function is not easy, especially when millions of free parameters in the network need to be tuned for a specific set of weight values. In this ablation study, the impacts of the weighting parameters in the overall loss function on the quality of generated images are analysed with the KDEF dataset. It is difficult to determine the weighting values because the hyperparameters cannot be learned using gradient-based methods, and no universal optimization methods can be followed to quickly discover the optimal weighting values. A decimal scale is used with trail-and-error approaches to preliminarily examine the effect of different values for weighting parameters  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  on the quality of the generated facial expression images. Figure 6.4 shows the effects of different weight values in the overall loss function on the generated surprise images based on a small KDEF dataset consisting of 20 surprise images and 70 neutral images.

Three points can be found from the experimental results: 1) Firstly, small weight values of  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$  (e.g.,  $\lambda_1 = 1$ ,  $\lambda_2 = 1$  and  $\lambda_3 = 1$ ) easily cause distortions in the generated images, but large weight values (e.g.,  $\lambda_1 = 1,000$ ,  $\lambda_2 = 1,000$  and  $\lambda_3 = 1,000$ ) cannot lead to photorealistic expressional results either. 2) Secondly, relatively large values of  $\lambda_1$ ,  $\lambda_2$  (e.g.,  $\lambda_1 = 100$ ,  $\lambda_2 = 1,000$  or  $\lambda_1 = 1,000$ ,  $\lambda_2 = 100$ ) can mitigate synthetic blurs and generate images similar to the input images. Fortunately, with relatively small values of  $\lambda_1$ ,  $\lambda_2$  (e.g.,  $\lambda_1 = 10$  and  $\lambda_2 = 10$ ), the generative results are still acceptable results without obvious distortions. 3) Thirdly, a large weight value of  $\lambda_3$  (e.g.,  $\lambda_3 = 1,000$ ) coupled with a small weight ratio of  $\lambda_1$  and  $\lambda_2$  (e.g.,  $\lambda_1 = 10$ ,  $\lambda_2 = 10$  or  $\lambda_1 = 10$ ,  $\lambda_2 = 100$ ) leads to desirable results, *i.e.*, expected expression transfer and photorealistic quality.

It can be concluded from the experimental results shown in Figure 6.4 that the weight values have a significant impact on the performance of the GAN model for facial expression transfer. Without appropriate weight values in the overall loss function, it is difficult for the proposed GAN model to preserve the facial attributes in the neutral face images and generated images with desired facial expressions by facial expression transfer. According to the above observations, the following weight values were chosen for further experiments:  $\lambda_1 = 10$ ,  $\lambda_2 = 100$  and  $\lambda_3 = 1,000$ . Based on the above experimental findings, the weight values of  $\lambda_1 = 10$ ,  $\lambda_2 = 100$  and  $\lambda_3 = 1,000$  achieve



competitive performance to generate expected expression results than the images generated by other weight values. Accordingly, this set is picked as the default weight value to train the proposed model in the remaining experiments. Moreover, the set found from the KDEF dataset is still functional to the datasets of CK+ and TFEID for visualising the obvious expressional changes. Even if these selected weighting values are not the best for achieving the best synthetic results, they could be available as a baseline to evaluate the preliminary performance and enable further optimisation in the future.



Figure 6.4: Effect of different weight values in the overall loss function on the generated surprise images with the KDEF dataset.

### 6.3.3.2 Adversarial Loss

The adversarial loss function is fundamental for adversarial learning in a GAN-based structure. In this ablation study, the state-of-the-art loss functions, including those used in WGAN [130], WGAN-GP [131] and LSGAN [132], were adopted to replace the adversarial loss defined in (6.2) and (6.3) in order to evaluate the effect of different adversarial loss functions on the performance of adversarial learning in the generators and discriminators of the proposed GAN model. Surprise images generated by the proposed GAN model based on different adversarial loss functions are shown in Figure 6.5 and Figure 6.6 with the KDEF and CK+ datasets respectively.

It can be seen that the images generated by WGAN and WGAN-GP contain more blurring distortions and uncertainty than the results generated by the proposed GAN model and LSGAN. LSGAN and the proposed GAN model achieved competitive results in transferring neutral faces to surprise expression images, but the adversarial loss function in the proposed GAN model is simpler and can achieve a good balance between training the generators and discriminators with a small number of training samples.

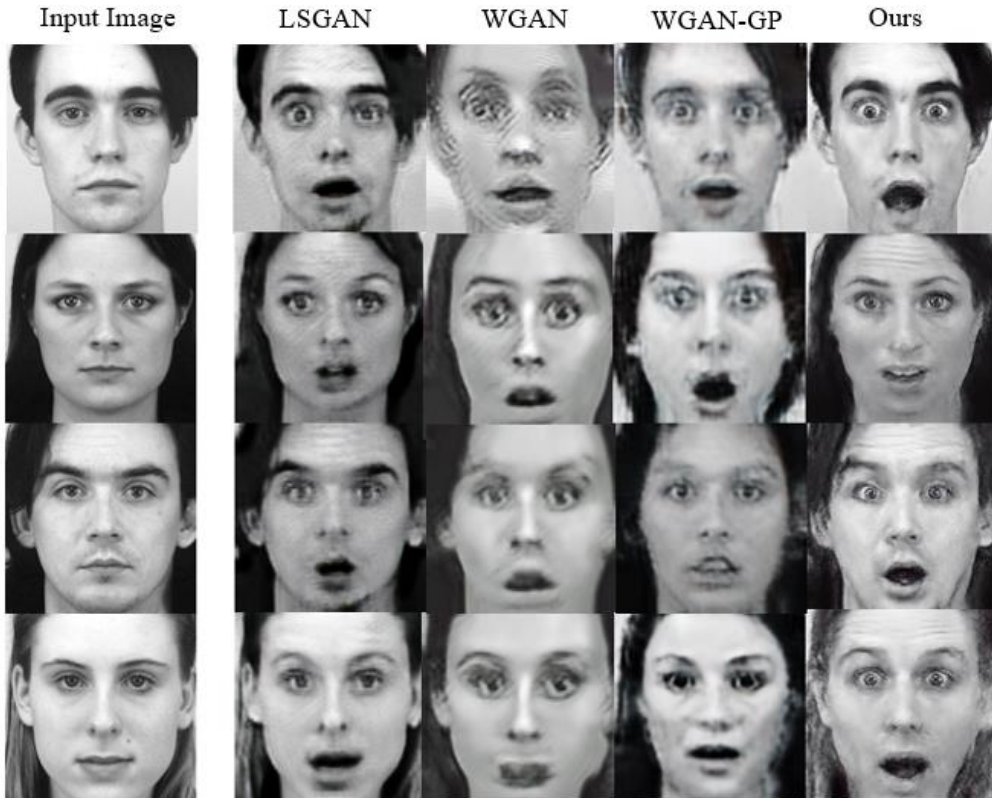


Figure 6.5: Effect of different adversarial loss functions on the generated surprise images with the KDEF dataset.

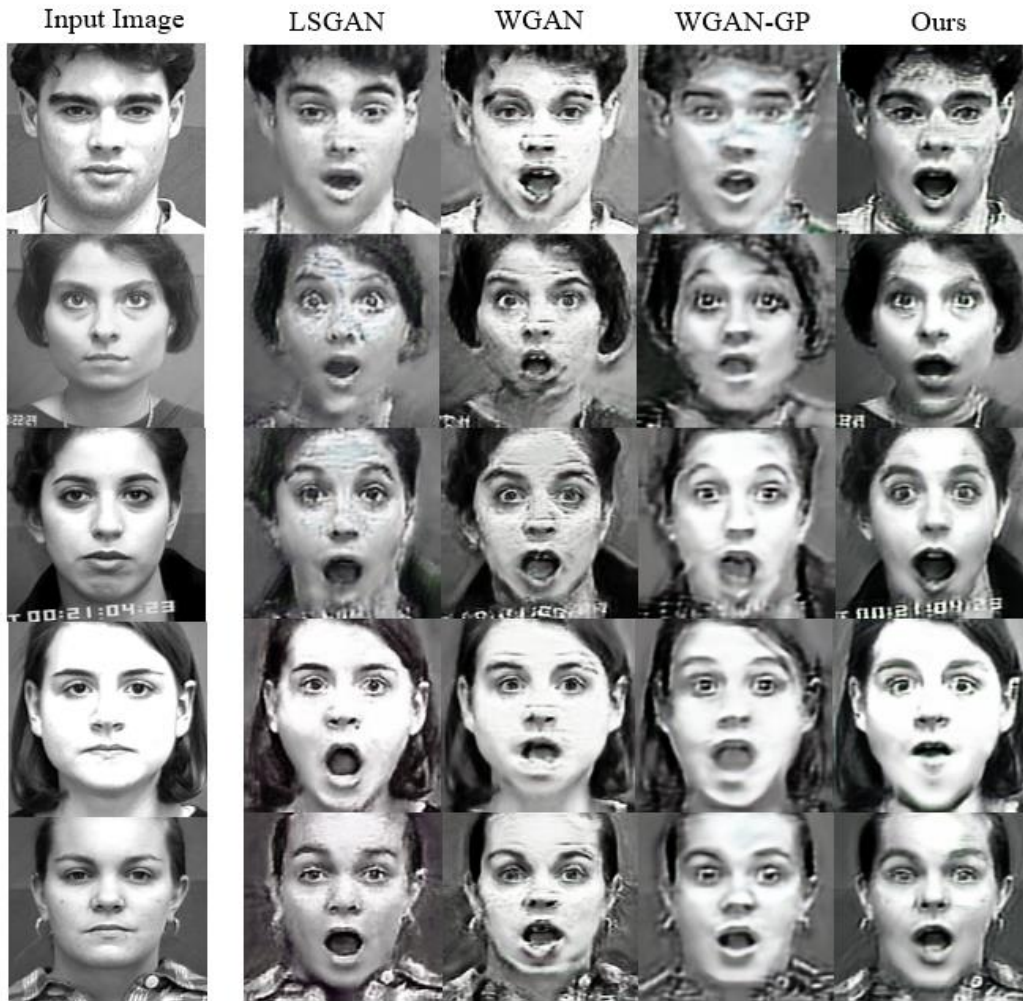


Figure 6.6: Effect of different adversarial loss functions on the generated surprise images with the CK+ dataset.

It has to be emphasised although many studies have proven that WGAN, WGAN-GP and LSGAN reached excellent performance, compared to the adversarial loss in some specific cases and applications [130], [131], [132], the adversarial loss is still an efficient method by using a binary classifier to simply check the synthetic quality with real or fake information, especially when a very small number of training samples are collected. The proposed GAN model with the adversarial loss can outperform the other state-of-the-art methods, which represents that loss functions need to be carefully validated in different applications. On the other hand, it is evident based on the experimental results that a powerful loss function generally needs a large amount of training data to support the algorithm for fine-tuning the hyperparameters of deep networks. Nevertheless, the adversarial loss contains a fundamental function to check the synthetic quality, which makes two deep networks, discriminator and generator, simply discover a potential data distribution with a small number of training images

and have a lower computation cost to cooperate with other loss functions.

### 6.3.3.3 Feature Map Mechanism

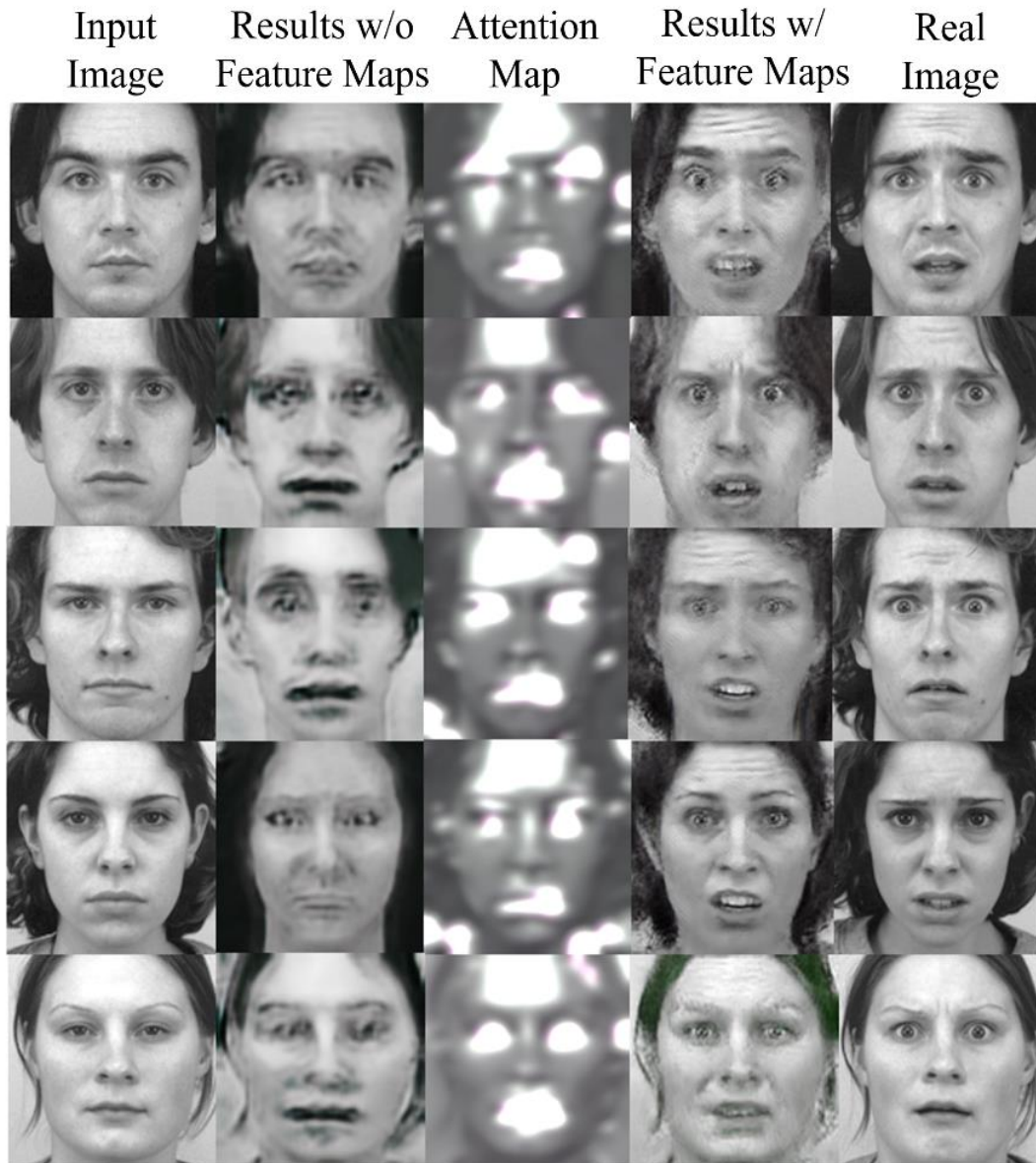


Figure 6.7: Comparison of facial expression images generated by the proposed GAN model with and without the feature map mechanism. The real images are from the KDEF dataset without being involved in the training, during which fear images were used as target samples and neutral images were used as inputs.

This ablation study is to demonstrate the role of the feature map mechanism in improving the performance of the proposed GAN model. The use of the feature map mechanism aims to regionally recognise the mapping difference between the target

domain and source domain based on a small number of training samples. In this experiment, the training samples were chosen from the KDEF dataset with 70 neutral images in the source domain and 20 images of fear class in the target domain. Figure 6.7 shows the images generated by the proposed GAN model with or without the feature map mechanism respectively, which demonstrates the benefit of using the feature map mechanism that can pay attention to the expressional attributes of the transforming regions (lighter areas), such as eyes, mouth, teeth and eyebrows, as indicated on the attention maps. On the other hand, the feature map mechanism enables the generated images to maintain the original features of the neutral faces with low distortions in the light-weight regions (darker areas). Consequently, the feature map mechanism further assists the proposed GAN model to generate more photorealistic and high-quality images, compared to the situation without using it, especially when only a small number of samples were used as the training data.

## **6.4 Performance Evaluation**

Both visual analysis and quantitative evaluation are presented in this section. Firstly, for visual analysis, images generated by the proposed GAN model with different numbers of training samples in the target domain are visually compared, and the comparative results with various expressional classes are illustrated in the visual analysis as well. Secondly, for quantitative evaluation, FID, KID and transfer learning are employed separately to evaluate the synthetic reality of augmented images and their role in improving the accuracy of image classification using deep learning models. Additionally, the student's t-test is used to further evaluate the FER performance of the CNNs trained with augmented facial expression images. Four CNNs, *i.e.*, AlexNet, VGGNet, GoogLeNet and ResNet, were trained with and without using the augmented data respectively for the purpose of evaluating the role of using the facial expression images generated by the proposed GAN model in enhancing the FER performance of CNNs.

### **6.4.1 Qualitative Comparison**

#### **6.4.1.1 Visual Analysis with Different Number of Training Images in the Target Domain**

In this experiment, visual analysis was adopted to evaluate the performance of the

generative quality of the proposed GAN model with different numbers of training samples. The generative results were obtained by using different numbers of expressional face images in the target domain coupled with a constant number of neutral face images in the source domain. Forty neutral images were used in the source domain, and different numbers (5, 10, 15, 20, 40) of surprise face images were randomly chosen for the target domain, which formed a small training dataset for training the proposed GAN. For comparing the effect of using different numbers of training samples on the quality of the generated images, Figure 6.8 and Figure 6.9 show typical examples of the generated surprise face images on the KDEF and TFEID datasets respectively, where the first column shows the real neutral faces and the last column the ground truth of targeted surprise faces, which are from the datasets but not involved in the training, and the other columns show the surprise images generated by the proposed GAN model trained with different numbers of training images.

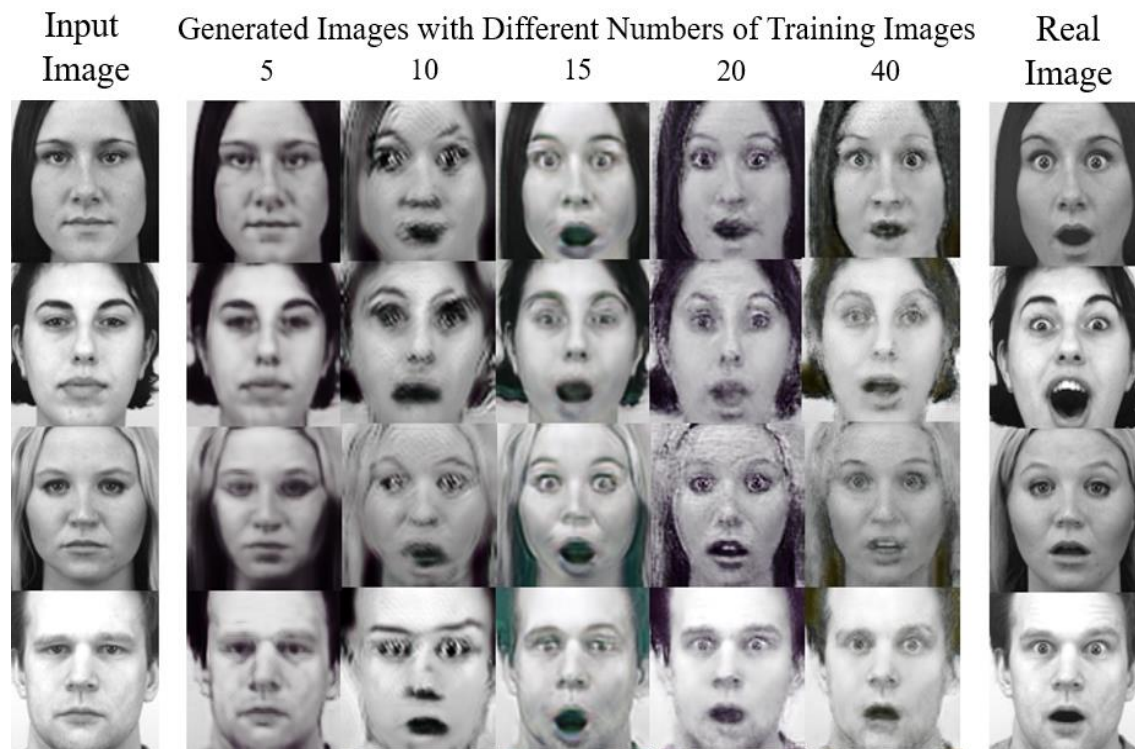


Figure 6.8: Images generated by the proposed GAN model for augmenting the surprise class of the KDEF dataset. All the synthetic results were obtained by using 40 neutral images in the source domain and up to 40 surprise images in the target domain. The real images shown in the figure were not involved in the training.

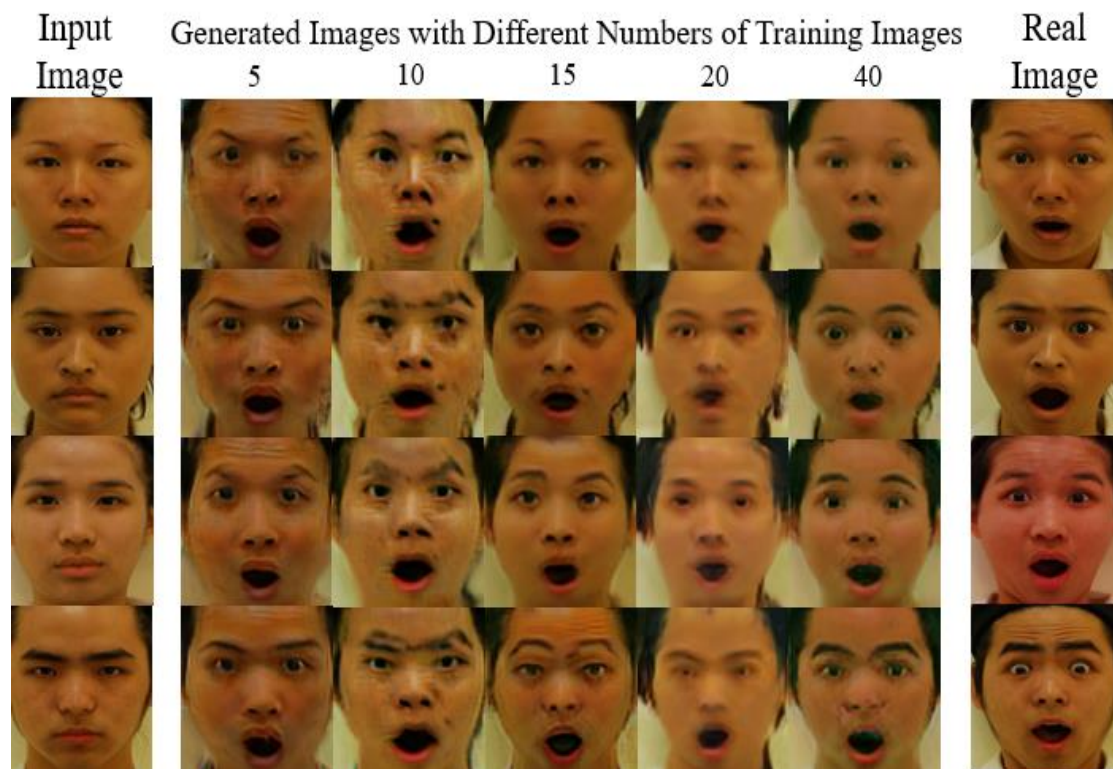


Figure 6.9: Images generated by the proposed GAN model for augmenting the surprise class of the TFEID dataset. All the synthetic results were obtained using 40 neutral images in the source domain and up to 40 surprise images in the target domain. The real images shown in the figure were not involved in the training.

It can be seen from Figure 6.8 and Figure 6.9 that the proposed GAN model produced realistic facial expression images even with a small number of training samples. However, when the number of training samples is too small, *i.e.*, 5 or 10, unexpected distortions easily happen in the generated images. This is because the GAN model cannot well learn the mapping relationship between the inputs and expected outputs from such a small number of training samples. As the number of training samples increased, the quality of the augmented images improves correspondingly. According to our own experience, more than 20 training images per expressional class is recommended in real FER applications for two reasons: Firstly, from our experimental results, the proposed GAN model trained with 20 expressional images can robustly generate facial expression images with acceptable visual quality. The generated outputs are easily blurry and cannot promise to always achieve results with expected expressions when the model was trained with less than 20 images. Furthermore, 20 images are a rational small data size that can be collected by general users. Therefore, in the remaining experiments, 20 training samples per class were used

to train the proposed GAN model for generating images with various facial expressions.

### 6.4.1.2 Visual Analysis with Different Expressional Classes

In this experiment, different facial expression images chosen from the three FER datasets were used to form small datasets to train the proposed GAN model. The training data were produced as follows: 20 training samples per expressional class were randomly chosen as the data in the target domain whilst a different number of neutral face images, which depends on the number of subjects on each dataset, were used as the data in the source domain to form a small training dataset for training the proposed GAN model. The numbers of neutral face images chosen from the TFEID, CK+ and KDEF datasets are 40, 50, and 70 respectively, which are expected to contain sufficient original characteristics of the subjects' neutral faces in the source domain.



Figure 6.10: Images generated by the proposed GAN model by augmenting the TFEID dataset, where 40 neutral images and 20 images of each emotional class were used to train the proposed GAN model for transferring neutral images to images with various facial expressions.



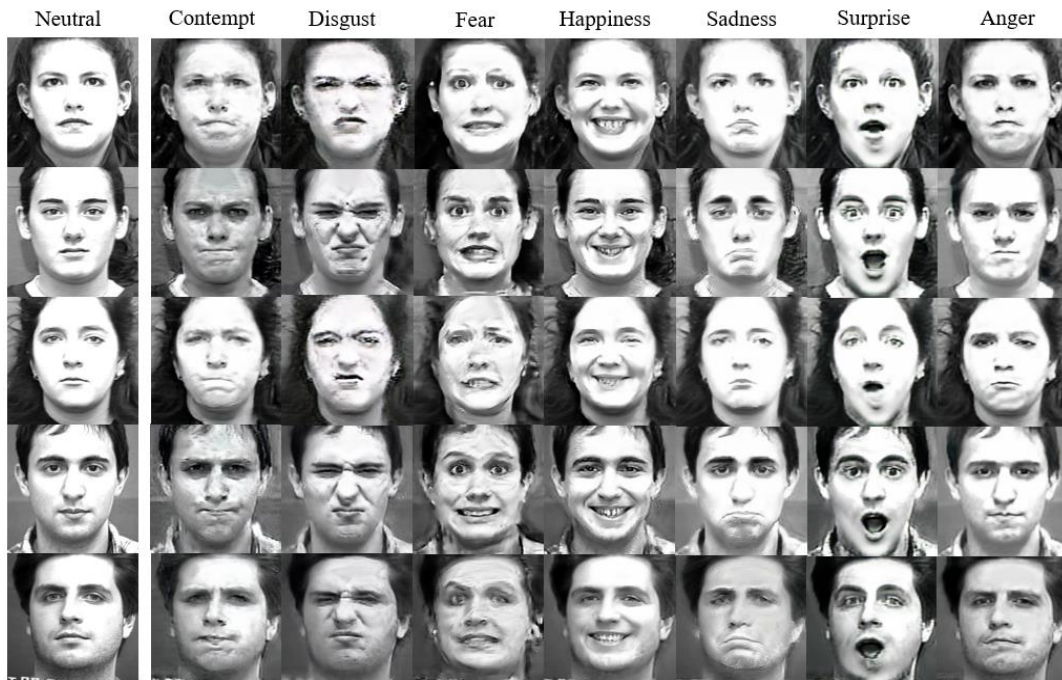


Figure 6.11: Images generated by the proposed GAN by augmenting the CK+ dataset, where 50 neutral images and 20 images of each emotional class were used to train the proposed GAN model for transferring neutral images to images with various facial expressions.



Figure 6.12: Images generated by the proposed GAN by augmenting the KDEP dataset, where 70 neutral images and 20 images of each emotional class were used to train the proposed GAN model for transferring neutral images to images with various facial expressions.

Figure 6.10 to Figure 6.12 demonstrate the synthetic results of the proposed GAN model by transferring neutral face images to targeted facial expression images, based on the three facial expression datasets respectively, where the first column shows neutral face images from the datasets that were the real data in the source domain, and the other columns show the face images generated by the proposed GAN model with 6 or 7 different targeted facial expressions. The experimental results demonstrate that all the generated face images are of appropriate expressional features and acceptable image quality. Compared to the input neutral faces, the generated images with targeted facial expressions not only contain the expected expressional features but also maintain critical face characteristics in the original neutral faces for meeting the data augmentation expectations. To sum up, the proposed GAN model trained with a small training dataset can effectively generate desirable facial expressions from neutral face images without unacceptable distortions.

### **6.4.1.3 Comparison with the State-of-the-art**

In this experiment, the state-of-the-art generative models were used to visually compare the generative results based on the same small training dataset. The selected models should follow the critical factors that the unpaired labelled images need to be applied as the input data by using unsupervised image-to-image translation methods, and their network structures are based on GANs. Therefore, the generative models for comparison are the CycleGAN [168], unsupervised image-to-image translation (UNIT) [227], multimodal unsupervised image-to-image translation (MUNIT) [228], unpaired image-to-image translation using attention-guided generative adversarial networks (AttentionGAN.v2) [229] and ours (the proposed GAN model). Firstly, UNIT learns joint data distributions between two different domains. A shared latent space is used as a critical component for image generating relying on the variational autoencoder and GAN structure. Moreover, MUNIT is an unsupervised image-to-image translation method, which additionally translates image representations from a latent space into a content space and further shares the learned information with both the source and target domain. The content space is aligned with a Gaussian distribution, and the image representations are decomposed into a content code and a style code. To transfer images between the source domain and target domain, a combination of the content code with an assigned style code can specifically control the generative style. Finally, AttentionGAN.v2 adopts attention-guided generators, which can produce attention masks and use them to fuse the generative outputs for obtaining high-quality synthetic images. The attention mechanism proposed in AttentionGAN.v2 can preserve the

background of the input images and discover the discriminative contents by producing attention masks and content masks respectively.

The number of used neutral face images in the source domain is shown in Table 6.5. Only one-way translation, from neutral faces to surprise faces, was recorded in this experiment. Twenty samples were randomly chosen from the surprise class as the training data in the target domain. The codes of CycleGAN, UNIT, MUNIT and AttentionGAN.v2 were downloaded from their official Github websites and trained with the default parameter settings, except that the batch size was reset to a small number suitable for training with small datasets.

Figure 6.13 shows the results of the state-of-the-art methods. Compared with the original input images, CycleGAN cannot effectively produce the desired expression attributes in the generated images because the expressional attributes with a larger spatial range are difficult to be learned by merely using convolutional layers. UNIT produces unexpected distortions in the generated surprise images, especially in areas of the eyes and mouth. The distortive phenomenon usually happens when feature maps fail to contain sufficient information, which is the case when UNIT was trained with a small training dataset. MUNIT performed better than UNIT for correctly transferring neutral face images to the expected facial expression images. The generated surprise images not only contain the expressional attributes but also keep the characteristics of the neutral faces. However, the images generated by MUNIT have limited expressional attributes, leading to low diversity in the generated images, which is mainly due to the insufficient expressional information in the latent space, caused by training with a small training dataset. AttentionGAN.v2 achieved competitive results but some of the facial attributes, *i.e.*, hairstyles, facial contour, shapes, *etc.*, have a large-scale shifting compared to the input neutral faces.

Compared with CycleGAN, UNIT and MUNIT, our proposed GAN model (ours) with the feature map mechanism is able to effectively extract expressional attributes from a small number of training samples with a focus on specific regions such as eyes, nose, and mouth for facial expression transfer. It also maintains the important characteristics in the neutral face images so as to avoid or reduce distortions in the generated facial expression images. It can be seen from the experimental results in Figure 6.13 that the proposed GAN model trained with a small number of samples can not only transfer neutral face images to expressional images but also preserve important facial attributes in neutral faces. This is a desirable property for image data augmentation.

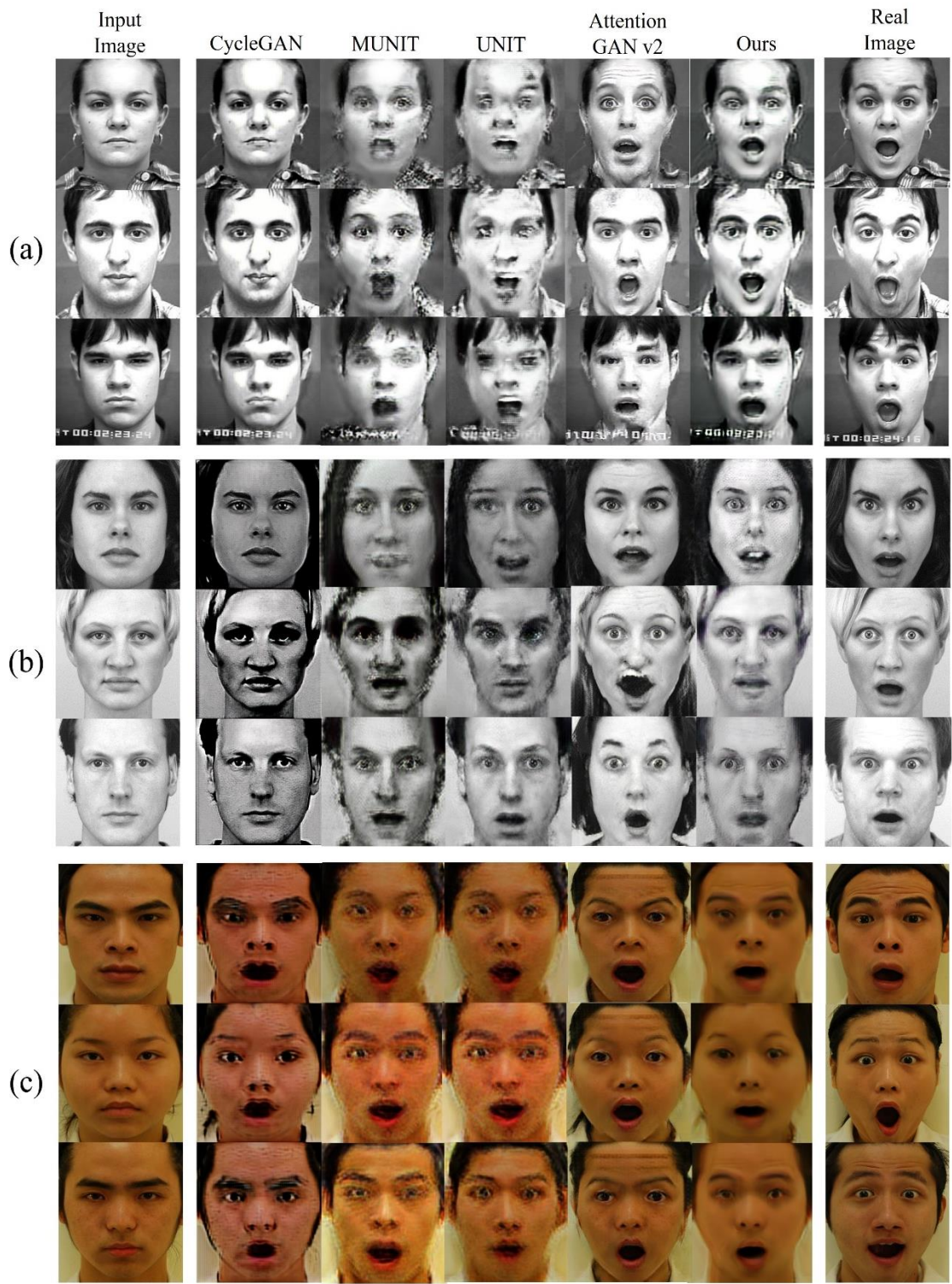


Figure 6.13: Comparisons of facial expression transfer from neutral face images (first column) to surprise expression by CycleGAN, MUNIT, UNIT, AttentionGAN.v2 and the proposed GAN model respectively: (a) CK+ dataset. (b) KDEF dataset. (c) TFEID dataset.

## 6.4.2 Quantitative Evaluation

### 6.4.2.1 Evaluation by FID and KID

In this experiment, to evaluate the facial reality and expression modification in the generated images, FID and KID are used to assess the reality between pairs of images. To be specific, the neutral face images are used as the untrained data, which are designed to transfer images from the source domain to the target domain, and then these neutral face images can be augmented by modifying neutral facial images with the expected expressions. The generated images with the desired expressional faces in different classes are directly compared to the same participants with the real expressional faces provided from the datasets for calculating the FID and KID values. A low value of FID or KID indicates a high similarity between a pair of images.

Table 6.6 shows the average FID and KID values between the real facial expression images and the corresponding facial expression images generated by CycleGAN, MUNIT, UNIT, and the proposed GAN model (ours). It can be seen that ours generally generated more photorealistic results than CycleGAN, MUNIT and UNIT. There is a special case in which CycleGAN outperformed ours on the CK+ dataset in terms of FID and KID values. However, as shown in Figure 6.13, CycleGAN did not always produce expected facial expressions but the generated images look like the input neutral face images. In our analysis, the main reason causing the deviation between the FID and KID values and visual judgment is that CycleGAN attempts to synthesise realistic results with low image distortions but ignores the importance of expressional attributes that need to be essentially transferred. Even though low distortion can achieve high realistic performance in terms of the FID and KID values, it is not beneficial for data augmentation purposes when the expected expressions are not specifically transferred in the generated results. To verify the assumption mentioned above, in the next experiment, performance enhancement in transfer learning will be utilised as another evaluation metric to confirm the assumption that using the augmented images generated from the CK+ dataset by CycleGAN in transfer learning may not promote the FER performance of CNNs.

Table 6.6: Reality scores estimated by FID and KID metrics. Lower FID and KID values indicate higher visual similarity between real and generated images.

<i>Dataset</i>	<i>Methods</i>	<i>FID</i>	<i>KID</i>
<i>CK+</i>	CycleGAN	<u><b>108.6633</b></u>	<u><b>9.096827</b></u>
	MUNIT	118.2545	17.44388
	UNIT	158.9326	20.57399
	AttentionGAN.v2	116.4804	14.28450
	Ours	114.0547	12.11755
<i>KDEF</i>	CycleGAN	163.0129	19.75672
	MUNIT	156.1264	19.55005
	UNIT	116.0834	22.2877
	AttentionGAN.v2	79.08684	<b>11.15851</b>
	Ours	<b>71.95389</b>	12.56995
<i>TFEID</i>	CycleGAN	169.167	32.1291
	MUNIT	135.3592	23.72745
	UNIT	163.4647	32.30063
	AttentionGAN.v2	94.17466	19.78435
	Ours	<b>81.17159</b>	<b>16.32507</b>

#### 6.4.2.2 Evaluation by Performance Enhancement in Image Classification

Transfer learning was conducted in this experiment, and four CNNs, *i.e.*, AlexNet, GoogLeNet, ResNet, and VGGNet, were applied as the classifiers for facial expression recognition. A small number of training images were randomly chosen from the facial expression datasets as the small training datasets, facial expression images were

generated by CycleGAN and the proposed GAN model respectively, and the FER performances of the CNNs trained with and without using the augmented facial expression images were compared to show the effect of using the augmented facial expression images on enhancing the FER performance of the CNNs.

The experimental setup was as follows: To form a small training dataset for training the proposed GAN model and the CNNs, 20 expressional images from each facial expression class were randomly chosen from each facial expression dataset, and 70 neutral images from KDEF, 50 from CK+, 40 from TFEID were randomly chosen as real data in the target domain. The remaining images in each dataset were used as validation data to evaluate the FER performance of the CNNs, which were trained with the above-formed small training datasets with or without using augmented facial expression images. To evaluate the effect of the augmented facial expression images on the FER performance of the CNNs, the augmented facial expression images were combined with the original 20 images of each facial expression class to retrain the four CNNs. For reducing the negative influence of overfitting caused by using small training datasets, the presented validation accuracies are the mean values of the best 10 runs of each CNN.

Table 6.7: Comparison of validation accuracies of four CNNs trained with 20 training samples from each class and augmented facial expression images generated by the proposed GAN and CycleGAN on the CK+ dataset. (Unit: %)

<i>Dataset</i>	<i>Accuracy(%)</i>				<i>Avg.</i>
	<i>CNNs</i>				
	<i>AlexNet</i>	<i>GoogLeNet</i>	<i>VGGNet</i>	<i>ResNet</i>	
<i>CK+</i>	71.62	64.74	69.04	70.41	68.95
<i>CK+ &amp; CycleGAN</i>	75.91	66.46	58.49	62.74	62.56
<i>CK+ &amp; our GAN</i>	81.58	77.63	80.21	78.71	79.53

Table 6.8: Comparison of validation accuracies of four CNNs trained with 20 training samples from each class and augmented facial expression images generated by the proposed GAN on the CK+, KDEF and TFEID facial expression datasets respectively. (Unit: %)

<i>Dataset</i>	<i>CNNs</i>				<i>Avg.</i>
	<i>AlexNet</i>	<i>GoogLeNet</i>	<i>VGGNet</i>	<i>ResNet</i>	
<i>CK+</i>	71.62	64.74	69.04	70.41	68.95
<i>CK+ &amp; our GAN</i>	81.58	77.63	80.21	78.71	79.53
<i>KDEF</i>	60.65	64.05	58.30	63.17	61.54
<i>KDEF &amp; our GAN</i>	82.46	83.17	85.40	82.18	83.30
<i>TFEID</i>	64.31	68.45	58.86	60.14	62.94
<i>TFEID &amp; our GAN</i>	74.83	76.76	70.17	73.32	73.77

Table 6.7 shows the validation accuracies of the four CNNs on the CK+ dataset under three situations: trained with 20 original samples from each expression class only, trained with 20 original samples from each expression class plus augmented facial expression images generated by CycleGAN, and trained with 20 original samples from each expression class plus the same amount of augmented facial expression images generated by our proposed GAN model. It can be seen that using the augmented images generated by the proposed GAN model improved the FER performance by over 10% whilst using the augmented images generated by CycleGAN actually degraded the FER performance of the CNNs. This may be because CycleGAN sometimes generated facial expression images incorrectly, as shown in Figure 6.13.

Figure 6.8 shows the validation accuracies of the four CNNs on the CK+, KDEF and TFEID datasets, with the CNNs trained with 20 original samples from each expression class only and trained with 20 original samples from each expression class plus the augmented facial expression images generated by our proposed GAN model,



respectively. It can be seen that using the facial expression images generated by our proposed GAN model to augment small training datasets improved the FER validation accuracy of the CNNs by 10% ~ 22% on average.

### 6.4.2.3 Student's T-test

Because training with small datasets is usually unstable and easily causes overfitting, to confirm the statistical significance of the FER performance improvement by using the augmented facial expression images generated by the proposed GAN model, the student's t-test was conducted based on the FER performance data from 10 runs of each CNN. The null hypothesis is that there is no significant difference between the validation accuracies of the CNNs trained with or without using the augmented facial expression images. The significant level  $\alpha$  is set to 0.05 as usual, which implies when the p-value of the student's t-test is smaller than 0.05, the null hypothesis will be rejected.

Table 6.9 shows the p-values of the student's t-test for comparing the performances of the four CNNs trained with and without using the augmented facial expression images generated by the proposed GAN model. It can be seen that all the p-values for the four CNNs are smaller than 0.05, which leads to an entire rejection of the null hypothesis and shows that the validation accuracies obtained with augmented training data are significantly greater than those without using augmented training data.

Table 6.9: The p-values of the student's t-test for comparing the performances of four CNNs trained with and without using augmented facial expression images respectively.

	<i>CK+</i>	<i>KDEF</i>	<i>TFEID</i>
<i>AlexNet</i>	3.65e-04	1.33e-12	3.29e-04
<i>VGGNet</i>	0.0026	5.79e-13	5.80e-04
<i>GoogLeNet</i>	0.0206	6.49e-12	7.84e-06
<i>ResNet</i>	0.0301	1.51e-14	1.26e-06

## 6.5 Conclusion

In this chapter, a many-to-many image translation method is developed to synthesise augmented data from neutral face images to expressional images with the proposed GAN model. Experimental results with a small number of training images show that the augmented images with our proposed GAN significantly improve the accuracy by using the CNNs as classifiers for FER tasks. It is also demonstrated that our proposed GAN model is more functional for correctly transferring neutral face images to photorealistic expression images than other many-to-many image translation methods when a small number of facial expression images are involved as the training data in the target domain. Consequently, the proposed GAN model can not only mitigate the negative effects of training with a small training dataset but also enlarge the potential applications of data augmentation, especially if a large amount of expressional data is difficult to be collected.

# Chapter 7

## Conclusions and Future Work

### 7.1 Summary of Contributions

A common problem with deep learning is addressed in this thesis, which is caused by insufficient labelled data for training deep learning models. Building novel GAN models for augmenting small training datasets to promote the classification performance of deep learning models is the main objective of this thesis. Three GAN models based on different input and output image mapping relationships, including one-to-many mapping, one-to-one mapping and many-to-many mapping, are proposed respectively. According to the experimental results using a small number of images as the training dataset, the images generated by the proposed GAN models are of high photorealistic quality and good diversity, but with low distortions. The synthetic images can be used as augmented data to mitigate the negative effects caused by training with limited feature information (*e.g.*, insufficient labelled training data, sparse conditional inputs, *etc.*) and further promote the performance of image classification when CNNs are applied as classifiers with a small training dataset. Detailed contributions of this thesis work are summarised as follows.

In Chapter 4, a novel GAN model is proposed with a perturbation mechanism and a novel network framework for synthesising many diverse images from one target image, which aims to solve the data scarcity problem in training deep learning models. The proposed model is designed to augment a small number of training images for the purpose of enhancing the image classification performance of deep neural networks. Experimental results show that the augmented dataset can effectively enhance the image classification performance of CNNs through transfer learning. It can be concluded that the proposed GAN model, which can generate many diverse images from one pattern only, can significantly improve the image classification performance of deep neural networks originally trained with a very small number of training samples.

As the second contribution presented in Chapter 5, a new condition-based GAN framework is proposed for edge-to-image translation based on a small set of training data, which can synthesise photorealistic diverse facial images using incomplete edges as conditional inputs for data augmentation purposes. To solve the problem of training condition-based GANs with a small dataset, an interim domain for refining images is introduced in the proposed condition-based GAN, which can effectively reduce

unexpected distortions and improve the quality of the generated images. Experimental results have demonstrated that blending segmentation masks and regional binary images as refined reference images can reduce the distortions in generative components to efficiently learn realistic features from a small number of training images. Compared with the existing edge-to-image translation methods, the proposed condition-based GAN can not only automatically transfer incomplete conditional edges to reference images with more facial features in the interim domain but also effectively reduce unexpected distortions caused by small training data. Compared to directly translating images from one domain to another domain, the proposed method can have a more comprehensive view to generate more photorealistic edge-to-image translation results when using various incomplete conditional edges for data augmentation. More informative reference images can be constructed in the interim domain from incomplete edge inputs to integrate useful facial components. The proposed condition-based GAN trained with a small dataset can synthesise various photorealistic facial images by manipulating conditional edges or using hand-drawn facial sketches to synthesise diverse augmented image data. Contrasted to the existing one-to-one image translation methods, the images generated by the proposed condition-based GAN have less distortion and more diversity, which is desirable for data augmentation purposes.

Finally, in Chapter 6, a novel GAN model is proposed for generating facial expression images based on a small number of training facial expression samples. The proposed GAN model adopts a feature map mechanism to extract useful spatial information related to targeted facial expressions from a small number of training samples by a feature extractor, which can mitigate unexpected distortions to generate photorealistic images for meeting data augmentation requirements on many-to-many mapping relationship. The experimental results show that the proposed GAN model can not only successfully generate desired facial expression images from a small number of facial expression samples but also maintain the original characters in the corresponding input neutral face images. It is also demonstrated that using the synthetic facial expression images generated by the proposed GAN model can significantly improve the FER performance of CNNs. The application of this study is limited to FER tasks, but it can be extended to other classification problems.

## **7.2 Limitations and Future Work**

For the proposed GAN model presented in Chapter 4, the image resolution and data reality can be further improved by further exploring the model structure and learning algorithm. Several limitations need to be considered to promote synthetic quality in the

future. Firstly, the diversity of the generated images is still dependent on traditional augmentation methods by using transformation matrices in the perturbation mechanism, although the proposed GAN model has demonstrated the capability of creating more diverse results than traditional techniques in the experiments. An improved GAN model with a more powerful feature extractor could extract complex features to improve the diversity of the generated images. Secondly, the quality of the generated images based on a small set of training images is to be further improved. Advanced methods need to be developed to improve the learning efficiency and avoid mode collapse and gradient vanishing problems in training with small datasets.

For the work presented in Chapter 5, due to the limited GPU computing facilities available for our experiments, it is hard to optimise the hyperparameters of the tested models, and the performance evaluation is based on the comparison with two state-of-the-art methods only. More extensive comparative studies would be desirable in future research to draw more reliable conclusions. The advantage of the proposed condition-based GAN framework over the existing methods becomes less obvious when the number of training samples is relatively large. For future work, the interim domain could be improved, so that the proposed condition-based GAN framework would also significantly outperform existing methods for image data augmentation when a reasonably large number of training images are available. On the other side, although the experimental results have demonstrated the effectiveness of our proposed method in image classification, there are still some limitations for real applications of image data augmentation. For instance, more diverse results can be created from the undefined areas in the conditional features. Besides, regarding generating photorealistic results based on a small training dataset, the mode collapse problem cannot be comprehensively eliminated in our experiments. Advanced normalization methods and training strategies could be explored in future work to address the mentioned problems.

Referring to the experimental results from visual analysis and quantitative comparison in Chapter 6, the effectiveness of the many-to-many image translation using the proposed GAN model has been demonstrated. However, some limitations are also discovered in the experiments, and the proposed model could be improved based on the following ideas. Firstly, the diversity in the generated facial expression images is still very limited when only a small number of labelled samples are used because the proposed GAN model pays foremost attention to the reality of the generated images in terms of facial expressions. Data similarity and diversity are two critical factors affecting the performance of data augmentation, an advanced GAN model should be able to generate more diverse facial expressions for good data augmentation performance with a small number of training samples. Secondly, compared with other facial synthesis tasks, the good quality of fake expressions is difficult to be generated

from a small amount of training data. Expressional attributes commonly have a large range of spatial relationships, and the correct mapping positions and attributes need to be identified by improving the feature map mechanism in the proposed GAN model to mitigate the distortions in the generated images. Thirdly, the mode collapse problem is not completely eliminated in this proposed GAN model either, which degrades the efficiency of data augmentation. Finding the optimal values is still difficult and has to rely on high computational capabilities, advanced techniques to optimise hyperparameters could be adopted for mitigating the overfitting problem, especially when training without sufficient training data. Furthermore, based on the experience from our experiments, image pre-processing and facial normalisation are critical in the proposed model for reducing synthetic blurs and distortions caused by training with small FER datasets, which could bring about a limitation to the proposed model. If the input facial images cannot be aligned as the images to a mapped position, (*i.e.*, front faces or profile faces), the proposed model may be inefficient to discover the correct mapping relationship from small training datasets and thus difficult to generate desired expressional features. Finally, because of only limited features being learnt, the proposed model may be hard to acquire comprehensive knowledge to deal with all the expressional attributes relying on a small training dataset, which generally causes unexpected distortions in the generated images.

In addition, a series of general limitations in deep learning models still need to be further analysed in real applications, *i.e.*, the training efficiency, model complexity, processing speed, computational cost, improvement level, capabilities of generalization, regularisation, optimisation, universality, and so on. All in all, it is continuous work to explore advanced GAN structures and develop applicable learning algorithms, which can synthesise high-quality and diverse images from a small number of training samples for promoting the performance of deep learning models.

# Bibliography

- [1] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, “Applications of machine learning to machine fault diagnosis: A review and roadmap,” *Mechanical Systems and Signal Processing (MSSP)*, Apr. 2020, Art. no. 106587.
- [2] X.-W. Chen and X. Lin, “Big data deep learning: Challenges and perspectives,” *IEEE Access*, vol. 2, pp. 514–525, May 2014.
- [3] A. Thakur, D. Thapar, P. Rajan, and A. Nigam, “Deep metric learning for bioacoustic classification: Overcoming training data scarcity using dynamic triplet loss,” *Journal of the Acoustical Society of America*, vol. 146, no. 1, pp. 534–547, 2019.
- [4] S. Akila, and U. Srinivasulu Reddy, “Data imbalance: Effects and solutions for classification of large and highly imbalanced data,” in *Proceedings of International Conference on Rock Engineering, Classification and Testing*, 2016, pp. 28–34.
- [5] S. O’Gara and K. McGuinness, “Comparing data augmentation strategies for deep image classification,” in *Proceedings of Irish Machine Vision and Image Processing (IMVIP)*, 2019.
- [6] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, “Data synthesis based on generative adversarial networks,” in *Proceedings of International Conference on Very Large Data Bases Endowment (VLDB Endowment)*, vol. 11, no. 10, 2018, pp. 1071–1083.
- [7] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, Jul. 2019.
- [8] L. Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” *arXiv preprint arXiv:1712.04621*, 2017. [Online]. Available: <https://arxiv.org/abs/1712.04621>
- [9] O. Serradilla, E. Zugasti, J. Rodriguez, and U. Zurutuza, “Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects,” *Applied Intelligence*, pp. 10934–10964 2022.
- [10] M. Awiszus, F. Schubert, and B. Rosenhahn, “TOAD-GAN: Coherent style level generation from a single example,” in *Proceedings of AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 16, no. 1, 2020, pp. 10–16.
- [11] J. Kim, H. Zeng, D. Ghadiyaram, S. Lee, L. Zhang, and A. C. Bovik, “Deep convolutional neural models for picture-quality prediction: Challenges and solutions to data-driven image quality assessment,” *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 130–141, 2017.

- [12] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. C. Courville, “Augmented CycleGAN: Learning many-to-many mappings from unpaired data,” in *Proceedings of International Conference on Machine Learning (ICML)*, 2018, pp. 195–204.
- [13] Y. Shen, P. Luo, J. Yan, X. Wang, and X. Tang, “FaceID-GAN: Learning a symmetry three-player GAN for identity-preserving face synthesis,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 821–830.
- [14] H. Alqahtani, M. Kavakli-Thorne, and G. Kumar, “Applications of generative adversarial networks (GANs): An updated review,” *Archives of Computational Methods in Engineering*, vol. 28, no. 2, pp. 525–552, 2021.
- [15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proceedings of Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680.
- [16] N.-T. Tran, V.-H. Tran, N.-B. Nguyen, T.-K. Nguyen, and N.-M. Cheung, “On data augmentation for GAN training,” *IEEE Transactions on Image Processing*, vol. 30, pp. 1882–1897, 2021.
- [17] C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. N. Gunn, A. Hammers, D. A. Dickie, M. del C. Valdés Hernández, J. M. Wardlaw, and D. Rueckert, “GAN augmentation: Augmenting training data using generative adversarial networks,” *arXiv preprint arXiv:1810.10863*, 2018. [Online]. Available: <https://arxiv.org/abs/1810.10863>
- [18] J. Chen, M. Zhong, J. Li, D. Wang, T. Qian, and H. Tu, “Effective deep attributed network representation learning with topology adapted smoothing,” *IEEE Transactions on Cybernetics*, pp. 1–12, 2021.
- [19] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, “A literature review and classification of recommender systems research,” *Expert Systems with Applications*, vol. 39, no. 11, pp. 10059–10072, 2012.
- [20] P. P. Shinde and S. Shah, “A review of machine learning and deep learning applications,” in *Proceedings of Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1–6.
- [21] M. A. Nielsen, *Neural Networks and Deep Learning*, vol. 1. Determination Press, 2014.
- [22] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, “Deep neural networks and tabular data: A survey,” *arXiv preprint arXiv:2110.01889*, 2021.
- [23] E. Breck, N. Polyzotis, S. Roy, S. Whang, and M. Zinkevich, “Data validation



- for machine learning,” in *Proceedings of Conference on Systems and Machine Learning (SysML)*, 2019.
- [24] Y. Li, J. Yang, Z. Zhang, J. Wen, and P. Kumar, “Healthcare data quality assessment for cybersecurity intelligence,” *IEEE Transactions on Industrial Informatics*, pp. 841–848, 2022.
- [25] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, “State-of-the-art deep learning: Evolving machine intelligence toward tomorrow’s intelligent network traffic control systems,” *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2432–2455, May 2017.
- [26] G. Kostopoulos and S. Kotsiantis, “Exploiting semi-supervised learning in the education field: A critical survey,” *Advances in Machine Learning/Deep Learning-Based Technologies*, pp. 79–94, 2022.
- [27] W. Liu and S.-F. Chang, B, “Robust multi-class transductive learning with graphs,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 381–388.
- [28] L. Schmarje, M. Santarossa, S.-M. Schröder, and R. Koch, “A survey on semi-, self- and unsupervised learning for image classification,” *IEEE Access*, vol. 9, pp. 82146–82168, 2021.
- [29] Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, “The history began from AlexNet: A comprehensive survey on deep learning approaches,” *arXiv preprint arXiv:1803.01164*, 2018. [Online]. Available: <https://arxiv.org/abs/1803.01164>
- [30] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” *arXiv preprint arXiv:2006.09882*, Jun. 2020, [Online]. Available: <http://arxiv.org/abs/2006.09882>.
- [31] Z. Chen, H. Zhang, W. G. Hatcher, J. Nguyen, and W. Yu, “A streaming-based network monitoring and threat detection system,” in *Proceedings of IEEE International Conference on Software Engineering Research, Management and Applications (SERA)*, Jun. 2016, pp. 31–37.
- [32] J. selvakumar, A. Lakshmi, and T. Arivoli, “Brain tumor segmentation and its area calculation in brain MR images using K-mean clustering and Fuzzy C-mean algorithm,” in *Proceedings of IEEE International Conference on Advances in Engineering, Science and Management (ICAESM)*, March 2012, pp. 186–190.
- [33] X. Liu, Z. Deng, and Y. Yang, “Recent progress in semantic image segmentation,” *Artificial Intelligence Review*, pp. 1–18, 2018.
- [34] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas,

- “Reinforcement learning with augmented data,” *arXiv preprint arXiv:2004.14990*, 2020.
- [35] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep reinforcement learning: A brief survey,” *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, Nov. 2017.
- [36] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *Proceedings of International Conference on Engineering and Technology (ICET)*, Aug. 2017, pp. 1–6.
- [37] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” *IEEE Symposium on Security and Privacy*, 2017, pp. 39–57.
- [38] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representations using convolutional neural networks,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1717–1724.
- [39] L. Rice, E. Wong, and J. Z. Kolter, “Overfitting in adversarially robust deep learning,” in *Proceedings of International Conference on Machine Learning (ICML)*, Jul. 2020, pp. 8093–8104.
- [40] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, “Learning pooling for convolutional neural network,” *Neurocomputing*, vol. 224, pp. 96–104, Feb. 2017.
- [41] Y.-L. Boureau, J. Ponce, and Y. LeCun, “A theoretical analysis of feature pooling in visual recognition,” in *Proceedings of International Conference on Machine Learning (ICML)*, 2010, pp. 111–118.
- [42] P. Baldi and P. J. Sadowski, “Understanding dropout,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2013, pp. 2814–2822.
- [43] X. Liang, L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, and T.-Y. Liu, “R-drop: Regularized dropout for neural networks,” in *Proceedings of Neural Information Processing Systems (NIPS)*, 2021, pp. 10890–10905.
- [44] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How does batch normalization help optimization?,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2018, pp. 2483–2493.
- [45] C. Laurent, G. Pereyra, P. Brakel, Y. Zhang, and Y. Bengio, “Batch normalized recurrent neural networks,” in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016, pp. 2657–2661.
- [46] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proceedings of International Conference on Machine Learning (ICML)*, vol. 30, 2013, p. 3.

- [47] S. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, “Impact of fully connected layers on performance of convolutional neural networks for image classification,” *Neurocomputing*, vol. 378, pp. 112–119, 2020.
- [48] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” *arXiv preprint arXiv:1811.03378*, 2018. [Online]. Available: <https://arxiv.org/abs/1811.03378>
- [49] S. Sharma, “Activation functions in neural networks,” *Towards Data Science*, vol. 6, pp. 1–7, Sep. 2017.
- [50] Y. Zhong, W. Deng, J. Hu, D. Zhao, X. Li, and D. Wen, “SFace: Sigmoid-constrained hypersphere loss for robust face recognition,” *IEEE Transactions on Image Processing*, vol. 30, pp. 2587–2598, 2021.
- [51] S. H. Wang and Y. Chen, Y, “Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique,” *Multimedia Tools and Applications*, vol. 79, no. 21, pp. 15117–15133, Dec. 2020.
- [52] X. Kang, S. Li, and J. A. Benediktsson, “Spectral-spatial hyperspectral image classification with edge-preserving filtering,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2666–2677, May 2014.
- [53] A. de Brébisson and P. Vincent, “An exploration of softmax alternatives belonging to the spherical loss family,” *arXiv preprint arXiv:1511.05042*, 2015. [Online]. Available: <https://arxiv.org/abs/1511.05042>
- [54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2013, pp. 1097–1105.
- [55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research (JMLR)*, vol. 1, no. 60, p. 11, 2014.
- [56] G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for LVCSR using rectified linear units and dropout,” in *Processing of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 8609–8613.
- [57] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.
- [58] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in *Proceedings of International Conference on 3D Vision*, Oct. 2016, pp. 239–248.
- [59] D. Mishkin, N. Sergievskiy, and J. Matas, “Systematic evaluation of

- convolution neural network advances on the ImageNet,” *Computer Vision and Image Understanding*, vol. 161, pp. 11–19, 2017.
- [60] M. S. Majib, M. M. Rahman, T. M. S. Sazzad, N. I. Khan, and S. K. Dey, “VGG-SCNet: A VGG net-based deep learning framework for brain tumor detection on MRI images,” *IEEE Access*, vol. 9, pp. 116942–116952, 2021.
- [61] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.
- [62] T.-J. Yang, Y.-H. Chen, and V. Sze, “Designing energy-efficient convolutional neural networks using energy-aware pruning,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2017, pp. 5687–5695.
- [63] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, “Deep neural network concepts for background subtraction: A systematic review and comparative evaluation,” *Neural Networks*, vol. 117, pp. 8–66, Sep. 2019.
- [64] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari, “A state-of-the-art survey on deep learning theory and architectures,” *Electronics*, vol. 8, no. 3, p. 292, Mar. 2019.
- [65] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [66] A. Khamparia and K. M. Singh, “A systematic review on deep learning architectures and applications,” *Expert Systems*, vol. 36, no. 3, pp. 1–22, 2019.
- [67] S. LaValle, E. Lesser, R. Shockley, M. S. Hopkins, and N. Kruschwitz, “Big data, analytics and the path from insights to value,” *MIT Sloan Management Review*, vol. 52, no. 2, p. 21, 2011.
- [68] S. Rezaei and X. Liu, “Deep learning for encrypted traffic classification: An overview,” *IEEE Communications Magazine*, vol. 57, no. 5, pp. 76–81, May 2019.
- [69] M. L. Hutchinson, E. Antono, B. M. Gibbons, S. Paradiso, J. Ling, and B. Meredig, “Overcoming data scarcity with transfer learning,” *arXiv preprint arXiv:1711.05099*, 2017. [Online]. Available: <http://arxiv.org/abs/1711.05099>
- [70] J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification,” *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, Mar. 2017.
- [71] R. Hataya, J. Zdenek, K. Yoshizoe, and H. Nakayama, “Meta approach to data

- augmentation optimization,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2574–2583.
- [72] M. Lango and J. Stefanowski, “What makes multi-class imbalanced problems difficult? An experimental study,” *Expert Systems with Applications*, vol. 199, 2022, Art. no. 116962.
- [73] S. Maheshwari, R. Jain, and R. Jadon, “A review on class imbalance problem: Analysis and potential solutions,” *International Journal of Computer Science Issues (IJCSI)*, vol. 14, no. 6, pp. 43–51, 2017.
- [74] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Hussain, “Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study,” *IEEE Access*, vol. 4, pp. 7940–7957, 2016.
- [75] Y. Chen, R. Chang, and J. Guo, “Effects of data augmentation method borderline-SMOTE on emotion recognition of EEG signals based on convolutional neural network,” *IEEE Access*, vol. 9, pp. 47491–47502, 2021.
- [76] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010
- [77] W. Li, R. Huang, J. Li, Y. Liao, Z. Chen, G. He, R. Yan, and K. Gryllias, “A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges,” *Mechanical Systems and Signal Processing*, vol. 167, 2022, Art. no. 108487.
- [78] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in *Proceedings of International Conference on Artificial Neural Networks (ICANN)*, Oct. 2018, pp. 270–279.
- [79] S. Deepak and P. M. Ameer, “Brain tumor classification using deep CNN features via transfer learning,” *Computers in Biology and Medicine*, vol. 111, Aug. 2019, Art. no. 103345.
- [80] H. Wu and S. Prasad, “Semi-supervised deep learning using pseudo labels for hyperspectral image classification,” *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018.
- [81] J. E. van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” *Machine Learning*, vol. 109, no. 2, pp. 373–440, Feb. 2020.
- [82] Z. Chen, Y. Fu, Y.-X. Wang, L. Ma, W. Liu, and M. Hebert, “Image deformation meta-networks for one-shot learning,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 8680–8689.
- [83] J. Snell, K. Swersky, and R. S. Zemel, “Prototypical networks for few-shot learning,” in *Proceedings of Conference on Neural Information Processing*

- Systems (NeurIPS)*, 2017, pp. 4077–4087.
- [84] S. Ravi and H. Larochelle, “Optimization as a model for few-shot learning,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2017, pp. 1–11.
  - [85] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez, “The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3234–3243.
  - [86] A. Mikolajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” in *Proceedings of International Interdisciplinary PhD Workshop (IIPhDW)*, May 2018, pp. 117–122.
  - [87] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.
  - [88] A. Oliver, A. Odena, C. Raffel, E. Cubuk, and I. Goodfellow, “Realistic evaluation of deep semi-supervised learning algorithms,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2018, pp. 3235–3246.
  - [89] A. Power, Y. Burda, H. Edwards, I. Babuschkin, and V. Misra, “Grokking: Generalization beyond overfitting on small algorithmic datasets,” *arXiv preprint arXiv:2201.02177*, 2022.
  - [90] S. D. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2613–2617.
  - [91] E. Lashgari, D. Liang, and U. Maoz, “Data augmentation for deep-learning-based electroencephalography,” *Journal of Neuroscience Methods*, vol. 346, Dec. 2020, Art. no. 108885.
  - [92] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool, “Deep convolutional neural networks and data augmentation for acoustic event detection,” in *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, 2016, pp. 2982–2986.
  - [93] S. Afzal, M. Maqsood, F. Nazir, U. Khan, F. Aadil, K. M. Awan, I. Mehmood, and O.-Y. Song, “A data augmentation-based framework to handle class imbalance problem for Alzheimer’s stage detection,” *IEEE Access*, vol. 7, pp. 115528–115539, 2019.
  - [94] F. Calimeri, A. Marzullo, C. Stamile, and G. Terracina, “Biomedical data augmentation using generative adversarial neural networks,” in *Proceedings of*

- International Conference on Artificial Neural Networks (ICANN)*, Oct. 2017, pp. 626–634.
- [95] A. Bartoli, “Groupwise geometric and photometric direct image registration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 30, no. 12, pp. 2098–2108, 2008.
- [96] N. E. Khalifa, M. Loey, and S. Mirjalili, “A comprehensive survey of recent trends in deep learning for digital images augmentation,” *Artificial Intelligence Review*, vol. 55, pp. 2351–2377, Mar. 2022.
- [97] Z. Hussain, F. Gimenez, D. Yi, and D. Rubin, “Differential data augmentation techniques for medical imaging classification tasks,” in *Proceedings of AMIA annual symposium proceedings*, 2017, pp. 979–984.
- [98] M. M. Krell and S. K. Kim, “Rotational data augmentation for electroencephalographic data,” in *Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, Jul. 2017, pp. 471–474.
- [99] C.-Y. Lin, M. Wu, J. Bloom, I. Cox, M. Miller, and Y. Lui, “Rotation, scale, and translation resilient watermarking for images,” *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 767–782, May 2001.
- [100] G. Zhao and J. Yuan, “Mining and cropping common objects from images,” in *Proceedings of ACM Multimedia*, 2010, pp. 975–978.
- [101] L. Nataraj, A. Sarkar, and B. S. Manjunath, “Improving re-sampling detection by adding noise,” *SPIE Electronic Imaging, Media Security and Forensics*, vol. 7541, Jan. 2010, Art. no. 75410I.
- [102] L. Hua, M. Yu, Z. He, R. Tu, and G. Jiang, “CPC-GSCT: visual quality assessment for coloured point cloud based on geometric segmentation and colour transformation,” *IET Image Processing*, vol. 16, no. 4, pp. 1083–1095, 2022.
- [103] Y. Wang, X. Wei, X. Tang, H. Shen, and L. Ding, “CNN tracking based on data augmentation,” *Knowledge-Based Systems*, vol. 194, Apr. 2020, Art. no. 105594.
- [104] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random erasing data augmentation,” *arXiv preprint arXiv:1708.04896*, 2017. [Online]. Available: <https://arxiv.org/abs/1708.04896>
- [105] G. Chen, T. Zhang, J. Lu, and J. Zhou, “Deep meta metric learning,” in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 9547–9556.
- [106] J. Lemley, S. Bazrafkan, and P. Corcoran, “Smart augmentation learning an optimal data augmentation strategy,” *IEEE Access*, vol. 5, pp. 5858–5869, 2017.

- [107] P. Enkvetchakul and O. Surinta, “Effective data augmentation and training techniques for improving deep learning in plant leaf disease recognition,” *Applied Science and Engineering Progress*, 2021.
- [108] M.-Y. Liu and O. Tuzel, “Coupled generative adversarial networks,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2016, pp. 469–477.
- [109] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, and M. Michalski, “Medical image synthesis for data augmentation and anonymization using generative adversarial networks,” in *Proceedings of International Workshop Simulation and Synthesis in Medical Imaging*, 2018, pp. 1–11.
- [110] D. Lee, C. Kim, S. Kim, M. Cho, and W.-S. Han, “Autoregressive Image Generation using Residual Quantization.” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11523–11532.
- [111] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative adversarial networks: An overview,” *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, Jan. 2018.
- [112] A. Borji, “Pros and cons of GAN evaluation measures: New developments,” *Computer Vision and Image Understanding*, vol. 215, Jan. 2022, Art. no. 103329.
- [113] V. Sorin, Y. Barash, E. Konen, and E. Klang, “Creating artificial images for radiology applications using generative adversarial networks (GANs) – A systematic review,” *Academic Radiology*, vol. 27, no. 8, pp. 1175–1185, 2020.
- [114] B. Neyshabur, S. Bhojanapalli, and A. Chakrabarti, “Stabilizing GAN training with multiple random projections,” 2017, *arXiv preprint arXiv:1705.07831*. [Online]. Available: <http://arxiv.org/abs/1705.07831>
- [115] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, “Differentiable augmentation for data-efficient GAN training,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2020.
- [116] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2016, pp. 1–16.
- [117] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014. [Online]. Available: <https://arxiv.org/abs/1411.1784>
- [118] W. R. Tan, C. S. Chan, H. Aguirre, and K. Tanaka, “ArtGAN: Artwork synthesis with conditional categorical GANs,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 3760–



3764.

- [119] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier GANs,” in *Proceedings of International Conference on Machine Learning (ICML)*, vol. 70, Aug. 2017, pp. 2642–2651.
- [120] H. Bao, Z. Hua, H. Li, M. Chen and B. Bao, “Memristor-based hyperchaotic maps and application in auxiliary classifier generative adversarial nets,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5297–5306, 2022.
- [121] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow, “Adversarial autoencoders,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2016, pp. 1–16.
- [122] J. Donahue, P. Krähenbühl, and T. Darrell, “Adversarial feature learning,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2017, pp. 1–18.
- [123] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “It takes (only) two: Adversarial generator-encoder networks,” in *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018, pp. 1250–1257.
- [124] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.
- [125] J. Lee, I. Lee, and J. Kang, “Self-attention graph pooling,” in *Proceedings of International Conference on Machine Learning (ICML)*, 2019, pp. 3734–3743.
- [126] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” in *Proceedings of International Conference on Machine Learning (ICML)*, 2019, pp. 7354–7363.
- [127] Z. He, M. Kan, J. Zhang, and S. Shan, “PA-GAN: Progressive attention generative adversarial network for facial attribute editing,” *arXiv preprint arXiv:2007.05892*, 2020. [Online]. Available: <https://arxiv.org/abs/2007.05892>
- [128] L. J. Ratliff, S. A. Burden, and S. S. Sastry, “Characterization and computation of local Nash equilibria in continuous games,” in *Proceedings of Annual Allerton Conference on Communication, Control and Computing, Monticello*, 2013, pp. 917–924.
- [129] A. Rizwan, A. Abu-Dayya, F. Filali, and A. Imran, “Addressing data sparsity with GANs for Multi-fault diagnosing in emerging cellular networks,” in *Proceedings of International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, 2022, pp. 318–323.
- [130] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proceedings of International Conference on Machine Learning (ICML)*, vol. 70, 2017, pp. 214–223.

- [131] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of Wasserstein GANs,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5767–5777.
- [132] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, “On the effectiveness of least squares generative adversarial networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 12, pp. 2947–2960, Dec. 2019.
- [133] H. Thanh-Tung and T. Tran, “Catastrophic forgetting and mode collapse in GANs,” in *Proceedings of International Joint Conference on Neural Networks. (IJCNN)*, 2020, pp. 1–10.
- [134] A. Srivastava, L. Valkoz, C. Russell, M. U. Gutmann, and C. Sutton, “VEEGAN: Reducing mode collapse in GANs using implicit variational learning,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 3308–3318.
- [135] A. T. Gnanha, W. Cao, X. Mao, S. Wu, H. S. Wong, and Q. Li, “The residual generator: An improved divergence minimization framework for GAN,” *Pattern Recognition*, vol. 121, Jan. 2022, Art. no. 108222.
- [136] M. Baiocchi, G. D. Bari, V. Poggioni, and C. A. C. Coello, “Smart multi-objective evolutionary GAN,” in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2021, pp. 2218–2225.
- [137] J. Su, “GAN-QP: A novel GAN framework without gradient vanishing and Lipschitz constraint,” *arXiv preprint arXiv:1811.07296*, 2018. [Online]. Available: <http://arxiv.org/abs/1811.07296>
- [138] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2107–2116.
- [139] Y. Hu, A. E. G. Huber, J. Anumula, and S. Liu, “Overcoming the vanishing gradient problem in plain recurrent networks,” *arXiv preprint arXiv:1801.06105*, 2018. [Online]. Available: <http://arxiv.org/abs/1801.06105>
- [140] A. Ghosh, V. Kulharia, V. P. Namboodiri, P. H. S. Torr, and P. K. Dokania, “Multi-agent diverse generative adversarial networks,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018, pp. 8513–8521.
- [141] Y. Yaz, C.-S. Foo, S. Winkler, K.-H. Yap, G. Piliouras, and V. Chandrasekhar, “The unusual effectiveness of averaging in GAN training,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.
- [142] D. Xu, Y. Wu, S. Yuan, L. Zhang, and X. Wu, “Achieving causal fairness

- through generative adversarial networks,” in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 1452–1458.
- [143] J. Lorraine, P. Vicol, and D. Duvenaud, “Optimizing millions of hyperparameters by implicit differentiation,” in *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2020, pp. 1540–155.
- [144] D. Saxena, J. -N. Cao, “Generative adversarial networks (GANs): challenges, solutions, and future directions,” *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–42, 2021.
- [145] C. He, S. H. Huang, R. Cheng, K. C. Tan, and Y. Jin, “Evolutionary multiobjective optimization driven by generative adversarial networks (GANs),” *IEEE Transactions on Cybernetics*, pp. 1–14, 2020.
- [146] Q. Lei, J. D. Lee, A. G. Dimakis, and C. Daskalakis, “SGD learns one-layer networks in WGANs,” *arXiv preprint arXiv:1910.07030*, 2019. [Online]. Available: <https://arxiv.org/abs/1910.07030>
- [147] Z. Zhang, “Improved Adam optimizer for deep neural networks,” in *Proceedings of IEEE/ACM International Symposium on Quality of Service (IWQoS)*, 2018, pp. 1–2.
- [148] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, “Understanding batch normalization,” in *Proceedings of Neural Information Processing Systems (NIPS)*, 2018, pp. 7705–7716.
- [149] K. Kurach, M. Lucic, X. Zhai, M. Michalski, and S. Gelly, “A large-scale study on regularization and normalization in GANs,” in *Proceedings of International Conference on Machine Learning (ICML)*, 2019, pp. 3581–3590.
- [150] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, “Are GANs created equal? a large-scale study,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2018, pp. 700–709.
- [151] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training GANs,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2016, pp. 2234–2242.
- [152] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6626–6637.
- [153] M. Binkowski, D. J. Sutherland, M. Arbel, and A. Gretton, “Demystifying MMD GANs,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.
- [154] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari, “How good is my GAN?” in *Proceedings of European Conference on Computer Vision (ECCV)*,

- 2018, pp. 1–20.
- [155] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 105–114.
- [156] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, “ESRGAN: Enhanced super-resolution generative adversarial networks,” in *Proceedings of European Conference on Computer Vision (ECCV)*, 2018, pp. 63–79.
- [157] Y. Li, S. Liu, J. Yang, and M.-H. Yang, “Generative face completion,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5892–5900.
- [158] P. Vitoria, J. Sintes, and C. Ballester, “Semantic image inpainting through improved Wasserstein generative adversarial networks,” in *Proceedings of International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019, pp. 249–260.
- [159] H. Dhamo, K. Tateno, I. Laina, N. Navab, and F. Tombari, “Peeking behind objects: Layered depth prediction from a single image,” *Pattern Recognition Letters*, vol. 125, pp. 333–340, Jul. 2019.
- [160] R. Huang, S. Zhang, T. Li, and R. He, “Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis,” in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2458–2467.
- [161] J. Zhao, L. Xiong, P. Karlekar Jayashree, J. Li, F. Zhao, Z. Wang, P. Sugiri Pranata, P. Shengmei Shen, S. Yan, and J. Feng, “Dual-agent GANs for photorealistic and identity preserving profile face synthesis,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 66–76.
- [162] Y. Tian, X. Peng, L. Zhao, S. Zhang, and D. N. Metaxas, “CR-GAN: Learning complete representations for multi-view generation,” in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 942–948.
- [163] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4401–4410.
- [164] X. Di, V. A. Sindagi, and V. M. Patel, “GP-GAN: Gender preserving GAN for synthesizing faces from landmarks,” in *Proceedings of International*

- Conference on Pattern Recognition (ICPR)*, 2018, pp. 1079–1084.
- [165] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018, pp. 8789–8797.
- [166] P. Isola, J. Y. Zhu, T. H. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1125–1134.
- [167] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. “High-resolution image synthesis and semantic manipulation with conditional GANs,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8798–8807.
- [168] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2242–2251.
- [169] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, “Learning to discover cross-domain relations with generative adversarial networks,” in *Proceedings of International Conference on Machine Learning (ICML)*, vol. 70, Aug. 2017, pp. 1857–1865.
- [170] Z. Yi, H. Zhang, P. Tan, and M. Gong, “DualGAN: Unsupervised dual learning for image-to-image translation,” in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2849–2857.
- [171] C. Vondrick, H. Pirsiavash, and A. Torralba, “Generating videos with scene dynamics,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, Barcelona, Spain, 2016, pp. 613–621.
- [172] S. Tulyakov, M. Y. Liu, X. D. Yang, and J. Kautz, “MoCoGAN: Decomposing motion and content for video generation,” *arXiv preprint arXiv:1707.04993*, 2017. [Online]. Available: <https://arxiv.org/abs/1707.04993>
- [173] V. Mayya, R. Pai, and M. Pai, “Combining temporal interpolation and DCNN for faster recognition of micro-expressions in video sequences,” in *Proceedings of International Conference on Advanced Computing, Communication and Information Sciences (ICACCI)*, 2016, pp. 699–703.
- [174] A. Samavat, E. Khalili, B. Ayati, and M. Ayati, “Deep learning model with adaptive regularization for EEG-based emotion recognition using temporal and frequency features,” *IEEE Access*, vol. 10, pp. 24520–24527, 2022.
- [175] N. McLaughlin, J. M. Del Rincon, and P. Miller, “Data-augmentation for reducing dataset bias in person re-identification,” in *Proceedings of IEEE*

- International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2015, pp. 1–6.
- [176] S. Hauberg, O. Freifeld, A. B. L. Larsen, J. Fisher, and L. Hansen, “Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation,” in *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2016, pp. 342–350.
- [177] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, “Synthetic data augmentation using GAN for improved liver lesion classification,” in *Proceedings of International Symposium on Biomedical Imaging*, 2018, pp. 289–293.
- [178] L. Deng, “The MNIST database of handwritten digit images for machine learning research,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
- [179] R. J. Williams and D. Zipser, “Gradient-based learning algorithms for recurrent networks and their computational complexity,” in *Backpropagation, Theory, Architectures, and Applications*, 1995, pp. 443–446.
- [180] Datasets for Machine Learning Rock Paper Scissors – Rock Paper Scissors Dataset [Online]. Available: <https://laurencemoroney.com/datasets.html>
- [181] S. Visa, B. Ramsay, A. L. Ralescu, and E. Van Der Knaap, “Confusion matrix-based feature selection,” in *Proceedings of Midwest Artificial Intelligence and Cognitive Science Conference (MAICS)*, vol. 710, 2011, pp. 120–127.
- [182] G. C. Linderman, M. Rachh, J. G. Hoskins, S. Steinerberger, and Y. Kluger, “Efficient algorithms for t-distributed stochastic neighborhood embedding,” *arXiv preprint arXiv:1712.09005*, 2017. [Online]. Available: <https://arxiv.org/abs/1712.09005>
- [183] K. Liu, Y. Li, J. Yang, Y. Liu, and Y. Yao, “Generative principal component thermography for enhanced defect detection and analysis,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, pp. 8261–8269, Oct. 2020.
- [184] J. Viola, Y. Chen, and J. Wang, “FaultFace: Deep convolutional generative adversarial network (DCGAN) based ball-bearing failure detection method,” *Information Sciences*, vol. 542, pp. 195–211, Jan. 2021.
- [185] P. L. Suárez, A. D. Sappa, and B. X. Vintimilla, “Infrared image colourization based on a triplet DCGAN architecture,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 212–217.
- [186] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2015, pp. 1–15.
- [187] P. Zhou, J. Feng, C. Ma, C. Xiong, S. C. H. Hoi, “Towards theoretically

- understanding why SGD generalizes better than ADAM in deep learning,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [188] W. Li, C. Chen, M. Zhang, H. Li, and Q. Du, “Data augmentation for hyperspectral image classification with deep CNN,” *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 4, pp. 593–597, Apr. 2019.
- [189] H. Tang, X. Qi, D. Xu, P. H. S. Torr, and N. Sebe, “Edge guided GANs with semantic preserving for semantic image synthesis,” *arXiv preprint arXiv:2003.13898*, 2020. [Online]. Available: <http://arxiv.org/abs/2003.13898>
- [190] Y. Shen, J. Gu, X. Tang, and B. Zhou, “Interpreting the latent space of GANs for semantic face editing,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9243–9252.
- [191] Z. Shen, S. K. Zhou, Y. Chen, B. Georgescu, X. Liu, T. Huang, “One-to-one mapping for unpaired image-to-image translation,” in *Proceedings of IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 1170–1179.
- [192] F. H. K. dos Santos Tanaka and C. Aranha, “Data augmentation using GANs,” *arXiv preprint arXiv:1904.09135*, 2019. [Online]. Available: <http://arxiv.org/abs/1904.09135>
- [193] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.
- [194] Y. Blau and T. Michaeli, “The perception-distortion tradeoff,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6228–6237.
- [195] Z. Zhao, Z. Zhang, T. Chen, S. Singh, and H. Zhang, “Image augmentations for GAN training,” *arXiv preprint arXiv:2006.02595*, 2020. [Online]. Available: <http://arxiv.org/abs/2006.02595>
- [196] C. Han, K. Murao, T. Noguchi, Y. Kawata, F. Uchiyama, L. Rundo, H. Nakayama, and S. Satoh, “Learning more with less: Conditional PGGAN-based data augmentation for brain metastases detection using highly-rough annotation on MR images,” in *Proceedings of Conference on Information and Knowledge Management (CIKM)*, 2019, pp. 119–127.
- [197] W. Nie and A. Patel, “Towards a better understanding and regularization of GAN training dynamics,” in *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019, Art. no. 91.
- [198] J. Elder, “Are edges incomplete?” *International Journal of Computer Vision*, vol. 34, no. 2/3, pp. 97–122, 1999.
- [199] A. Dosovitskiy and T. Brox, “Generating images with perceptual similarity metrics based on deep networks,” in *Proceedings of Conference on Neural*

- Information Processing Systems (NeurIPS)*, 2016, pp. 658–666.
- [200] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang, “Improving the improved training of Wasserstein GANs: A consistency term and its dual effect,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.
- [201] X. Wang, K. Yu, C. Dong, and C. Change Loy, “Recovering realistic texture in image super-resolution by deep spatial feature transform,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 606–615.
- [202] A. Jolicoeur-Martineau, “The relativistic discriminator: a key element missing from standard GAN,” *arXiv preprint arXiv:1807.00734*, 2018. [Online]. Available: <http://arxiv.org/abs/1807.00734>
- [203] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Proceedings of International Conference on Medical Image Computing and Computer Assisted Intervention*, 2015, pp. 234–241.
- [204] N. Ibtehaz and M. S. Rahman, “MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation,” *Neural Networks*, vol. 121, pp. 74–87, Jan. 2020.
- [205] X.-Y. Zhou and G.-Z. Yang, “Normalization in training U-Net for 2D biomedical semantic segmentation,” *IEEE Robotics and Automation Letters*, 2019.
- [206] A. Royer, K. Bousmalis, S. Gouws, F. Bertsch, I. Moressi, F. Cole, and K. Murphy, “XGAN: Unsupervised image-to-image translation for many-to-many mappings,” *Domain Adaptation for Visual Understanding*, pp. 33–49, 2020.
- [207] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, “DRIT++: Diverse image-to-image translation via disentangled representations,” *International Journal of Computer Vision*, pp. 1–16, 2020.
- [208] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3730–3738.
- [209] C. -H. Lee, Z. Liu, L. Wu, and P. Luo, “MaskGAN: Towards diverse and interactive facial image manipulation,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5549–5558.
- [210] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, Jun. 1986.
- [211] P. Afshar, K. N. Plataniotis, and A. Mohammadi, “Capsule networks for brain tumor classification based on MRI images and coarse tumor boundaries,” in



- Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 1368–1372.
- [212] J. Cheng, W. Yang, M. Huang, W. Huang, J. Jiang, Y. Zhou, R. Yang, J. Zhao, Y. Feng, Q. Feng, and W. Chen, “Retrieval of Brain Tumors by Adaptive Spatial Pooling and Fisher Vector Representation,” *PLoS ONE*, vol. 11, no. 6, Jun. 2016, Art. no. e0157112.
- [213] R. Picard, “Toward agents that recognize emotion,” in *Proceedings of IMAGINA, 1998*, pp. 153–165.
- [214] F.J. Moreno-Barea, J. M. Jerez, and L. Franco, “Improving classification accuracy using data augmentation on small data sets,” *Expert Systems With Applications*, vol. 161, Dec. 2020, Art. no. 113696.
- [215] S. Li and W. Deng, “Deep facial expression recognition: A survey,” *IEEE Transactions on Affective Computing*, vol. 2010, pp. 2006–2014, 2020.
- [216] Y. Peng and H. Yin, “ApprGAN: Appearance-based GAN for facial expression synthesis,” *IET Image Processing*, vol. 13, no. 14, pp. 2706–2715, Dec. 2019.
- [217] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of StyleGAN,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8110–8119.
- [218] G.-J. Qi and J. Luo, “Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2168–2187, Apr. 2022.
- [219] R. Webster, J. Rabin, L. Simon, and F. Jurie, “Detecting overfitting of deep generative networks via latent recovery,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11273–11282.
- [220] L. Chen, L. Wu, Z. Hu, and M. Wang, “Quality-aware unpaired image-to-image translation,” *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2664–2674, Oct. 2019.
- [221] Y. Peng and H. Yin, “Facial expression analysis and expression-invariant face recognition by manifold-based synthesis,” *Machine Vision and Applications*, vol. 29, no. 2, pp. 263–284, 2018.
- [222] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6924–6932.
- [223] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *Proceedings of European Conference on*

- Computer Vision (ECCV)*, 2016, pp. 694–711.
- [224] P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, “The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2010, pp. 94–101.
- [225] D. Lundqvist, A. Flykt, and A. Öhman, “The Karolinska directed emotional faces - KDEF,” CD ROM, Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, 1998.
- [226] L.F. Chen and Y.S. Yen, “Taiwanese facial expression image database,” Brain Mapping Laboratory, Institute of Brain Science, National Yang-Ming University, Taipei, Taiwan, 2007.
- [227] M.-Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 700–708.
- [228] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, “Multimodal unsupervised image-to-image translation,” in *Proceedings of European Conference on Computer Vision (ECCV)*, 2018, pp. 172–189.
- [229] H. Tang, H. Liu, D. Xu, P. H. Torr, and N. Sebe, “AttentionGAN: Unpaired image-to-image translation using attention-guided generative adversarial networks,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–16, 2021.

# Appendix A

## Publications

### 1. Journal Papers

- Shih-Kai Hung and John Q. Gan, “Small facial image dataset augmentation using conditional GANs based on incomplete edge feature input,” *PeerJ Computer Science*, volume 7, 2021, Art. no. e760.
- Shih-Kai Hung and John Q. Gan, “Facial expression transfer using a novel GAN-based image data augmentation approach,” *IEEE Access*. (Under Review)

### 2. Conference Papers

- Shih-Kai Hung and John Q. Gan, “Augmentation of small training data using GANs for enhancing the performance of image classification,” in *Proceedings of IEEE International Conference on Pattern Recognition (ICPR)*, 2020, pp. 3350–3356.
- Shih-Kai Hung and John Q. Gan, “Facial image augmentation from sparse line features using small training data,” in *Proceedings of International Work-Conference on Artificial Neural Networks (IWANN)*, 2021, pp. 547–558. (Shih-Kai Hung is the winner of the Early Career Researcher Award in IWANN 2021)
- Shih-Kai Hung and John Q. Gan, “Boosting facial emotion recognition by using GANs to augment small facial expression dataset,” in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN)*, 2022.