Deep Multi-task Learning for Farm Animal Monitoring from Images

By

Hongtao Zhang

A thesis submitted for the degree of

Doctor of Philosophy

School of Computer Science and Electronic Engineering

University of Essex

Jan 2025

Abstract

Large-scale applications of camera equipment has reduced the cost of image and video acquisition, making image and video processing technology be widely used in daily life. The development of deep learning techniques has achieved high accuracy in many image or video tasks, including object detection, object segmentation, key point detection, depth estimation, etc. The success also enables deep learning techniques to be successfully applied in many practical scenarios. However, most deep learning tasks often require a large amount of labeled data for training purpose. But it is difficult to obtain labeled data in many practical application scenarios, which limits effective applications of deep learning techniques.

Multi-task deep learning is a method of artificially combining multiple deep learning tasks in series according to specific task objectives in order to meet the main task requirements. Compared with the end-to-end deep learning strategy, multi-task learning has three main advantages in some scenarios: 1) multi-task learning has a clear learning path, which effectively improves the learning speed. 2) the intermediate tasks or sub-tasks can be supervised during multi-task learning, making the learning process more interpretable. 3) some sub-tasks can be replaced with pre-trained models with good generalization ability without additional training, thereby reducing unnecessary computing power and data requirements. For example, when predicting the object pose, depth estimation is usually used as a sub-task. When performing optical flow estimation, camera pose estimation and depth estimation are also considered as sub-tasks of multi-task learning. In the application of monitoring farm animals, usually only videos can be used as input, and it is difficult to obtain sufficient and effective training data. Therefore, it is more suitable to adopt the multi-task learning method for the application.

In the process of animal breeding, there are many complex scenario tasks that need to be solved. For example, the identification of farm animals is usually carried out by ear tagging. However, this method often has a high cost of use, and it is difficult to ear tag

farm animals. The process is difficult to operate and can easily cause harm to animals and operators. Similarly, in the process of managing farm animals, there are also many risks in the human operation of measuring the length and weight of animals. How to use an effective combination of deep learning sub-tasks to achieve regulatory identification of farm animals under farm conditions is the focus of this thesis.

This thesis makes the novel contributions on how to use multi-task learning methods to achieve the monitoring of farm animals. Firstly, to handle with the common issue of sparse training data in farm animal deep learning tasks, a object detection method is proposed. This method combines the pattern recognition method with the application of the pre-training model for classification tasks to generate the detection results - the bounding box of an animal. It achieves this goal with a small amount of training data in application scenarios, and the object detection results are further improved by adding a monocular depth estimation sub-task. Through experiments, we found that when the training data has less than 200 labels, this method can obtain more accurate object detection results compared to existing supervised learning and semi-supervised learning object detection. Secondly, an animal body length measurement method is proposed which is based on multiple sub-tasks: monocular depth estimation, object segmentation and key point detection. Compared with traditional 2D length measurement methods, the experiment results show a promised improvement. Finally, a multi-task based animal face recognition method is proposed. Again, the proposed method uses the combination of multiple sub-tasks, including monocular depth estimation and 3D convolution feature extraction. Through experiments, we found that this method is better than the 2D method as a whole. Due to the addition of monocular depth estimation results and the calibration of the depth estimation results, this method is more accurate when estimating small-chest cattle. In addition, multi-frame images and corresponding monocular depth estimates are used as input to improve the accuracy of animal face recognition. Through experiments, we found that when a single image is input for face recognition, we achieved the same accuracy as existing face recognition on public datasets. When three images are input for face recognition, the unique feature extraction structure improves the feature extraction efficiency and reduces the loss of high-resolution features caused by the convolution process. Compared with existing face recognition algorithms, the accuracy is improved by more than 20 %.

The proposed methods of object detection and face recognition have been tested and evaluated on public datasets and our own data sets. Compared with other methods, the

results show that this methods have higher accuracy and better scene adaptability in the practical application scenes.

Acknowledgment

I would like to express my sincerest gratitude and respect to my supervisor, Professor Gu Dongbing. Due to the global pandemic caused by the new coronavirus, my doctoral progress was slowed down by two years. During the two years of my gap, my supervisor still insisted on communicating with me and gave me confidence and guidance, so that I could smoothly connect and continue my studies after the epidemic. He gave me invaluable guidance, support and encouragement during my doctoral studies. His words and suggestions make people feel like spring breeze, bringing warmth and hope both in study and life. Although there are many gaps in my basic knowledge, at the beginning of my study, he still patiently guided me step by step into the latest academic frontiers in the field of robotics, and guided me to switch from pure scientific research to more market-oriented application research.

Finally, I would like to thank my family. My parents raised me with endless love since I was a child. I can't go abroad to study for a doctorate without their support and encouragement. It is my parents' hard work that enables me to complete my self-funded study abroad.

Contents

List of Figures

List of Tables

Abbreviations

CNN	Convolutional Neural Networks
VIT	Visual Transformer
ReLU	Rectified Linear Unit
FCN	Fully Convolutional Network
RCNN	Recurrent Convolutional Neural Network
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
ConvLSTM	Convolutional Long Short-Term Memory
2D	Two Dimension
3D	Three Dimension
GPS	Global Positioning System
RGB	Red Green Blue
RGB-D	Red Green Blue Depth
ICP	Iterative Closest Point
CPU	Central Processing Unit
GPU	Graphic Processing Unit
SVM	Support Vector Machine
ATE	Absolute Trajectory Error
RMSE	Root-Mean-Square Error
SSIM	Structural Similarity Index Measure
IOU	Intersection Over Union
SLAM	Simultaneous Localisation and Mapping

SfM	Structure-from-Motion
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
ORB	Oriented Fast and Rotated BRIEF

Chapter 1

Introduction

Animal husbandry is the highest and most significant sources of protein for humans, making it an essential part of agriculture. An effective farm management system is particularly vital. Farm management involves many tedious and repetitive tasks, such as monitoring the health status of farm animals, taking inventory the farm, and identifying livestock. In the past, during an era of pure manual labour, monitoring the health status of farm animals required multiple individuals to cooperate in using instruments such as rulers to measure animals' physiological data. This process was not only cumbersome but also involved many hazards, as the operator needed to restrain the animal. In addition, when it came to identifying farm animals, ear tags were often considered a relatively accurate auxiliary tool. However, the hardware costs can be somewhat prohibitive for farmers compared to simply taking photos of the livestock, and incorrect application of ear tags could lead to infection or tissue damage. The emergence of deep learning computer vision has made it possible for machines to replace manual labour in farm operations. This emerging technology not only greatly reduces operational costs, but also mitigates the ethical risks of harming animals.

Since farm animal monitoring algorithm applications are often carried out in environments that are dirty and cluttered, with unstable lighting conditions and complex tasks, a single computer vision algorithm is usually insufficient to carry out relevant tasks effectively. Therefore, the combination of multiple computer vision algorithms is often necessary. This thesis selects farm animal management as an application scenario and, based on various deep learning-based computer vision tasks, makes many innovative attempts and contributions to solve practical problems encountered in real-life practice. This chapter introduces

the research motivation, research challenges, methods used, and novel contributions. It also includes an outline of the thesis.

1.1 Motivation

The applications of deep learning in farm animal management can effectively reduce the costs and safety issues associated with human and animal involvement in purely manual farm animal monitoring processes. Deep learning applications mainly consist of two structures: end-to-end learning and multi-task learning. Compared to multi-task learning, the end-to-end deep learning model is often simpler and easier to build. An effective network structure and loss function for training and application are usually sufficient for most tasks. However, when facing complex problems in the application process, end-to-end learning models often struggle to converge. A single end-to-end structure is insufficient to effectively extract features from data in different forms simultaneously. When performing transfer learning based on end-to-end deep learning, any changes in the output format require retraining some or all of the parameters in the pre-trained weights, which inevitably results in a waste of computing power.

In recent years, deep learning researchers have gradually developed specialized algorithms and network structures for different data types and characteristics. These include monocular depth estimation, object detection, object segmentation, key point detection for image feature extraction, LSTM [1] and GRU [2] modules for extracting features from temporal feature data including numbers and text, and MLP [3] and KAN [4] networks for polynomial feature extraction. This allows data of different forms and characteristics to be effectively extracted in multi-task learning-related tasks. The emergence of the Transformer structure [5] enabled basic deep learning tasks to achieve strong performance in different scenarios simultaneously. It provides effective support for the transfer learning of subtasks in multi-task learning. At the same time, the emergence of multimodal technology has accelerated the progress of research related to feature fusion and feature alignment of different data forms. It also provides favorable conditions for multi-task learning.

Multi-task learning is usually designed by manually setting up several sub-tasks according to the requirements of the final task. The sub-tasks are usually connected in serial or parallel manners according to the task requirements. After the output of all sub-tasks is

completed, a merging network is used to combine the features extracted from the sub-tasks and complete the output of the final result. Compared with end-to-end structure, multi-task learning usually has the following advantages:

1. Fast training speed. Multi-task learning can usually set sub-tasks based on experience. Therefore, once the feature extraction of the sub-task is completed, only a shallow feature extraction network is often needed to output the final task. Since the design of the sub-task contains prior knowledge, multi-task learning is often easier to converge and the training speed of the feature merging network is generally faster.

2. Efficient utilization of various pre-trained model weights. Normally, the change in application scenarios and output content of an end-to-end task often requires retraining some or all of the network weights to perform transfer learning, which undoubtedly wastes valuable computing power. On the other hand, when performing multi-task learning, the rich variety of sub-tasks makes transfer learning possible. Meanwhile, the emergence of transformers has made many deep learning tasks adapt to various scenarios, which allows easier transfer learning of sub-tasks in multi-task learning, and also enables more effective applications of pre-trained model wights across different models and scenarios.

3. More flexible solution in model deployment. When handling some complex tasks, the end-to-end model often requires a relatively large number of parameters, imposing high requirements for the deployment device. Conversely, when deploying multi-task learning models, since the inference task is completed by multiple models, multi-task learning is often composed of multiple sub-tasks with relatively fewer parameters, allowing the model to be distributed across multiple devices with lower computing power, reducing the model's demand for devices during the inference process.

This thesis aims to use multi-task learning to solve practical problems encountered in farm animal monitoring whilst taking into consideration actual scenario requirements. This research will also argue that a multi-task learning structure is the most suitable neural network design for farm animal monitoring scenarios through comparative evaluation experiments.

1.2 Challenges and Objectives

This thesis aims to apply multi-task learning to solve three tasks in farm animal management: object detection, parameter measurement, and farm animal identification. Although these tasks are different, their challenges are almost identical, such as improving estimation accuracy and robustness to small amounts of data. This thesis will set out to accomplish the following objectives and address the challenges they present:

1. **Object Detection**

   Object detection is one of the most basic tasks in computer vision. Usually, the input of object detection is an image, and a deep learning network outputs the result after completing feature extraction. The output result consists of three parts: (1). The four corner coordinates of the rectangular box that encloses the object. (2). The class ID of the detected object. (3). A confidence number, representing the confidence of the object detection result. Object detection in general scenarios is mostly obtained through supervised learning training based on labeled data. Usually, this method relies on a large amount of labeled data, and once the environment changes drastically, it will affect the detection accuracy.

   Compared with object detection in general scenarios, object detection for farm animal monitoring faces the following main challenges:

   A. Little training data. It is difficult to find enough object detection datasets containing farm animals on the Internet, which makes it difficult to collect the corresponding object detection data.

   B. Significant variation in the morphology of the detected targets. In the actual application of object detection for farm animal monitoring-related operations, since most of the detection objects are living organisms, unlike the detection of rigid bodies, their morphologies are easily changed during the detection process. Therefore, it is difficult to directly train an object detection model that meets the application conditions using training data from a single morphology.

   C. Complex environment and unstable lighting conditions. The application environment

of farm monitoring includes indoor or semi-outdoor scenes with dim light or a single light source, and partial overexposure happens frequently. In most cases, compared with general indoor and outdoor object detection, the background in the application process of farm animal monitoring is more complex, and requires higher robustness.

2. **Parameter Measurement**

   In the process of farm animal monitoring, it is important to obtain the physiological data of farm animals promptly. Parameter measurements, are designed to measure physiological parameters such as weight, length, etc. of farm animals in an image. Usually, the input of such tasks is an image, and the output is the corresponding measurement parameters.

   Farm animal parameter measurement is not a common task in computer vision, making it difficult to directly refer to existing computer vision tasks. Due to the characteristics of the task itself and the particularity of the application scenario, there are the following challenges in designing farm animal parameter measurement:

   A. It is difficult to obtain a valid scale. Objects in an image are larger when they are near and vice versa, so it is difficult to measure the actual size of an object solely based on its size in the image.

   B. It is difficult to measure three-dimensional objects in space using images. Since it is difficult to ensure the viewing plane and the measurement object have a stable position and distance relationship when measuring farm animal parameters, once the relative position between the viewing plane and the measurement object changes, the operator will have a hard time directly estimating the proportional relationship between the object in the image and the actual target.

   C. Complex environment. The complex environmental background and varied lighting conditions of the farm demand algorithms with higher robustness.

   D. The quality of data acquisition equipment varies greatly. In actual operation, it is difficult to ensure that the data acquisition equipment has stable performance and resolution. Therefore, more detailed data normalization operations are required before model training.

3. **Farm Animal Identification**

Correctly identifying farm animals is one of the most important tasks in achieving farm animal monitoring. Farm animal recognition involves processing the images of farm animal through a deep learning network to extract their features. The extracted features are compared to those in a database to determine their identity information.

Although farm animal recognition shares certain similarities with facial recognition algorithms in terms of network design, the characteristics of the recognition targets and the specificity of the application scenarios make it challenging in actual operations, namely:

A. Compared with human faces, animal faces are more three-dimensional and difficult to capture within a single image (see figure 1.1). When designing a facial recognition algorithm, it is often necessary to use an object detection or a segmentation network to locate the face and remove the background in the image for subsequent feature extraction and recognition tasks. Usually, facial object detection or recognition algorithms use the eyes, nose, and mouth as the main features for localization and recognition, and these parts of the face are located on the same plane. However, the faces of farm animals, including pigs, cattle, and sheep, are more three-dimensional, making it difficult to use a single image to simultaneously capture effective features including horns, eyes, mouths, noses, etc., so the direct application of facial recognition algorithms is impossible. Furthermore, the three-dimensional characteristics of farm animal faces also make it difficult to define multiple images and merge features effectively, which mainly manifests challenges in locating a valid plane and defining shooting criteria. Additionally, researchers will also need to consider the options, whether to perform image merging before feature extraction, complete feature merging after feature extraction, or complete feature extraction and comparison from multiple images at different angles individually before merging the comparison results.

B. Compared with human faces, some farm animals lack effective facial features, making it difficult to identify them directly using plane information. Compared with facial recognition, the single color and obstruction of facial hair with animals like yaks or cattle make it even more difficult to extract effective features.

C. Complex environment and unstable lighting conditions. Like object detection and farm animal parameter measurement, farm animal facial recognition also faces issues such as

Figure 1.1: Animal faces are more three-dimensional and have low color contrast

uneven lighting, overexposure and dirt, which increase the difficulty of the recognition task.

Hence, this thesis will use robust deep multi-task learning structures to solve the above challenges for object detection, parameter measurement, and farm animal identification.

1.3 Methodology

The large-scale application of deep learning in the field of computer vision has led to the emergence of many visual tasks. The emergence of deep learning-based visual tasks such as monocular depth estimation, camera pose estimation, optical flow estimation, object detection, object segmentation, key point detection, etc. has made it possible to effectively obtain various visual cues from images. These visual cues have become the subtasks in multitask learning. In the process of researching farm animal management applications based on multi-task learning, the experimental method is mainly divided into four steps:

1. **Data collection and data augmentation** After determining the purpose of the experiment, the first step is to collect the data required for the experiment according to the requirements. Although the aim is to reduce the data requirements by using multi-task learning method, a small amount of data that fits the nature of the scene is essential. This research aims to study the algorithms required in the process of farm animal management. It is often difficult to find valid public data on the Internet. Therefore, in addition to retrieving public data, our own data collection is also required. Since the algorithm application scenarios have the characteristics of complex lighting conditions and chaotic environments, after completing the data collection, it is also necessary to perform targeted data augmentation.

2. **Multi-task structure design** Since the algorithm is based on multi-task learning, after completing data collection, it is necessary to select sub-tasks based on the characteristics of the scene and the task. While meeting the accuracy of the results, the zero-shot algorithm with a small number of parameters should be used as much as possible to facilitate the deployment and application of subsequent algorithms. Typically, object detection, object segmentation, and key point detection are used in the multi-task design process to remove the background and improve the network's attention on key areas, while depth estimation is often used to increase 3D information in multi-task learning. After determining the sub-tasks required for the multi-task learning architecture, it is also necessary to design the multi-task learning architecture and the sub-task operation process according to the requirements. Finally, multi-task learning requires a merging network to merge the outputs of the sub-tasks and further perform feature extraction to complete the output of the final result.

3. **Loss function Design** Most of the sub-tasks used in the multi-task learning process directly use pre-trained models or directly perform transfer learning based on the sub-task model and loss function, so there is no need to design a loss function for the sub-task. However, after completing the multi-task learning structure design, it is usually necessary to comprehensively consider the characteristics of the data, neural network and task to design a loss function. For example, when performing some multi-task learning to predict values, we usually use L1 loss or L2 loss. When the results are output in the form of classification, cross-entropy is more commonly used. When the number of data categories is unbalanced, it can be alleviated by weighting each category in cross-entropy or using focal loss. When performing some image comparison tasks such as image similarity comparison, SSIM is a commonly used loss function. In addition, MAE and MSE are also used when performing image comparison.

4. **Design test methods and conduct test experiments** After completing the training of all tasks in the multi-task process, the results need to be evaluated. In addition to evaluating the accuracy of sub-tasks and final tasks, including recall rate and accuracy, considering the deployment cost, it is also necessary to evaluate the flops and parameters of the merged network, and finally complete a comprehensive evaluation of the

algorithm results.

1.4 Contribution

The main contributions of this thesis consist of the following three parts. They are:

A. This research proposed a new method that utilizes pre-trained classification network models to address practical scenarios where there is insufficient annotated data for object detection in images. Our method generates multiple scale gridded heatmaps from a pre-trained classification network, which can effectively integrate various pre-trained features of different categories without training. Similarity segmentation results were also used to further refine the object detection performance. The experiment results and datasets show this method can produce better detection results than existing object detection networks [6] [7] [8] when the amount of training data is less than 200.This part is based on the following conference publication:

- Hongtao, Zhang, Dongbing Gu and Kang Liang. "A Breeding Animal Detection Method from Images With Grided Heatmaps." 2024 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, 2024.

B. We proposed a new deep-learning framework to estimate the chest circumference of domestic animals from images. This parameter is a key metric for breeding and monitoring the quality of animals in animal husbandry. We design a set of feature extraction methods based on a multi-task learning framework to address the challenging issues in the main estimation task. The multiple tasks in our proposed framework include object segmentation, keypoint estimation, and depth estimation of cows from monocular images. The domain-specific features extracted from these tasks improve upon our main estimation task. In addition, we also attempt to reduce unnecessary computations during the framework design to reduce the cost of subsequent practical implementation of the developed system. Our proposed framework was tested on our own collected dataset to evaluate its performance. This part is based on the following journal publication:

- Hongtao, Zhang, and Dongbing Gu. "Deep Multi-task Learning for Animal Chest Circumference Estimation from Monocular Images." Cognitive Computation 14.2 (2024).

C. The advent of 3D convolution over spatial and temporal domains has substantially improved the performance of many visual tasks. In this work, we used 3D convolution to extract spatial and temporal features from consecutive images for face recognition tasks, particularly for farm animal face recognition tasks in breeding management applications. We also applied a multi-task learning approach to the face recognition tasks by adding depth and object segmentation results. Through the application of multi-task learning and large visual models, we reduced the number of training steps and the reliance on complex labeled data. We contributed a multi-scale 3D convolution network to extract spatio-temporal features from the sub-tasks, depth and mask, for face recognition tasks. In addition, we also developed a method for generating a 3D point cloud for the detected face based on monocular depth estimation and key point detection, making it possible to register a limited number of images from different angles. With the dataset collected by ourselves,and also public datasets on human faces LFW [9] and CelebA [10], we tested our network and the experiment result shows that our proposed network outperforms some of the state-of-the-art networks. The test results on the LFW [9] dataset are slightly better than those of the existing methods [11] [12] [13] [14] [15], while the test results on the CelebA [10] dataset significantly outperformed existing methods. The comparison results indicate that our test results on CelebA are generally 50 % more accurate than those of existing methods.

1.5 Outline of Thesis

This first chapter serves as an introduction to the full thesis. The second chapter provides the background literature review of my research. All of my work's contributions are distributed in Chapter 3, Chapter 4, and Chapter 5. The last one, Chapter 6 presents the conclusions and potential future work of this thesis. The details of each chapter are summarized as follows:

- Chapter 2 This chapter reviews relevant works about the research topic and serves as a theoretical framework. It introduces some related algorithms of multi-task learning and presents classic cases from four aspects: scene characteristics, structural design, sub-task selection and merged network. This chapter also reviews the monocular depth estimation commonly used in visual multi-tasks to increase stereo information. According to the chronological order, the development of monocular depth estimation is divided into traditional monocular depth estimation, joint estimation monocular depth estimation and

transformer-based monocular depth estimation. In the following three sections, three commonly used multi-task learning sub-tasks for removing background and highlighting key points are introduced, namely object detection, object segmentation and key point detection.

- Chapter 3 A method for object detection based on a pre-training classification network is proposed. This chapter introduces the research motivation, method characteristics, the weakly supervised object detection and pattern recognition work, and the problems of existing object detection algorithms. An object detection method based on a pre-training classification network without the need for additional data to participate in training is proposed. This method uses the pre-training model to generate preliminary object detection results, and uses pattern recognition-related algorithms for post-processing to improve the detection accuracy of object detection. To evaluate the performance, we compared it with some existing object detection algorithms on VOC [16], COCO [17] and our own collected datasets. This chapter concludes that with a small amount of training data, the system generates more accurate object detection results compared with other published object detection learning networks.

- Chapter 4 A method based on multi-task learning to measure the chest circumference of dead cattle using a single image is proposed. This chapter introduces the background and application scenarios of the research, and the use of subtasks in the multi-task learning framework, including monocular depth estimation, object detection, object segmentation and key point detection. Then a method for measuring the chest circumference of dead cattle based on multi-task learning is proposed. The input of this method is an RGB image. The stereo information is added to the process by adding monocular depth estimation, and the scale information in the image space is calculated using the landmarks in the image. Finally, the chest circumference of the dead cattle is measured using the multi-task learning framework. To evaluate the performance, we measured the running speed of the network by flops and parameter, and calculated the mean absolute error of the current method and the pure 2D method. This chapter concludes that the robustness and accuracy of the chest circumference measurement method can be improved by adding monocular depth estimation and applying multi-task learning.

- Chapter 5 A method for animal face recognition based on multi-task learning and 3D convolution [18] is proposed. First, a brief research background of this study is introduced, and then the research progress of face recognition-related algorithms and related sub-tasks applied in this study is introduced. Then, a method for animal face recognition based on multi-task learning and 3D convolution [18] is proposed. The input of this method is a single RGB image. The depth is added as a new channel by using monocular depth estimation to generate an RGBD four-channel image. The 3D convolution [18] is used to shape the input image into the 3D convolution network [18] either in the form of a channel sequence or an image sequence according to the characteristics of animal faces and human faces, and spatial attention is used to improve the network performance and accelerate the convergence speed of the training process. Finally, feature extraction and tensor alignment are completed to achieve animal face and human face recognition. The face recognition is tested and evaluated on the LFW [9] and CelebA [10] datasets respectively, and a dataset containing 500 cow faces is collected to complete the test of the animal face recognition. This chapter concludes that through the application of monocular depth estimation and 3D convolution [18] to extract features sequentially, good recognition results can be obtained in human and animal face recognition.

- Chapter 6 gives a conclusion for the entire thesis and summarizes the contributions. Our novel ideas provide some new directions for future work in the direction of the multi-task research area.

This first chapter introduces my thesis, consisting of motivation, research tasks and challenges, methodology and contributions, and an outline of the whole thesis. The next chapter will present the thesis literature review.

Chapter 2

Literature Review

Before this thesis presents its main contribution work, it is necessary to lay out some background reviews. Section 2.1 introduces a few existing studies on multi-task learning. By listing the applications of multi-task learning in different scenarios, this chapter aims to enhance the understandings of the characteristics of multi-task applicable scenarios and learn the skills of sub-task selection in multi-task learning. Section 2.1.4 introduces the backbone networks of deep learning. By comparing the structures and performance of various backbone networks, it is convenient to design corresponding deep learning networks according to task requirements when performing multi-task learning. Sections 2.2, 2.3, and 2.4 introduce some common sub-tasks in multitask learning respectively. Section 2.2 introduces the main methods of monocular depth estimation. Depth, as an important component of 3D information, can provide rich stereoscopic information for many scenes and be effectively applied in many scenarios. Sections 2.3 and 2.4 introduces the mainstream methods for object detection, object segmentation, and keypoint detection. Typically, these three tasks are considered as subtasks of multi-task learning, using object detection, object segmentation, or keypoint detection to highlight the effective parts in the image, thereby improving the processing efficiency of subsequent steps. Section 2.5 provides a brief summarization of this chapter.

2.1 Multi-task Learning

This section introduces the literature related to multitask learning from three perspectives: scenario characteristics, task structure, and subtask selection.

Firstly, focusing on the characteristics of multi-task learning scenarios, Section 2.1.1 summarizes and analyzes the features of some scenarios suitable for multi-task learning. Some multi-task learning tasks seek to improve the accuracy of existing machine learning tasks by incorporating sub-tasks into the multi-task architecture, while others aim to achieve non-traditional machine learning objectives, which typically require a regression network to connect sub-tasks with the final output.

Secondly, in terms of the structural design of multi-task learning, Section 2.1.2 analyzes various multi-task learning applications to understand the structural characteristics of multi-task learning sub-tasks. Some multi-task algorithms run sub-tasks in parallel to accelerate the overall operation of multitasking learning and improve algorithm efficiency. Additionally, there are certain multi-task learning processes where sub-tasks rely on the outputs of previous sub-tasks for computation, which requires a serial execution approach.

Thirdly, section 2.1.3 provides examples of multi-task learning tasks to illustrate how to select sub-tasks in multi-task learning. Compared to end-to-end learning, one of the major features of multi-task learning is that it can replace sub-tasks with mature pre-trained models. Therefore, this section lists some methods that directly use pre-trained models as sub-tasks for multi-task learning. Different sub-tasks have different roles in multi-task learning. Generally, sub-tasks serve two functions:

1. Filtering information. In many multi-task learning tasks, sub-tasks such as keypoint detection, object detection, and object segmentation are used to filter the effective information from the image, which facilitates the processing of subsequent tasks.

2. Adding information. Some multi-task learning tasks will add depth estimation or optical flow estimation networks to provide additional spatial or temporal information.

Finally, in section 2.1.4, this thesis introduces some network structures for multi-task feature merging.

2.1.1 Scenario Characteristics

Multi-task learning, when applied in the field of modern machine learning, can improve the accuracy of existing machine learning tasks by incorporating sub-tasks. The core idea of multi-task learning is that by learning multiple related tasks simultaneously, the sharing of knowledge between tasks becomes possible, which in turn improves the performance of

the main task. Many studies have shown that multi-task learning can effectively utilize the correlation between tasks and improve the generalization ability and accuracy of the model.

Multi-task learning has been widely applied to object detection and semantic segmentation tasks in the field of computer vision. Ren et al. proposed a joint learning framework that combines object detection and semantic segmentation tasks by sharing feature representations, thereby improving the performance of both tasks [19]. The logic of this method is that the features obtained during shallow feature extraction in object detection tasks and semantic segmentation tasks are highly similar. Therefore, sharing features allows the model to better understand the different features in the image, thus improving the performance of both tasks at the same time.

Multi-task learning also has mature application cases in the field of natural language processing (NLP). Collobert and Weston proposed a multi-task learning framework based on convolutional neural networks, which can simultaneously perform multiple tasks, such as part-of-speech tagging, named entity recognition, semantic role tagging, etc. by sharing the underlying feature extraction network [20]. This method captures semantic relationships between different NLP tasks through weight sharing in the underlying feature extraction network, improving each subtask's accuracy. In the field of machine translation, Luong et al. proposed a multi-task learning framework similar to this idea [21], which combines machine translation with language model tasks and improves the quality of machine translation by sharing encoders and decoders.

In addition to the above two fields, multi-task learning has also achieved remarkable results in speech recognition, recommendation systems and other fields. Huang et al. introduced multi-task learning in speech recognition and used deep learning networks to learn speech recognition and speech enhancement tasks simultaneously, which greatly improved the effectiveness of speech recognition [22]. Zhang et al. proposed a recommendation system framework based on multi-task learning, using deep learning networks to simultaneously learn the two tasks of user click rate estimation and purchase rate estimation, improving the overall performance of the recommendation system [23].

In summary, by adding subtasks, multi-task learning can effectively utilize the correlation and shared feature representations between tasks to improve the accuracy of existing machine learning tasks. This method has shown remarkable results in many fields such as

computer vision, NLP [5], speech recognition, and recommendation systems, and has become an important means to improve model performance.

Another application scenario of multi-task learning is to implement some non-traditional machine learning tasks, which often require a regression network to connect sub-tasks and the final output. In these applications, multi-task learning is not just about improving the accuracy of existing tasks, but about achieving complex, multi-step tasks often not addressed by traditional end-to-end learning methods.

Multi-task learning has also been applied to the field of medical image analysis, allowing for early diagnosis of cancer and prediction of treatment effects. Liu et al. proposed a multi-task learning framework that can simultaneously learn tasks such as tumour segmentation, tumour classification, and treatment effect prediction. The scholars used the characteristics of multi-task learning to achieve a comprehensive diagnosis of cancer patients and the formulation of personalized treatment plans [24]. This method used a regression network to connect the results of tumour segmentation and the prediction of treatment effects and improved the precision and reliability of the prediction by comprehensively considering multiple aspects of diagnostic information.

In the field of autonomous driving, multi-task learning is often employed to achieve information merging between various sensors on the vehicle and comprehensive decision-making. Chen et al. used multi-task learning to combine perception tasks (object detection, lane detection, etc.) with decision-making tasks (path planning, motion control, etc.), and employed regression networks to connect the output results of various sensors and merge decisions [25]. This method improved the safety and reliability of autonomous driving systems by making comprehensive and effective use of different sensors.

In the financial sector, multi-task learning is particularly suited for stock price prediction and risk management. He et al. used multi-task learning to simultaneously learn stock price prediction and risk prediction tasks, making stock market-related analysis more comprehensive [26]. By using a regression network, the research achieved feature alignment and information merging from different sources, making market prediction information more accurate and reliable.

Multi-task learning also has relevant application scenarios in robot control. In this scenario, multi-task learning is usually used to implement complex robot task planning and

execution. Levine et al. used multi-task learning to build a robot control framework. They leveraged the characteristics of multi-task learning to simultaneously learn robot grasping, movement, and obstacle avoidance tasks and enabling autonomous operation of the robot in a complex environment [27]. The regression network plays a crucial role in connecting different tasks in this approach.

In summary, multi-task learning is applied in many scenarios. Typically, multi-task learning can be used to optimize traditional machine learning tasks, and it can also be used to complete new machine learning tasks by combining subtasks in a multi-task learning framework.

2.1.2 Structural Design

Structural design is crucial for multi-task learning. When designing the structure of multi-task learning, there is often a cascade relationship between sub-tasks, where some tasks require the output of previous stages as input, necessitating a serial structure. In certain cases, sub-tasks can be run in parallel to minimize training and application time. In addition, when designing the structure of multi-task learning, it is also important to consider the rational design of the structure based on the operational requirements of sub-tasks and the efficient utilization of hardware computing power.

In cases where hierarchical tasks depend on the output of the previous task stage as input, the execution order of tasks strictly follows the dependency chain, and the preceding tasks provide necessary information for subsequent tasks. This serial design has broad applications in fields such as natural language processing(NLP) and computer vision. For example, in NLP tasks, the output of part-of-speech tagging can be used as input for Named Entity Recognition (NER) or Dependency Syntax Analysis, thereby improving the accuracy of subsequent tasks.

A classic example is the multi-task deep learning model proposed by Collobert et al. (2011) [28], which was applied to various sub-tasks in natural language processing, such as part-of-speech tagging, named entity recognition, syntax parsing, etc. In this model, the underlying shared neural network was used to extract text features, which are first applied to simpler tasks (such as part-of-speech tagging), and then used for more complex tasks (such as named entity recognition). This gradual construction approach, with layers of depen-

dencies, enables each task to fully utilize the semantic information generated by preceding tasks, thereby improving overall task performance. In addition, this cascaded task design effectively reduces the number of parameters in the model, as tasks share a significant amount of underlying feature representations.

The Taskonomy model [29] proposed by Zamir et al. in 2018 is another typical example of a multi-task learning application in the field of computer vision. This method enables cross-task information sharing based on the dependencies between different visual tasks. For example, the output of low-level tasks such as edge detection or normal estimation could provide the necessary basic information for high-level tasks such as 3D reconstruction or object recognition. By modelling these dependencies, Taskonomy could effectively utilize the correlation between cross-tasks and improve the accuracy of the overall task. This serial task relationship could not only enhances performance of the model, but also reduces feature redundancy and saves computing resources.

However, the serial task structure also has its limitations, primarily in terms of computational efficiency, as the execution of subsequent tasks must wait for the completion of preceding tasks, which increases the overall running time. In addition, the serial relationship introduces a risk of error propagation between tasks. Once there is a significant error in the output of the preceding task, the accuracy of subsequent tasks will also be significantly affected. Therefore, when designing such multi-task learning structures, how to balance the dependencies and information transmission between tasks has become a critical problem that requires further research.

In addition to the serial task structure, another common design in multi-task learning is the parallel task structure, where multiple tasks are processed simultaneously, without relying on the output of other tasks. The main advantage of this design is that it can significantly improve the running efficiency of the model, especially when sufficient computing resources are available and multi-task parallel execution is supported. The parallel task structure can greatly reduce overall running time.

In the field of image processing, the multi-task learning framework proposed by Kendall et al. in 2018 is based on a parallel structure. In this method [30], tasks such as image semantic segmentation, depth estimation, and scene recognition shared the same feature extraction network, while the specific parts of each task were processed in parallel through independent

output layers. This design not only ensured that each task could obtain enough shared information, but also reduced the inference time and improved the overall performance of the system by processing multiple tasks in parallel. In this way, the model still maintains a high accuracy when processing multiple tasks at the same time.

The parallel task structure is particularly suitable for scenarios where tasks are relatively independent but can share input data or features. A successful example of this is the MT-DNN(Multi-Task Deep Neural Networks) proposed by Liu et al. (2019) [31], which is a framework designed for natural language processing. In MT-DNN, multiple natural language processing tasks (such as text entailment, sentiment classification, and question-answering systems) could share the underlying language representation model and process different tasks in parallel through independent task headers. Due to the lack of dependencies between tasks, this model could fully utilize the parallel computing power of hardware, significantly reducing training and inference time while improving model accuracy.

Although parallel structure can effectively reduce running time, its limitation lies in the potential associations or dependencies between certain tasks. If these associations are ignored and forced to be executed in parallel, it may result in insufficient information sharing between tasks, thereby affecting overall performance. Therefore, when designing a parallel task structure, how to reasonably select which tasks are suitable for parallel processing and share partial information when necessary is a problem that needs to be balanced.

In practical applications, the design of multi-task learning models must not only consider the dependencies or parallelism between tasks, but also fully consider the computational efficiency of hardware. As deep learning models become increasingly complex, how to efficiently run multitasking learning models with limited hardware resources has become a critical research focus. Therefore, many multi-task learning models incorporate considerations of hardware characteristics in their design, optimizing the model structure to improve the efficiency of hardware resource utilization.

An important optimization direction is to leverage the parallel computing capabilities of hardware accelerators such as GPUs and TPUs. The multi-task learning version of Google's BERT model (MT-BERT) [32] takes full advantage of the parallel capability of TPU in its design, enabling multiple natural language processing tasks to be efficiently executed in parallel. MT-BERT greatly improves inference efficiency by sharing the encoder layer and

combining the benefits of hardware accelerators, meeting the demand for multi-task parallel processing in practical applications.

In addition, for application scenarios such as embedded devices and low-power consumption, hardware-aware multi-task learning models have gradually become a popular research direction. The hardware-aware neural architecture search (Hardware-Aware NAS) method proposed by He et al. (2020) [33] significantly improved the efficiency of the model on embedded devices by incorporating hardware constraints into the model structure design. This method achieved efficient operation in resource-constrained environments by reducing redundant calculations and optimizing memory usage.

With the continuous development of hardware technology, future multi-task learning model designs will increasingly rely on new technologies such as heterogeneous computing architectures and quantum computing. By fully utilizing the characteristics of this new hardware, future multitasking learning models will be able to achieve efficient task processing in a wider range of application scenarios, driving the practical application of deep learning models.

2.1.3 Sub-tasks Selection

Reasonable selection of sub-tasks is essential for completing multi-task learning. When selecting sub-tasks, the sub-task algorithm's accuracy must be the primary consideration. In addition, multiple sub-tasks may meet a requirement at the same time, and sub-task selection needs to be based on the specific details of the requirement. For example, in general, using object segmentation [34] [35] to highlight key areas in an image often has the best effect, and keypoint detection [36] [37] [38] and object detection can also play a similar role. Compared to object segmentation [39] [40], object detection [41] [42] [43] often has a faster running speed, and keypoint detection [44] [45] for small objects in scenes is more advantageous than object segmentation [46] [47]. Therefore, a reasonable selection of sub-tasks has a crucial impact on the effectiveness, runtime, and generalization ability of multi-task learning.

2.1.4 Merging Network

When designing a multi-task learning structure, it is often necessary to use a regression network to merge the results of each sub-task and output the final result. Therefore, an efficient and fast merging network is crucial for multi-task learning. In the following sections, this

part will introduce some key convolutional neural network structures.

In the first section, the backbone network structure is introduced using RepVGG [48] as an example to illustrate the general architecture of convolutional neural network (CNN) backbones [49] [50]. In second section, attention modules [51] [52] represented by CBAM [53] are introduced, which are often added as supplementary modules [54] to the backbone network [55] to improve network performance. In the third and fourth sections, two novel network architectures based on convolutional neural networks are presented, where the third section represents a network structure that combines transformer architecture [1] [56] [2] [57] to enable traditional convolutional neural networks [58] [59] to achieve better results on large amounts of data [60] [61]. The fourth section describes a GNN-based [62] architecture excels at extracting features from images with structural features.

RepVgg

RepVgg [48], introduced in 2021, presents two key innovations that distinguish it from traditional network structures:

1. Building upon the residual structure in ResNet [63], RepVgg addresses the issues of vanishing and exploding gradients, the combination of residual structure and trunk convolution layer parameters was achieved, which improved the utilization efficiency of network parameters and significantly reduced the total number of parameters.

2. In the process of backbone network design, 3 * 3 convolutional cores should be used as much as possible. Because of the acceleration libraries such as NVIDIA's cudNN, Intel's MKL and related hardware have very good performance optimization for 3x3 convolutional cores. Therefore, this modification further improves the reasoning speed of the network.

CBAM

The Convolutional Block Attention Module (CBAM) is a lightweight Attention Module proposed in 2018. It is designed to perform attention operations on spatial dimensions and channel dimensions. In their paper, CBAM modules were added to ResNet [63] and MobileNet [64] [65] for comparison, and experiments were carried out to apply the two attention modules successively. At the same time, CBAM visualization was carried out, which showed that attention pays more attention to the target object. CBAM consists two submodules: CAM (Channel Attention Module) [66] and SAM (Spatial Attention Module),

responsible for channel-wise and spatial-wise attention respectively. This not only saves parameters and computing power, but also ensures that it can be integrated into the existing network architecture as a plug-and-play module.

Placing the channel attention in CBAM before the roll-up layer allows for iterative adjustment of the input feature channel. For some tasks that require feature fusion, such as the joint input of RGB and depth, and the joint input of RGB and optical flow, the results will be better. The structure is illustrated in Fig 2.1.

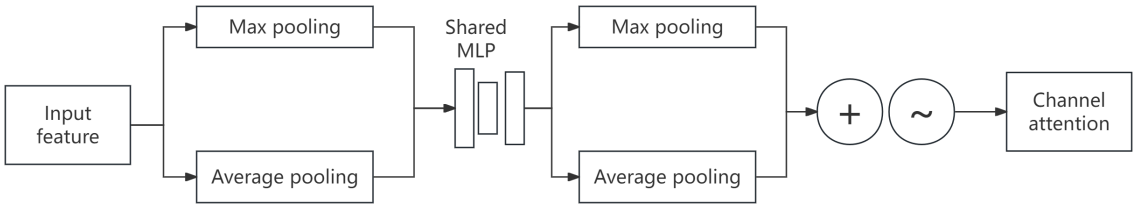

Figure 2.1: Overview of CBAM

Vision transformer(ViT)

ViT [67], published in 2019, applied the Transformer architecture to image classification. Although it was not the first paper to apply Transformer to visual tasks, it has become a milestone work in the computer vision field due to its simple model, good performance, and strong scalability (the larger the model, the better the effect), and has also inspired subsequent related research.

The core conclusion in the original ViT paper was that when there is sufficient data for pre-training, the performance of ViT will exceed that of traditional Convolutional Neural Networks (CNNs) [68] [18] [69], which addresses the limitations of the lack of inductive bias in the Transformers, and achieving better migration effects in downstream tasks.

However, when the training data set is not large enough, the performance of ViT generally lags behind that of ResNets of the same size, because Transformer lacks inductive—prior knowledge or assumptions built into the model. CNN has two kinds of inductive bias, one is locality/two-dimensional neighbourhood structure, that is, adjacent regions on the image have similar characteristics; One is translation equivariant. When CNN has the above two inductive biases, it has a lot of prior information and needs relatively little data to learn a better model.

ViT divided the input image into multiple patches (16x16), and projected each patch as a fixed-length vector into the Transformer. The subsequent encoder operation was identical to the original Transformer. However, due to the classification of images, a special token was added to the input sequence, and the corresponding output of this token was the final category prediction.

A ViT block can be divided into the following steps:

(1)Patch embedding: For example, if the input image size is 224x224 and the image is divided into fixed-size patches with a patch size of 16x16, each image will generate 224x224/16x16=196 patches, which means that the input sequence length is 196, each patch dimension is 16x16x3=768, and the dimension of the linear projection layer is 768xN (N=768). Therefore, the dimension after passing through the linear projection layer is still 196x768, which means there are a total of 196 tokens, each with a dimension of 768. Here, a special character cls needs to be added, so the final dimension is 197x768. So far, a visual problem has been transformed into a seq2seq problem through patch embedding.

(2)Position encoding (standard learnable 1D position embeddings): ViT also requires the addition of position encoding, which can be understood as a table with a total of N rows. The size of N is the same as the length of the input sequence, and each row represents a vector with the same dimensions as the embedding of the input sequence (768). Note that the operation of position encoding is sum, not concat. After adding location coding information, the dimension remains 197x768.

(3)Multi-Head Attention: The output dimension of LN(Layer Normalization) is still 197x768. When applying multi-head self-attention, the input is first mapped to queries (Q), keys (K), and values (V). If there is only one head, the dimensions of Q, K, and V are all 197x768. If there are 12 heads (768/12=64), the dimensions of Q, K, and V are 197x64, with a total of 12 sets of Q, K, and V. Finally, concatenate the outputs of the 12 sets of Q, K, and V, with an output dimension of 197x768. After passing through a layer of LN, the dimensions remain 197x768.

(4)MLP [3]: Zoom in and out the dimension, 197x768 to 197x3072, and then zoom out to 197x768.

After each block, the dimensions remain the same as the input, both 197x768, allowing multiple blocks to be stacked. Finally, the output corresponding to the special character class

will be used as the final output of the encoder, representing the final image presentation.

The structure is shown as follows Fig2.2.

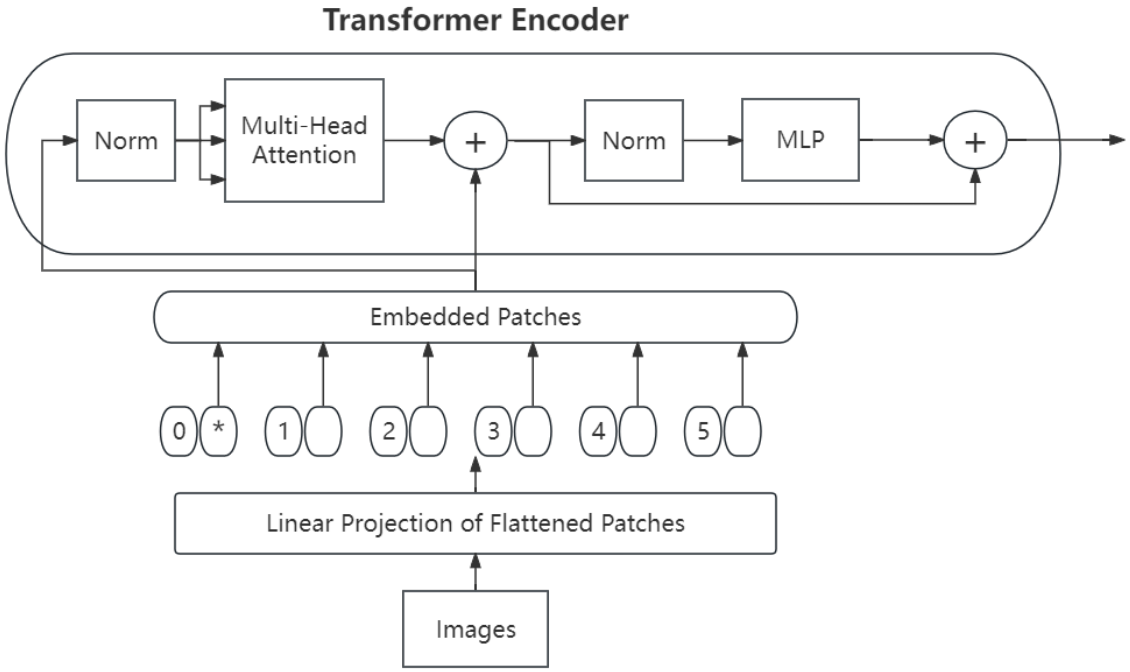

Figure 2.2: Overview of VIT

Vision graph neural network(VIG)

VIG [70], proposed in 2022, introduced a more flexible processing method that uses a graph structure to process images. An object can be seen as a combination of multiple parts; for example, a person can be seen as a combination of head, body, hands, and legs, and a fish can also be seen as a combination of head, body, tail, and fins. However, previous grid structures and sequence structures were not able to effectively demonstrate the connections between various parts, while graph structures were able to naturally connect various structures, thus achieving various downstream tasks in computer vision.

This research pioneered the representation of images in the form of Graphs and utilized graph neural networks to complete computer vision tasks. Each image is divided into different patches and treated as a node. Constructing a graph using these nodes can better represent irregular and complex objects in natural images. However, using only Graph Convolution [62] to process the graph structure of an image may have an oversmooth problem, resulting in poor performance. This research introduced additional feature transformations

within each node to encourage information diversity. Based on image representation and improved blocks, this research constructed two different structures of ViG networks, namely isotropic (similar to Transformer) and pyramid (traditional CNN) VIG networks. A large number of image recognition and object detection experiments have demonstrated the superiority of the ViG architecture proposed in this paper. The author hopes that this groundbreaking work on visual GNN [62] could serve as the basic framework for general visual tasks.

2.2 Monocular Depth Estimation

Depth estimation [71] [72] [73] is one of the common sub-tasks of multi-task learning, as 3D information often plays a good supporting role in many tasks. Although lasers and binoculars often achieve more accurate depth information, monocular depth estimation [74] [75] [76] is used in most multi-task learning processes due to its better scene adaptation capabilities. In this section, we will review the development of monocular depth estimation.

From the perspective of research directions, recent monocular depth estimation projects can be roughly divided into three categories. Initially, monocular depth estimation and camera pose estimation were treated as a unified task that relied on self-supervised learning based on the relationships between frames in image sequences [77]. Section 2.2.1 lists some classic tasks of monocular depth estimation based on self-supervised learning using image sequences. Later, many researchers incorporated additional visual cues to monocular depth estimation [78] to further improve the accuracy of monocular depth estimation. Relevant tasks are listed in section 2.2.2. Later, with the combination of convolutional neural networks(CNN) and transformers [79] [80], the bottleneck of the knowledge capacity of traditional convolutional networks was broken, enabling single-view depth estimation to achieve impressive results on multiple datasets [81] [82] [83] simultaneously. In section 2.2.3, some tasks [84] [85] that use transformers to improve the generalization ability of training results are listed.

2.2.1 Traditional Monocular Depth Estimation

With the emergence of CNN networks [86], a series of monocular depth estimation based on self-supervised learning [87] [88] have emerged based on traditional SLAM theory. Usually,

in such tasks, monocular depth estimation is combined with camera pose estimation. By utilizing spatial relationships and the disparity relationship between two images, depth and camera pose can be estimated.

SfmLearner

Zhou, et al. [89] used Slim in Tensorflow to call the VGG [90] network and performed depth and pose estimation separately. Building upon the traditional pose network, the decoder was added to generate a set of masks, which were described as having the ability to recognize dynamic objects.

SfmLearner is an unsupervised depth pose estimation. Two tasks have been implemented:

1. The network of VGG16 is used for depth pose estimation.

2. Based on the original pose encoder part, the decoder part is added, and the pose mask is generated layer by layer.

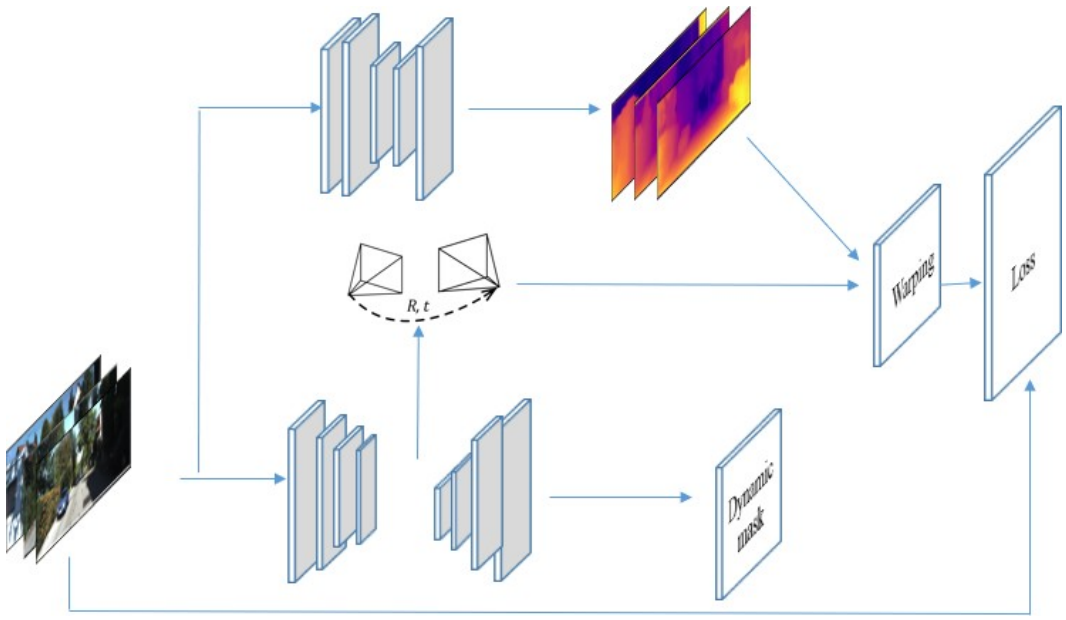

Figure 2.3: Overview of Sfmlearner

As shown in Fig 2.3, the network structure has two parts. For the depth estimation, it uses VGG16 [90]. For the pose estimation, the pose result is generated by the encoder part, and the decoder part is used to generate a dynamic mask of four scales.

GeoNet

GeoNet [91] is based on SfMLearner [89]. Compared to SfMLearner, GeoNet modifies and improves relevant network structure, input/output and loss function. The original VGG16 network structure in SfMLearner [89] was modified to ResNet50 [63], which increased the computation and improved accuracy. Additionally, by adjusting the parameters, the author adjusted the weight of the output of the encoder of the original depth of the network, to further improve the output of the depth of the network. At the same time, SfMLearner [89] carried out the estimation of depth and pose, but GeoNet outputs optical flow from the algorithm. However, GeoNet's optical flow is not generated directly by the network output but is obtained through the synthesis calculation of depth pose, which has room for improvement.

GeoNet has achieved two key breakthroughs:

1. Building upon, synthetic optical flow is obtained by depth pose calculation.

2. The original VGG16 network structure for depth pose estimation is replaced by ResNet50 to enable the system to extract more advanced features.

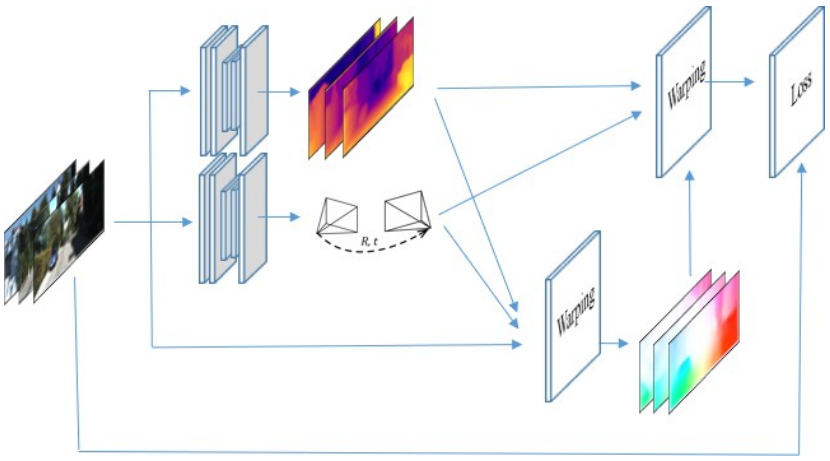

Figure 2.4: Overview of GeoNet

As shown in Fig 2.4, GeoNet uses two neural networks to perform depth and pose estimation separately. The optical flow is obtained through warping depth and pose.

3D Packing for Self-Supervised Monocular Depth Estimation

This paper [92], published in 2019, presents improvements to the traditional depth estimation structure from three aspects.

First, the researchers modified the network structure. In order to achieve more accurate depth estimation, researchers tried to optimize the structure in two aspects, the design of the loss function and the effective structure of the network. This paper introduces an innovative neural network approach to reduce the loss during convolution processing. The researchers tried to enlarge the input data before feeding them into the convolution layers. This method is called superpixel.

Additionally, the researchers made several changes in loss function design, the loss function includes a new part related to image speed. If the application environment is a car, a robot or a mobile phone with a sensor to get the speed data, this loss function can use those data to optimize the depth result.

In the end, a new dataset was introduced in this paper, called Dense Depth for Automated Driving (DDAD). The KITTI dataset [81] is a common dataset for SLAM researchers, which has some different groups of data. Both of them have their specific purpose, for example, the raw data set in KITTI is a data set for depth estimation, and people use the odometry data set in KITTI to train the neural network about depth estimation. They are all captured by a camera on a car which has a calibration system. For the DDAD dataset, greater emphasis was put on the calibration system to achieve improved performance.

This network achieves better results compared to previous researchers' work, but this structure can only provide depth and pose results. Since it does not have optical flow estimation, it cannot identify the motion objects and does not perform well in dynamic environments.

The structure of Pack-net is shown in Fig 2.5.

Self-supervised Object Motion and Depth Estimation from Video

This paper was published in 2019 [93], which came up with a new idea about supervised joint depth and pose estimation. For the traditional pose estimation, the input is always 3 images, i.e. the batch size is 3. But in this structure, the input data is 3 images with dynamic masks.

Since this research used images with segmentation information, the entire approach was supervised. The author employed a depth network and an object motion network to perform depth and object motion estimation respectively. The outputs from these two networks were

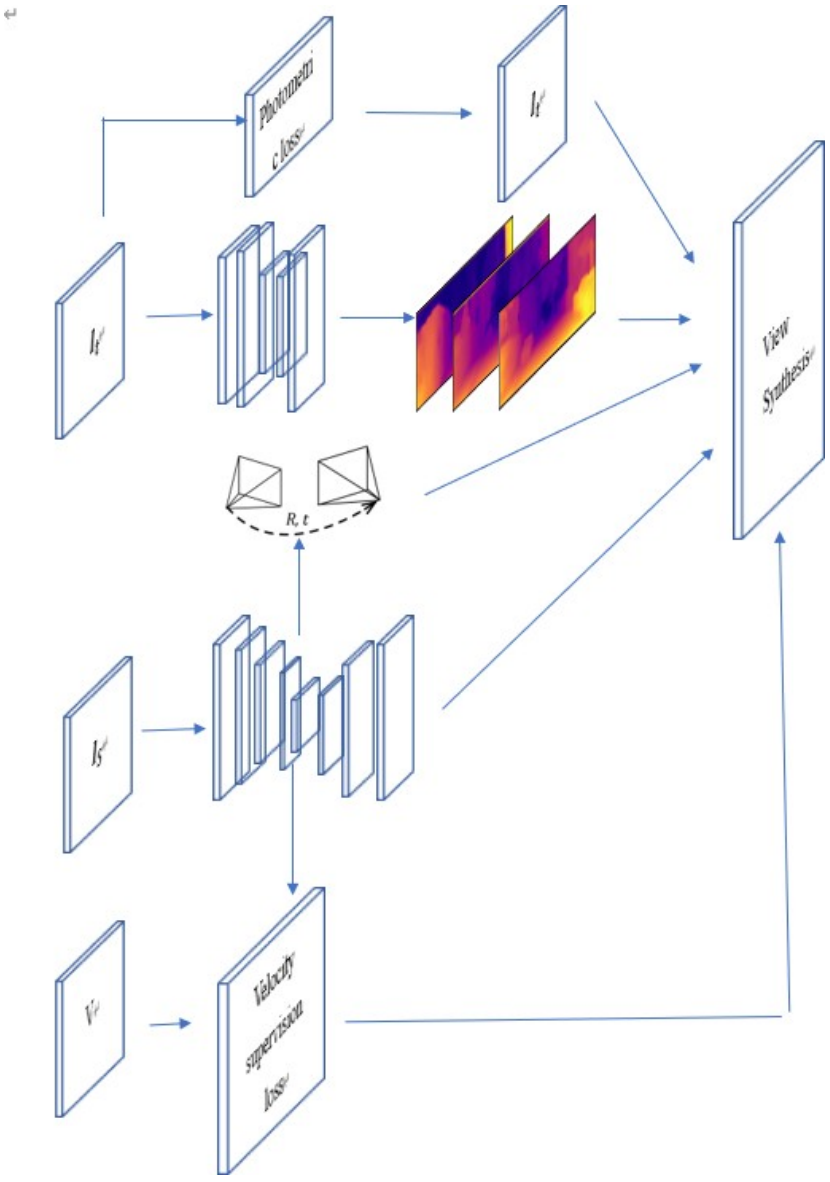

Figure 2.5: Overview of 3D PackingNet

then combined to perform joint estimation. The output data for the entire structure consists of two parts: dynamic object mask and static background.

This paper aimed to identify the background and objects in the image using depth and pose results. Similar to the approach proposed in UnDepthflow [87], they used the depth and pose result to generate a static sense. Then they applied their special method to distinguish the background and objects.

The structure is shown in Fig 2.6.

2.2.2 Joint-estimation Monocular Depth Estimation

With the maturity of monocular depth estimation, researchers have increasingly found that it is often difficult to effectively estimate the depth of dynamic objects in scenes. Therefore, a group of researchers have focused their studies on this part. Based on traditional self-supervised monocular depth estimation, they have improved the accuracy of dynamic object estimation in monocular depth estimation by adding masks and optical flow.

Undepthflow

Undepthflow is based on an article named PWC-Net [94], which first implemented the study and estimation of optical flow using KITTI-2012 and KITTI-2015. The code mainly performed two tasks:

1. Depth of stereo was directly applied to PWC-Net to conduct estimation for depth and pose (this part of the experiment had not been trained).

2. Depth and pose were used to generate a mask, and then the optical flow learned from the network was used to obtain the mask of dynamic objects. Finally, the generated mask was used to optimize the depth and pose parts to improve the accuracy of the depth pose.

The process of Undepthflow generating and applying the mask is divided into three parts in the code.

1. An approximate flow result is obtained by the estimation of optical flow.

2. Depth and pose estimation (this part is applied to PWC- Net)

3. Combining depth, pose and optical flow, a mask for dynamic objects was generated according to these three parameters, and the mask was used to improve the accuracy of depth and pose.

There are three main innovations of Undepthflow:

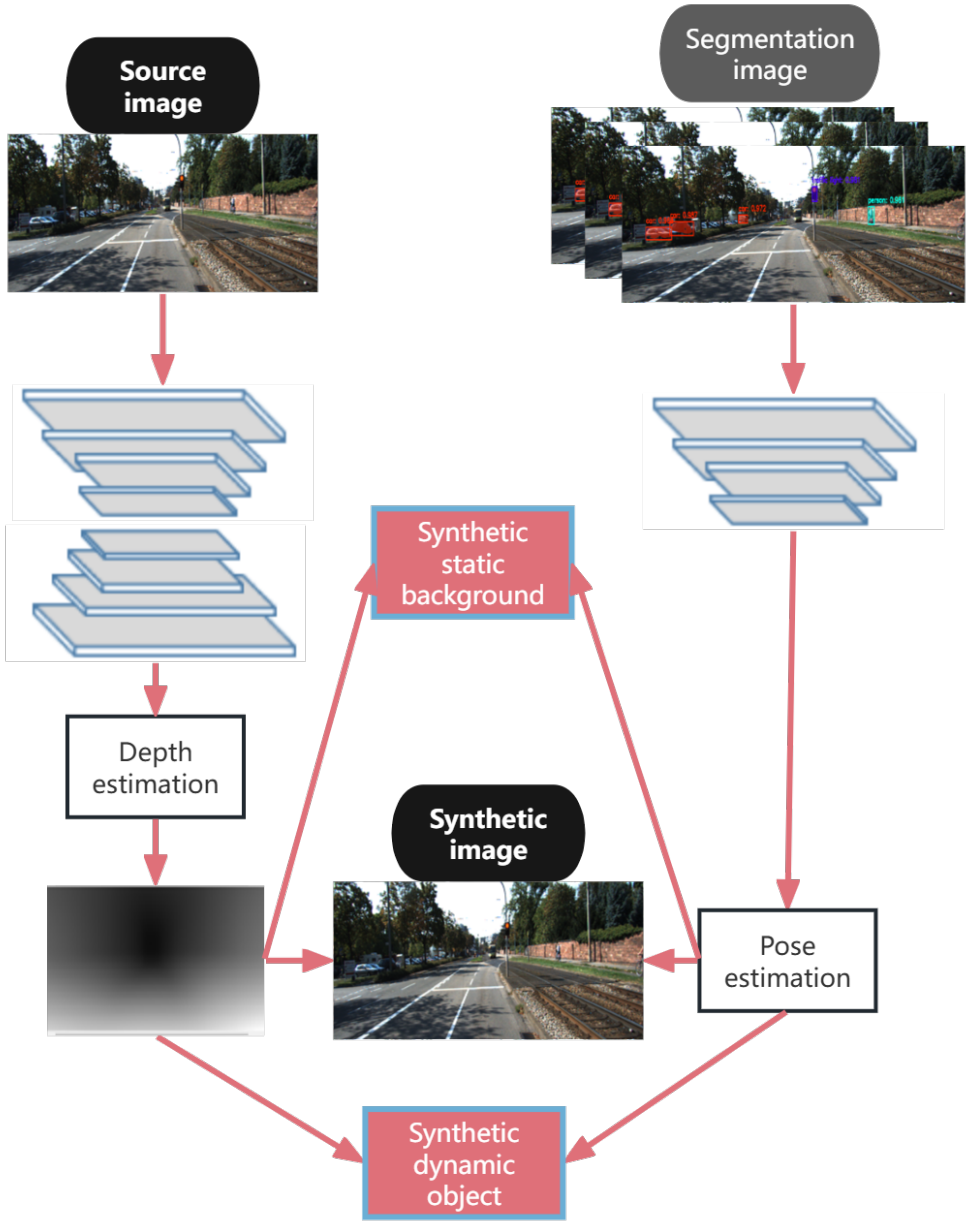

Figure 2.6: Overview of Self-supervised Object Motion and Depth Estimation from Video

1. Through KITTI2012 and KITTI2015, an independent neural network is used to achieve the optical flow estimation.

2. Through the algorithm, combined with the depth, pose, and flow, a mask is generated to recognize the dynamic object.

3. The newly emerging PWC-Net network architecture is used.

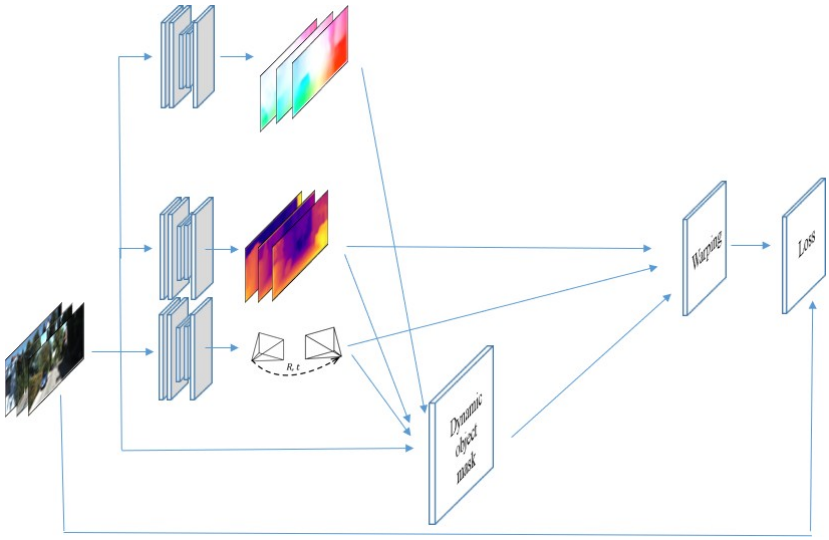

Figure 2.7: Overview of Undepthflow

As shown in Fig 2.7, the training process of Undepthflow consists of three steps. The first step is to generate optical flow using a neural network. The second step involves generating depth and pose with another neural network. Finally, joint training on depth, pose and optical flow. Undepthflow generates a dynamic mask by depth, pose and optical flow, and uses the mask to optimize the depth and pose.

Depth from Video in the Wild

Depth from video in the wild [95] is the work of Google researchers. This code is based on Vid2Depth [96] and Struct2Depth [97]. In general, the data set used in SLAM research is KITTI [81] data set. However, Depth from video in the wild utilizes its own data set (with dynamic object mask marking), and adopted its own algorithm, using the mask to improve the depth and pose accuracy.

Depth from video in the wild provides a set of ways to use masks. The input mask data is an image marked with dynamic objects. After the image is input into the program, the first step is to perform binarization. Subsequently, the binary mask image is converted into

a matrix of boolean values. and then the generated matrix and the original mask image are processed by bilinear interpolation to effectively utilize the mask.

DF-Net: Unsupervised Joint Learning of Depth and Flow using Cross-Task Consistency

This paper [98] was first published on 5 Sep 2018, and written by researchers from Virginia Tech and Stanford University.

This paper uses the joint estimation method to perform both depth estimation and optical flow estimation simultaneously. The main contribution of this paper is a specialized loss function. They used two groups of loss functions to perform the depth pose and optical flow estimation respectively. For the optical flow part, they used two loss functions to compute the pixel loss and smoothness loss respectively. Additionally, another loss function was used to identify invalid regions. They referred to this as the forward-backwards consistent loss.

This paper came up with a new structure of joint estimation about depth, pose and optical flow. It had complex input data in depth estimation, and also generated the forward and backward optical flow during the processing. But as the previous result, it did not make segmentation during the training processing, So, it also did not perform well in dynamic object estimation. Additionally, there is still some room for improvement in the network design.

The structure is shown in Fig2.8.

Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation

This paper [99] was first published on 24 May 2018, and written by researchers from Max Planck Institute for Intelligent Systems, NVIDIA and MIT.

This paper used the join estimation to conduct the depth, camera motion, optical flow and motion segmentation in one system. They designed some new loss functions to make each visual cue optimize others. So, the result received certain approval.

At the time of this dissertation, besides the depth, camera motion, and optical flow, motion segmentation is also useful. It can be used to identify the motion estimation of the objects. This can be used to generate dynamic masks. The idea discussed in this paper is also useful, as competitive collaboration is widely used in joint estimation.

Although this paper successfully performed joint estimation of depth, pose, optical flow

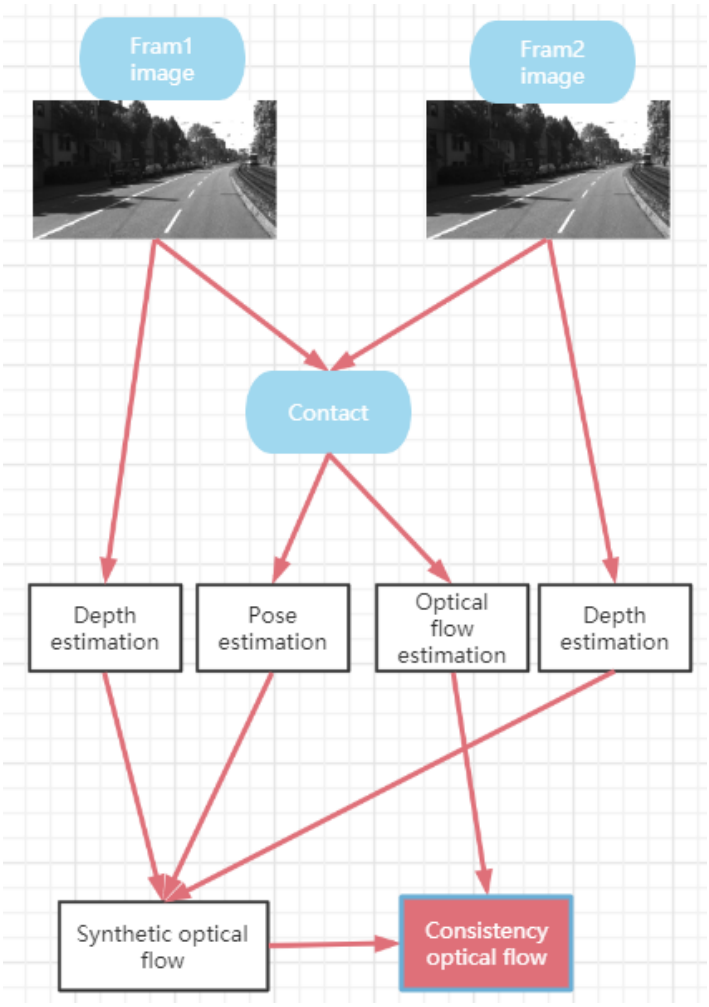

Figure 2.8: Overview of DF-Net

and segmentation, and some useful loss functions were used, the network structure in their four tasks could be further improved. Additionally, if a static sense could be created during the training process, the result could be better.

The structure is shown in Fig 2.9.

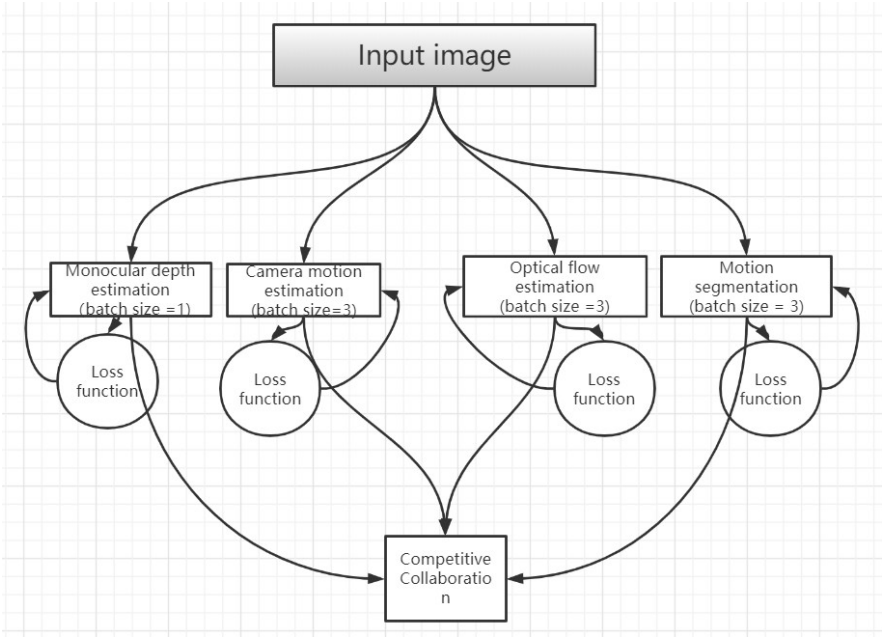

Figure 2.9: Overview of Competitive Collaboration

2.2.3 Transformer based Monocular Depth Estimation

In recent years, with the deep integration of transformers and traditional image tasks, more and more researchers have found that combining transformers with CNN networks can greatly improve the knowledge capacity of deep learning networks. Therefore, the research direction of monocular depth estimation has changed from traditionally obtaining good accuracy on a single dataset to obtaining good accuracy on multiple datasets at the same time, which greatly improves the scene adaptation ability of the monocular depth estimation network. This enables the effective application of monocular depth estimation in multi-task learning.

Midas

Midas [100] was published in 2019. The main idea of this paper is to mix multiple datasets for depth estimation, to realize zero-shot cross-dataset transfer (test on a new dataset that has never been seen before).

The article listed a series of RGBD data sets, and believed that training on a single data set is better when tested only on this data set, which is not enough to improve generalization, so the article chose to train on multiple data sets.

In addition, a new dataset was proposed, which used stereo matching to obtain relative depth from 3D movies.

However, obtaining data sets from movies faces the following challenges:

1. Unknown camera internal parameter.

2. The objects in the scene have moved.

3. Must be a 3D movie with a binocular camera.

To train so many divergent datasets at the same time, it is necessary to design a loss function that is flexible enough. The design of the loss function also faces some problems. This research designed a unique loss function to meet the needs of different scenarios.

It also improved the precision of monocular depth estimation and the generalization of monocular depth estimation by combining multiple data sets.

For livestock weighing, it is difficult to ensure that the depth values in the scene have a uniform distribution, so the excellent generalization ability of Midas can match our task needs well.

Depth Anything

Depth Anything [101] is a powerful foundational model for monocular depth estimation (MDE), designed to process images in various scenarios and provide more accurate depth predictions. This study greatly improved the model's generalization ability and robustness through the use of large-scale unlabeled datasets. The following is a review of its core ideas, technical methods, and application scenarios.

The model constructed an unprecedentedly large training dataset by using over 62M of unlabeled images and 1.5M of labeled images, significantly improving the depth prediction ability of the model. This model can not only perform well in zero sample learning, but also further improve its performance on multiple standard data sets through fine-tuning, such as NYUv2 [83] and KITTI [81].

The core innovation of the model lies in how to effectively utilize unlabeled data for deep estimation training. Unlike previous models that relied on annotated data, this model

has designed a data engine that can automatically collect and annotate large-scale unlabeled images. In addition, researchers have proposed two key strategies to achieve scalability and optimization of large-scale datasets:

1. Data augmentation and challenging objective optimization: By strongly perturbing unlabeled images, the model was forced to actively learn more visual knowledge during the training process, thereby obtaining more robust representations.

2. Auxiliary supervision mechanism: Introducing pre-trained encoders and using auxiliary supervision strategies to ensure that the model can inherit rich semantic prior information, enhancing the model's ability to understand complex scenes.

These methods not only expanded the data coverage range, but also reduced the generalization error of the model, making it perform well in depth estimation in various scenarios.

The model achieved leading performance in zero-sample learning tests on multiple publicly available datasets. For example, it established new benchmarks on the NYUv2 [83] and KITTI [81] datasets and outperformed previous optimal models such as MiDaS v3.1 [84] and ZoeDdepth [102] in both relative depth estimation and absolute depth estimation tasks.

This model has also demonstrated strong potential in video depth estimation and image generation tasks. By combining with ControlNet [103], researchers retrained a deep-based ControlNet model, significantly improving performance in video editing and generation tasks.

Although 'Depth Anything' demonstrated outstanding performance, it still faces some challenges. Firstly, although synthetic image data provides highly accurate depth annotation, the distribution deviation between synthetic data and real-world images makes it difficult for the model to migrate from synthetic data to real-world data. In addition, the scene coverage of synthesized data is limited, and the generalization ability of the model in complex and ever-changing real-world scenarios still needs further improvement. A powerful monocular depth estimation base model has been successfully constructed through the use of large-scale unlabeled data and innovative training strategies. Its excellent performance on multiple datasets indicates the enormous potential of using large-scale datasets for deep learning model training. In the future, with the introduction of more innovative methods, monocular depth estimation technology will be applied in a wider range of fields.

2.2.4 Discussion

Limited by the knowledge capacity of the backbone network, the generalization ability of monocular depth estimation based on self-supervised learning was often weak at first, so the pre-trained model could only meet the needs of a single scenario. With the application of joint estimation, the results of monocular depth estimation have been greatly improved, but the problem of weak generalization ability has not been substantially changed. In recent years, with the wide application of transformers in monocular depth estimation-related tasks, the knowledge capacity of the model has been greatly improved, and the generalization ability of the pre-trained model has been greatly improved, laying the foundation for the direct application of monocular depth estimation pre-trained models in multi-task scenarios.

Table 2.1: Comparison of different monocular depth estimation methods

Method	Model accuracy	Parameter quantity	Generalization	Learning Method
Traditional Method	Relatively low	Relatively low	Weak	One-stage self-supervision
Joint-estimation	Relatively high	Relatively low	Weak	Multi-stage self-supervision
Transformer based	Relatively high	High	Strong	Multi-mode

2.3 Object Detection and Object Segmentation

In many multi-task application scenarios, a pre-task is often required to remove the background and highlight the effective information. Usually, object segmentation [104] can accurately remove the background and provide effective support for subsequent feature extraction, but compared with object detection [105] [106] [107], such algorithms often run slowly and have higher labeling costs. Therefore, in some scenarios, object detection [108] [109] has more advantages. Compared with object segmentation, object detection [110] [111] has faster-running efficiency and lower computing power requirements. The following introduces some key tasks in these two tasks.

2.3.1 Object segmentation

The task of object segmentation is to segment the object in the image along the boundary. Before the advent of machine learning, object segmentation relied on the visual characteristics

of objects in images for segmentation. This approach often has unclear objectives, though it has low application cost due to the lack of labeling steps. Later, with the emergence of machine learning, machine-learning-based object segmentation appeared. Some object segmentation requires a pre-detection task, which first removes some background through object detection, and then uses an object segmentation network to perform more accurate segmentation on the object. Later, the improvement of the deep learning networks led to the advent of an end-to-end object segmentation approach. The following is a brief introduction to some representative parts of these works.

What is an Object

What is an object [112] was published by B Alexe, T Deselaers, V Ferrari in 2010.

This article is considered to be one of the pioneers in the field of object segmentation. It proposed four classic objectness measures, which are still inspiring many CV studies to this day. These four measures are:

Multi-scale Saliency(MS) This metric is used to find prominent objects in an image, and the method used is from a paper called "Saliency Detection A Spectral Residual Approach", This research found that prominent objects in an image have a significant response in the frequency domain, so these prominent objects can be found through the formula of spectral residuals. Based on this paper, the author has extended it to multiple scales, which means resizing an image to multiple sizes and then performing saliency detection separately.

Color Contrast(CC) CC is an indicator that measures the similarity between a box and its surrounding area. Generally speaking, a box containing a complete object is not similar in color to its surrounding area.

Edge Density(ED) ED measures the edge density near the box boundary. Its calculation method is to see how many pixels in the sub-regions inside the box are judged as edges, and edge detection is obtained through the Canny operator. The logic of ED is also very intuitive. Objects usually have closed contours, so the higher the proportion of edge pixels within a box, the more likely it is to contain a complete object.

Superpixels Straddling(SS) Firstly, the concept of superpixels comes from a paper on image segmentation called 'Efficient Graph-Based Image Segmentation [113]'. In this paper, pixels are merged based on their color similarity, resulting in similar regions being merged

into a single superpixel. Finally, the entire image is segmented into many superpixels,

BoxInst

BoxInst [114] was published in 2020. This paper proposed a very elegant weakly supervised instance segmentation method, which not only replaced the loss function in the fully supervised instance segmentation algorithm, but also could achieve fairly good weak supervised results. Compared to the previous weakly supervised methods, this research is considered groundbreaking. Moreover, the concept behind this is not complex, and the methods are very concise and effective, and can even be applied in semi-supervised scenarios.

This research proposed a high-performance method for implementing instance segmentation, which only required the use of box annotations during training. The researchers only used a simple design to achieve significant performance improvement. The core idea of this paper is to redesign the mask loss function in case segmentation, without modifying the segmentation network. The new loss function does not need mask annotation.

It is achieved through two loss items:

1)Minimize the error between the projection of ground truth and the predicted mask.

2)Using colour similarity to attribute adjacent pixels with similar colours to the same instance.

The researchers only used ResNet-101 backbone network and 3 × The training mechanism can achieve 33.2 mask AP on COCO test dev (vs. 39.1 for fully supervised AP). The experimental results on COCO and Pascal VOC in this research showed that this method greatly reduces the difference between weakly supervised and fully supervised instance segmentation results.

SOLO-V2

SOLO-V2 was published in 2002. The main task of SOLO-V2 [104] is instance segmentation. There are generally two methods for instance segmentation. One is top-down, which detects bounding box first, and then mask segmentation is performed within each bounding box, such as Mask R-CNN. The second is the bottom up method, which first segments each pixel and then classifies it. The two methods introduced in this paper take a different approach by directly splitting the instance mask, making it a box-free method. The paper mainly has two innovative points:

1. The author further introduced a dynamic mechanism to dynamically learn the mask head of the object divider. Mask branches are decoupled into kernel branches and mask feature branches, and convolution kernel weights are learned.

2. The author proposed Matrix NMS to reduce forward reasoning time.

As a partitioned network, Solo-V2 still has the following shortcomings:

1. Compared with Deep-Lab-V3, the accuracy still needs to be improved.

2. Because it was done in the box-free mode, there are often outputs that are completely unrelated to the results.

The experiment of Solo-V2 is worth learning from:

1. Through the experiment of Solo-V2, the overall process and loss design of the object segmentation algorithm is understood.

2. Matrix NMS reduces the computation without affecting the accuracy. This design idea of network structure is worth using for reference.

2.3.2 Object detection

ared to object segmentation, the objective of object detection is relatively simpler, as it only requires identifying the region in the image that contains the detection objects. Therefore, object detection tasks generally run faster than object segmentation. Based on whether the shape of the anchor box is defined before output, object detection is divided into two categories, one that does not have a pre-defined anchor box and the other that predefines the aspect ratio of the anchor box in advance. In addition, due to the high labeling cost of such tasks, semi-supervised and weakly supervised object detection tasks have become key research directions of object detection.

WSOD

WSOD [115] was published in 2019. CNN has achieved great success in tasks such as image classification, object detection, and semantic segmentation. Strong supervised object detection algorithms have been widely studied and achieved high accuracy, as well as a large number of publicly available datasets with annotated information. However, accurate target-level annotation requires expensive labor costs, and training an accurate object detection model requires a huge amount of data.

This paper focuses on weakly supervised object detection (WSOD), which only uses im-

age level category labels, thereby saving a lot of annotation costs. Due to the lack of precise annotation information, this problem has not been effectively addressed, and its performance cannot be compared to strongly supervised methods.

The so-called end-to-end WSOD merges CNN and MIL(Multiple Instance Learning) into a unified network to solve the task of weakly supervised object detection. Dibaet et al. proposed an end-to-end cascaded convolutional network that performs weakly supervised object detection and segmentation in a cascaded form. Bilenet et al. proposed WSDDN, which selects positive and negative samples by summarizing the scores of classification and detection branches. Based on WSDDN, Tanget al. proposed an online instance classifier refinement (OICR) algorithm to alleviate local optimization problems, and also proposed region clustering learning (PCL) to improve the performance of OCIR.

Fcos

Fcos [7] was published in 2010. It is a one-stage anchor-free object detection algorithm, with its main selling point being anchor-free. Object detection is achieved by regressing the distance between each position on the feature map and the target box. This research addresses the problem of a location falling within multiple target boxes through multi-scale and regression amplitude constraints. To address the problem of excessive target boxes, this research proposes a centre-ness method that learns a center-ness score for each position and multiplies it with the predicted category score, using this product as a non-maximum suppression input.

Compared to traditional anchor-based object detection, FCOS has the following advantages:

1,The biggest advantage of FCOS is that it does not require a prior box, greatly reducing the sample size and parameter size, as well as the computational complexity.

2,FCOS also performs well in small object detection, increasing the number of predictable boxes through multi-scale detection.

3,The biggest innovation of FCOS lies in the center ness module, which is the main part that distinguishes it from YOLOv1 [107].

Yolo-V4

YoloV4 [42] was published in 2020. Object detection tasks are applied in many industries, and Yolo-V4 has a more perfect training system than SSD making it the most widely used object detection algorithm.

In terms of data argumentation: in addition to incorporating some data argumentation methods including Resize, Blur, Noise, Flip, and color change, Yolo-V4 inherits and retains Yolo series methods such as CutMix, MixUp, and Copy And Paste. For the first time, data argumentation methods including Mosaic and adaptive image scaling are added, which further improves the efficiency of data utilization.

During the initialization of anchor, an adaptive anchor frame is added for the first time, allowing more accurate bonding boxes for targets with different aspect ratios in the image.

In the aspect of network structure: the FPN structure, which is commonly used in object detection, is effectively combined with PaNet to reduce the information loss in the encoding and decoding process.

Loss design: based on the traditional IoU, SIoU and DIoU are added as loss functions to further improve the convergence speed of the training process.

The shortcomings of Yolo-V4 are as follows:

1. In terms of running speed, Yolo-V4 still has room for improvement. The backbone network can be trimmed to meet the task requirements of individual scenarios.

2. The model is too large, which limits the number of processes that can run simultaneously during deployment. Although Yolo-V4 has a small model, there is still a gap between the accuracy of the small model and the large model.

3. The adaptive anchor frame can not effectively meet the requirements in many scenarios. In many scenarios, due to the special aspect ratio of the label anchor, the adaptive anchor frame cannot find the bounding box with the largest IoU value.

At the same time, Yolo-V4 also makes the following valuable improvements:

1. Adaptive image scaling, while normalizing the image, Yolo-V4 preserves the integrity of the image information as much as possible, which is worthy of reference.

2.The unique FPN [116] design of Yolo-V4 can reduce information loss during convolution in many scenarios.

2.3.3 Discussion

Although object detection and object segmentation are typically used as prerequisites for background removal in multi-task learning, their selection depends on the specific characteristics of the data and the requirements of the task, as the methods and tasks involved can vary.

Table 2.2: Comparison of Object Detection and Object Segmentation Algorithms

Method	Operation speed	Labeling	Data demand	Accuracy
Unsupervised Segmentation [112]	Slow	No	High demand	Relatively low
Weak-Supervised Segmentation [114]	Relatively slow	Less	Relatively high	Relatively low
Supervised Segmentation [104]	Relatively slow	High	Relatively high	Heigh
Weak-Supervised Detection [115]	Relatively slow	Less	Relatively high	Relatively low
Anchor-free Segmentation [7]	Relatively fast	High	Less demand	Heigh
Anchor-based Segmentation [42]	Fast	High	Relatively high	Heigh

2.4 Keypoint detection

Key point detection [117] [118] [119] is also used to highlight the target area in some scenarios. Compared with object segmentation and object detection, it performs better when the target is small and the image resolution is high.

2.4.1 CenterNet

CenterNet was published in 2019. Compared with YOLO [120], SSD [121] and fast-RCNN [109], which relied on a large number of anchor detection networks, CenterNet is an anchor-free object detection network that has advantages in speed and accuracy, and is worth learning.

The model calculation process of CenterNet is as follows:

1. The image is scaled to the size of 512x512 (the long side is scaled to 512, and the short side is filled with 0), and then the scaled 1x3x512x512 image is input into the network.

2. The feature 1 size of the image is 1x2048x16x16 after feature extraction by resnet50.

3.Feature1 passes the deconvolution module Deconv, and the size of feature2 is 1x64x128x128 after three times of up sampling.

4. Feature 2 is sent into three branches for prediction. The predicted heap size is 1x80x128x128 (representing 80 categories), the predicted length and width are 1x2x128x128 (2 representing length and width), and the predicted center point offset size is 1x2x128x128 (2 representing x, y).

Another thing worth noting about CenterNet is that its data argumentation part uses the affine transformation warp affine, which is actually clipping the original image, and then scaling to the size of 512x512 (long side scaling, short side zeroing). In the actual process, a center point and the length and width of a clipping are determined first, and then affine transformation is carried out.

CenterNet [122] mainly has the following points for reference:

1.Special data enhancement methods. In data argumentation, CenterNet can enrich the background form without affecting the target form, and effectively achieve data expansion.

2.CenterNet effectively combines key point detection tasks with object detection tasks in the way of heatmap.

2.4.2 Hr-Net

Hr-Net was published in 2019 [123]. It is a paper on human 2D key point detection. In such tasks as human posture recognition, it is necessary to generate a high-resolution heatmap to detect key points. This is different from the requirements of general network structures, such as VGGNet [90], because the final feature map resolution of VGGNet [90] is very low and the spatial structure is lost.

In the traditional 2D human pose estimation task, the method of obtaining high resolution is mostly to reduce the resolution first and then increase the resolution. U-Net [59], SegNet [124], DeconvNet [125] and Hourglass [126] are essentially of this structure. HrNet parallels feature maps with different resolutions. On the basis of paralleling, it adds fusion between feature maps with different resolutions.

Due to the increase in the number of convolution layers in the decoder process, the overall running speed is much slower than other key point detection algorithms. Its 2D Key point detection nature means it can not meet the actual needs of human pose estimation in many scenarios. However, the high-resolution network in HrNet performs well in pixel2pixel tasks [127] [128], including depth estimation [78], which is very instructive.

2.5 Discussion

This chapter lists several commonly used methods of multi-task learning, focusing on two aspects: multi-task learning structure and sub-tasks involved. When introducing sub-tasks, they are categorized into three types based on their functions: monocular depth estimation, object detection and segmentation, and key point detection.

When enumerating the multi-task learning structure, the multi-task learning structure is introduced from four aspects: scenario characteristics, structural design, sub-task selection and merged network. First, starting from the scenario characteristics, some methods that use the coordinated use of sub-tasks in multi-task learning to improve the accuracy of existing machine learning tasks are listed, as well as some multi-task learning structures that realize special tasks by merging existing deep learning sub-tasks. Then, for the structure of multi-task learning, some multi-task learning structures with serial and parallel designs are listed. When exploring the selection of sub-tasks in multi-task learning, this chapter cites some cases in which the accuracy, generalization ability and running speed of multi-task learning are improved through sub-task selection. Finally, a detailed description of the more characteristic network structures in computer vision multi-task learning is given, including the traditional CNN structure, the subsidiary module of deep learning, the transformer-based deep learning visual model and the GNN-based spatial feature extraction method.

When introducing monocular depth estimation, we divided it into three stages according to its development history. The first stage is monocular depth estimation based on self-supervised learning, which is self-supervised learning by establishing the relationship between the previous and next frames of the video or the relationship between the left eye and the right eye. The second stage introduces some methods of reducing depth estimation errors by adding visual clues such as dynamic object masks and optical flow. Finally, some transformer-based monocular deep learning architectures are introduced. This type of algorithm greatly improves the generalization ability of the model by training on multiple data sets, so that the model can achieve high accuracy in different scenarios.

When introducing object detection and object segmentation, for the object segmentation part, unsupervised object segmentation, weakly supervised object segmentation and object segmentation based on supervised learning are given as examples. When introducing ob-

ject detection, weakly supervised object detection, anchor free object detection and classic anchor based object detection are given as examples.

When summarizing the key point detection work, two key point detection tasks based on supervised learning are given as examples. Among them, HrNet [123] has been effectively applied in various tasks due to its outstanding performance.

By summarizing the above multi-task learning and the commonly used subtasks in computer vision, the research foundation for the entire paper is laid. Starting from the next chapter, the main research work of this paper will be introduced.

Chapter 3

A Breeding Animal Detection Method from Images With Grided Heatmaps

In practical scenarios for detecting objects in images where we cannot collect sufficient effective annotated data, we propose a new method that can utilize pre-trained classification network models to implement object detection tasks. Our method generates multiple scale gridded heatmaps from a pre-trained classification network, which can effectively integrate various pre-trained features of different categories without training. We also utilize similarity segmentation results to further refine the object detection performance. Our experiment results on COCO [17], VOC and our datasets show that our method can produce better detection results compared to some existing object detection networks [6] [7] [8] when the number of training data is less than 400.

3.1 Introduction

Data scarcity is a challenging issue for many practical applications of deep learning techniques. In animal breeding management applications, using camera images to detect the breeding animals plays a vital role in improving the quality of management services and maintaining low operation cost. In this chapter, we focus on the application of deep learning networks for detecting breeding animals from images where we cannot collect sufficient effective annotated data.

Object detection from images using deep learning networks [6] [7] [129] [8] is one of the mainstream tasks in computer vision. It can generate a bounding box for an object in input

image and has been widely applied in many scenarios. An effective object detection network requires a large amount of annotated data and a long period of time for training. It is often difficult to obtain a large amount of effective annotated data, which has led to the emergence of some weakly supervised object detection networks [129] [8]. Weakly supervised object detection (WSOD) networks are based on datasets used for classification task, but they rely on the ingenious design of loss functions to couple the classification task with object detection task together. In this way the object detection task can be effectively trained while training the classification network. However, weakly supervised object detection [129] [8] tasks also require a large amount of data for training. Their results only rely on classification data and it is difficult to generalise the results in object detection scenarios.

This chapter proposes a new object detection method based on a pre-training network model used for classification tasks [130] [49] [63] [48]. The main idea is that we do not need to perform additional training for the object detection task in our proposed object detection network, but only based on a pre-training network model for classification tasks. Then our proposed network can be used for new datasets, such as animal detection which could be used for animal breeding management systems. Due to the lack of training datasets for such applications, it is key to develop new methods to save the cost and administration burden. In our method, we take RGB images as input and do not need to train our object detection network. We just use the classification model pre-trained on ImageNet1K [60] for our object detection task. We exploit the heatmaps produced from the classification model, and the similarity segmentation from the superpixel technique [112].In additional experiments, we further implemented counting using this method and evaluated our counting results.

3.2 Related Works

Most deep learning networks for object detection tasks require lengthy training processes and costly data collection processes while similarity segmentation techniques are lack of sufficient robustness. Below we review some related research works and they are categorised into three subsections: supervised object detection, weakly supervised object detection, and similarity segmentation.

3.2.1 Supervised Object Detection

Object detection is one of the key tasks in computer vision. It requires a large amount of annotated data for training, and the training cycle is generally long. Although pruning [131], distillation [132], and quantification methods [133] could reduce the computational complexity of models. The demand for large annotated datasets remains a challenge in many applications. The use of data augmentation methods such as mosaic [6], copy and past [6], and cut mix [6] has improved the efficiency in object detection training. However, the effectiveness of these data augmentation methods is limited. The emergence of adversarial generation networks provides the possibility of generating more data. However, both diffusion [134] and adversarial generation models [58] require a large amount of data for training, making it difficult for such methods to be effectively applied in some niche scenarios where only limited number of images are available. Meanwhile, due to the lack of real external data, diffusion models and adversarial generation models find it difficult to generate data distributions beyond the original data.

3.2.2 Weak Supervised Object Detection

Weak supervision object detection (WSOD) aims to utilize some datasets with incomplete or indirect annotations for learning [8] [129] [135]. They can be divided into semi-supervised learning [136] or transfer learning [137]. Semi-supervised learning is aimed at the scenarios where there is only a small amount of labeled data and a large amount of unlabeled data. Transfer learning [137] applies the information learned by pre-trained models to new problems. It tries to generalise the object detection networks over different domains. It can accelerate the learning convergence of network models and improve the training efficiency. In most cases [138] [6], pre-training models are used with all the fixed parameters in front layers, and only the parameters of full connection layers are trained with small number of training samples. Due to the changes to model structure and parameters, this potentially could lead to the information loss.

3.2.3 Similarity Segmentation

Compared to deep learning networks, which requires a large amount of data for training, Similarity segmentation [112] usually only requires a very small amount of data to obtain

stable results. However, in order to obtain effective output, they often require a series of complex operations on images. This makes it difficult to be effectively applied [95]. In some cases [139], the results are not very reliable due to the manual tuning of algorithm parameters.

3.3 Methods

In this section, we will provide a detailed description of our object detection method (see Figure 3.1). We utilise a classification model pre-trained on the ImageNet1K [60] and VOC datasets [16]. Given an input image, it will generate object heatmaps for each category. Totally the number of categories is K (K=1000 in ImageNet1K). A heatmap is a 2D grid of scores associated with a specific object, which shows how important each location is with respect to the object under consideration. Our method generates multi-scale gridded heatmaps for the input image and for the sample images from each category, which are then used to test the similarity for object detection. We also apply the superpixel similarity segmentation technique [112] to the input image. It produces a similarity segmentation image, which is then used to refine the bounding box generated from gridded heatmaps.

3.3.1 Multi-scale Gridded Heatmap Generation

The overall process of our object detection is shown in Figure 3.1. An image is captured for a breeding animal cow. The image is gridded into small patches under multiple scales. Then the score of each grid is computed by the pre-trained classification network. A gridded heatmap is generated for each scale. The size of the grid patch also affects the detection performance. We grid the input image for multiple scales to preserve the structural property of the object (see Figure 3.2).

Then we sample images for each category in ImageNet1K. All the sampled images are gridded and processed by the classification network to generate the corresponding heatmaps in the same way as for the input image.

The use of grids is effective when the full body of animal is not visible. When only part of the body is visible, the gridded image can generate a score for each patch. A heatmap with score distribution where higher scores provide an indication to the object under consideration is produced.

After all the heatmaps are generated for input image and sampled images, the patches with the highest score are selected for the feature similarity test in next subsection.

3.3.2 Feature Similarity

The feature similarity is computed between the patches with the highest score in the input image and the patches with the highest score in the sampled images from all categories. The features are extracted from the original input and sampled images, including Hist [140], Hog [141], Lbp [142] and Phash [143] from raw image and the Hist features from depth map. The Phash similarity is computed by using equation (3.1). The Hist, Hog, and Lbp similarities are computed by using equation (3.2). f_1, f_2 represent the corresponding feature vectors extracted from two patches, and $\bar{f}_1, \bar{f}_2$ denote the means of the feature vectors. When these similarity indicators are greater than the thresholds, the similarity between two patches is high and they are selected as the region proposals. When all the patches between the input image and sampled images are tested with the feature similarities, a detection result which contains multiple region proposals for the object is generated (see Figure 3.3).

$$\text{Similarity} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}(f_1(i), f_2(i)) \tag{3.1}$$

$$\begin{aligned} v_1 &= \textstyle\sum_{i=1}^{n}(f_1(i)-\bar{f}_1)^2 \\ v_2 &= \textstyle\sum_{i=1}^{n}(f_2(i)-\bar{f}_2)^2 \\ c_1 &= \sqrt{v_1 \times v_2} \\ c_2 &= \textstyle\sum_{i=1}^{n}(f_1(i)-\bar{f}_1)\times(f_2(i)-\bar{f}_2) \\ &\text{Similarity} = \frac{c_1/c_2+1}{2} \end{aligned} \tag{3.2}$$

where n is the length of the current feature vector.

Similarity Segmentation

The superpixel technique [112] segments an image into regions by considering appearance similarity measures. They generate regions with similar looking pixels. We utilise the superpixel technique [112] to perform the segmentation on input image (see Figure 3.4). Usually, an object is composed of one or more superpixels. A superpixel can only contain one object or part of an object. After obtaining the superpixels and proposal bounding boxes of the im-

age, we determine the proportion of superpixel area in each proposal bounding box. When the proportion exceeds a threshold, we determine that the superpixel belongs to the object. Finally, a final bounding box is generated for the object.

3.4 Experiments

Our experimental results were based on object detection data from COCO [17], VOC [16], ImageNet1K, and the dataset collected by ourselves. Our dataset consists of 900 images containing a Corgi dog from different scenes and some cows captured by ourselves. We annotated our images for testing our method and training for other comparison networks.

3.4.1 Evaluation metrics

We use the *MIoU* value for evaluation. First we compute *IoU* by using equation (3.3), where B represents the area of currently predicted bounding box and G represents the area of real bounding box.

$$\mathbf{IOU} = f_{iou}(B,G) = \frac{B \cap G}{B \cup G} \tag{3.3}$$

Then we average the IoU values of all the images in a dataset to get the average Intersection over Union (MIoU) by using equation (3.4)

$$\mathbf{MIoU} = \frac{1}{n}\sum_{i=1}^{n} f_{iou}(B(i),G(i)) \tag{3.4}$$

The *MIoU* value is less than or equal to one. Generally, the larger the MIoU, the more accurate the object detection algorithm is.

3.4.2 Ablation Experiments

In order to prove the effectiveness of our method, we conducted a series of ablation experiments from three aspects: classification network selection, feature similarity, and similarity segmentation.

First, we evaluated the object detection results generated with different classification networks. We used MobileNetV3 [130], ShuffleNet [49], Res18 [63], Res101SE [144], RepVGGA0 [48], RepVGGD2SE [48], MobileViT-S [145], PVT-V2-B3 [146], Resnet50-spark [147], and Riformer-m48 [148] to generate multi-scale heatmaps. The patch with a

score greater than 50 in the heatmaps was selected as a bounding box and compared with the category label. The comparison results are shown in table3.1. It can be seen that Res18 [63] preformed better in terms of MIoU. The number of network parameters is closely related to the testing time (flops). The speed performance of Res18 [63] is reasonable fair when compared with other networks.

Table 3.1: Impact of different classification networks on object detection task

Network	MIoU	Parameters(M)	Flops(G)
ResNet-18 [63]	**0.55**	**11.69**	**1.82**
MobileViT-S [145]	0.52	5.59	0.184
RepVggA0 [48]	0.48	9.11	1.52
PVT-V2-B3 [146]	0.48	45.2	6.7
MobileNetV3 [130]	0.47	5.48	0.23
RepVggD2SE [48]	0.45	133.33	36.56
Shuff1eNet [49]	0.43	1.87	0.15
Resnet50-spark [147]	0.43	23.52	1.31
Riformer-m48 [148]	0.40	73.47	11.59
ResNet-101SE [144]	0.36	49.33	7.86

To test the effectiveness of feature similarities used, we compared different similarities, the combinations of Hist [140], Hog [141], Lbp [142], Phash [143] and depth. The testing results are shown in Figure 3.5. The abscissa is the number of images used, and the ordinate is the MIoU value. It can be found that the best object detection results can be obtained by using the combination of Lbp and depth [100] to compute the similarity between objects. This shows that color texture can describe object features more effectively than other methods, which could perform even better when combined with depth information. The depth estimation result produced by monocular depth estimation can also be used as an effective indicator to measure target similarity. However, color and brightness are weakly robust in measuring target similarity.

To test the effectiveness of similarity segmentation (superpixel technique), we compared the effects of feature similarity with similarity segmentation (ss) or without similarity segmentation. The testing results are shown in Figure 3.6. The abscissa is the number of images used and the ordinate is the MIoU value. It can be found that the superpixel technique [112] can effectively improve the detection results in this task.

3.4.3 Main Results

We evaluated our object detection algorithm on COCO [17], VOC [16], and own dataset we collected ourselves. Three object detection networks were used for the comparison, namely YOLOX [6] for supervised object detection, Fcos [7] for anchor free object detection, and WSOD [8] for weakly supervised object detection. We first used the training sets of COCO [17] and VOC [16] to train these three networks. Then we compared their testing results with our results. Our network did not need to train.

Figure 3.7 shows the evaluation results on the dataset we collected. In the comparison, we attempted to train the classification network used in our object detection process. The brown curve in the figure represents the test results of this method. Through the comparison, we found that although the classification network was trained using a dedicated dataset, this approach did not significantly improve the results. Figure 3.8 shows the evaluation results on the COCO [17] dataset when the object is cow. Figure 3.9 shows the evaluation results on the VOC dataset when the object is cow.

Through comparison, we found that when the training data is less than 200 images, our algorithm performed better than other object detection algorithms in terms of MIoU. This result provides clear evidence that we can use our proposed method to detect the animals in breading management applications with low cost. No labeled training data is required in our network. A demonstration for two images from Corgi dataset with comparison networks is shown in Figure 3.10.

In general, through comparison, we found that compared with YOLO [6], the object detection method Fcos [7] using the anchor free strategy tends to converge faster with a small amount of data. In contrast, weakly supervised object detection [8] often performs poorly with limited amount of data because it lacks a large learning rate during the supervised learning process. The training of the network depends heavily on a large dataset, and for most object detection networks, when the amount of data reaches 100, the performance begins to improve significantly.

3.5 Additional Experiment

In addition to object detection, our task also has good performance in counting. Compared to object detection tasks, counting tasks do not require the complete box selection of objects. However, unlike the relatively sparse distribution of objects in object detection tasks, the objects in counting tasks are often dense and there is a lot of stacking phenomenon. Solving this problem is the core research goal of counting tasks.

3.5.1 Methods

In this section, we will introduce our research on counting tasks. Similar to the object detection task, we do not need to train the object detection task during the counting process. We only need a pre-trained model trained on ImageNet1k [60] to complete the counting without the need for subsequent training.

When counting, our process is divided into four steps3.11. Firstly, we grid the original image and obtain the patch detection score. Then, we use these score results to generate heat maps of different scales. By calculation [149], we can obtain the number of objects in different scale heat maps, and finally calculate the final counting result based on the center coordinates of objects in different scale heat maps.

Grid based and heatmap generation

Just like our method in the object detection process, after obtaining the original image, we need to grid the image, use a classification network to perform score detection on patches, and use the detected results to synthesize a heatmap. Because we do not know the size of the objects in the image, and there are objects of different sizes in the image, it is not possible to use a patch of one size to detect objects of different scales. The smaller the patch partition, the more patches are genernated, and the greater the computational workload. Therefore, we should try to avoid the occurrence of small patches as much as possible.

After obtaining the image, we first perform large-scale gridding on the image and generate corresponding heatmaps. Then, we use score threshold to judge the obtained heatmaps and decide whether to perform more detailed patch partitioning on the image.

After completing the generation of the heatmap, we need to statistically analyze the possible objects included in the obtained heatmaps at different scales. Using the score threshold,

we can determine the score of each patch in the heatmap and count the objects in the current heatmap.

Post-processing based on pooling and filtering

Due to the fact that we count all scales of heatmaps during the counting process, before obtaining the final counting result, we need to judge and merge the number of points on different scales of heatmaps with similar distances based on positional information, in order to obtain the final number of points result. The specific method is as follows3.12:

First, in the peaks calculated by the multiple-scale heatmap, each peak represents the center of a suspected object. Different scale detections may result in missed detections or noise. Then, we resize the multiple-scale heatmap to the same size. Finally, we superimpose each heatmap to obtain 1. Easy-to-detect objects, which have high peaks in multiple scales after superposition; 2. Missed objects, which may be detected at other scales; 3. Noise, which only appears at a few scales and has low peaks in the region, can be eliminated.

3.5.2 Experiments

Datasets

For the counting task, we collected 100 images with multiple cows from the Internet. Each image containes 40 to 130 cows in different postures. We used these 100 images to effectively evaluate the counting part of our algorithm.

Evaluation Metrics

We used Mean Absolute Error (MAE) to evaluate the counting results, and the formula is as follows. Among them, y^p is the predicted count result, and y^g is the real count result:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i^p - y_i^g| \tag{3.5}$$

The smaller the MAE, the better the counting results are.

3.5.3 Main results

Using the above testing methods, we evaluated our counting results.

During the counting process, we compared it with YOLOX [6], Fcos [7], and WSOD [115], and the evaluation results are shown in Figures3.133.14.

Through comparison, we found that when the training set is less than 400 images, our counting results have obvious advantages.

Through the comparison on the above two datasets, we found that, similar to the results of object detection, our method achieved better performance with a small amount of data. As for Fcos [7], which used anchor-free strategy, it performed better than the counting results of weakly supervised [8] and anchor based [6] landmark detection when there was a small amount of data. In counting tasks, weakly supervised object detection generally requires more data for training compared to supervised learning. Unlike the supervised learning and weakly supervised learning methods that surpassed our method when the data volume reached 200, in technical tasks, these methods surpassed our counting accuracy when the data volume reached 400. This is because in our object detection process, it is often difficult to capture the boundary of the target, which leads to lower MIOU accuracy. Compared with the object boundary, our method has better results when detecting the most characteristic central part of the target. Therefore, compared to the detection task, our method performs better in the counting task.

3.6 Conclusions

In this chapter, we proposed a new object detection method for addressing the data scarcity issue in practical animal breeding applications. Our method generated multi-scale gridded heatmaps using a pre-trained classification model and a combination of feature similarities for object detection. This method can effectively integrate the features of different categories, thereby effectively utilising various features learned in the pre-trained model to improve the accuracy of object detection tasks. In addition, we further refined the bounding box by using a similarity segmentation technique (superpixel technique) to improve the detection performance. Based on our comparative experiments, we found that when the number of training data is small, our method outperformed some existing object detection networks. In the future, we plan to test our methods on more datasets to further demonstrate its effectiveness. We also plan to use lightweight classification networks to test its real-time performance.

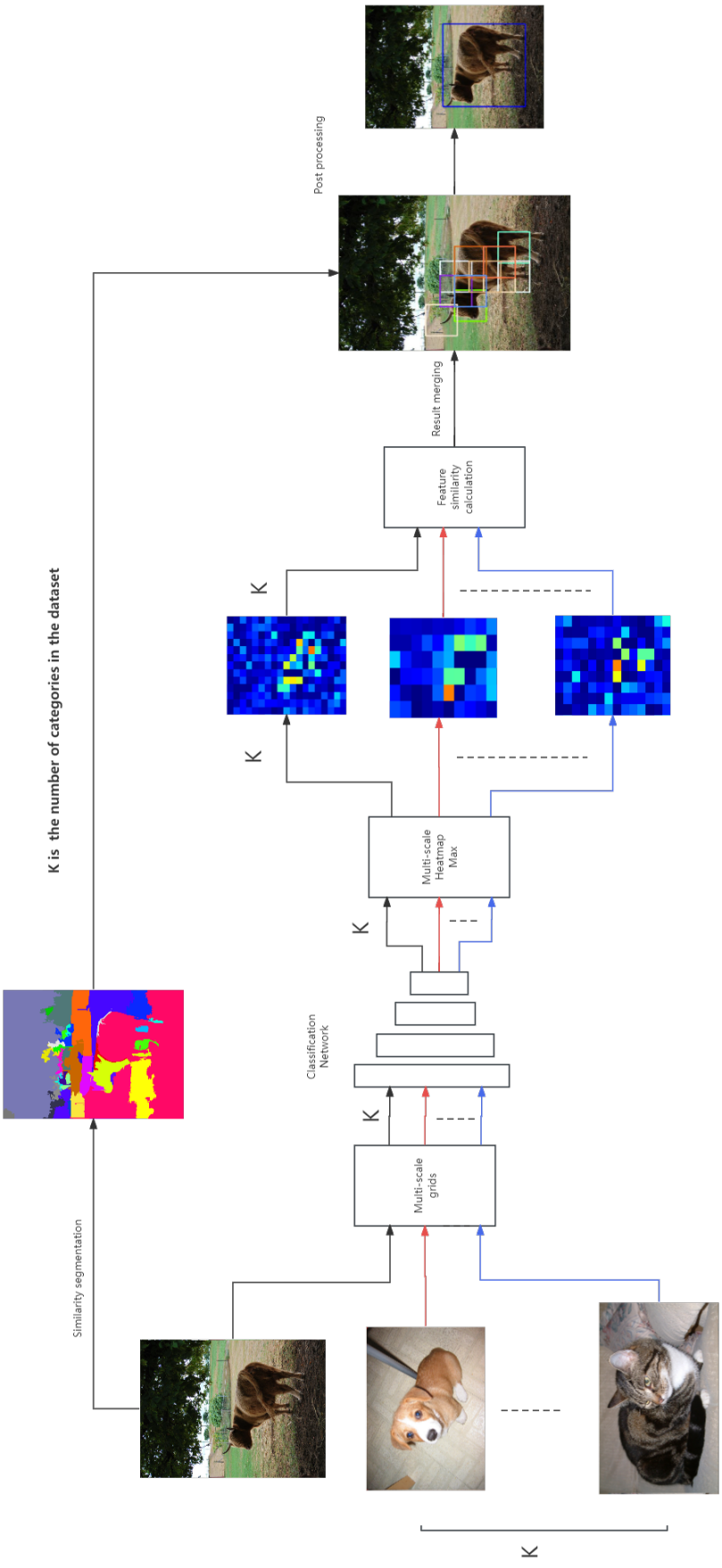

Figure 3.1: Structure of our object detection algorithm

Figure 3.2: Multi-scale gridded heatmaps

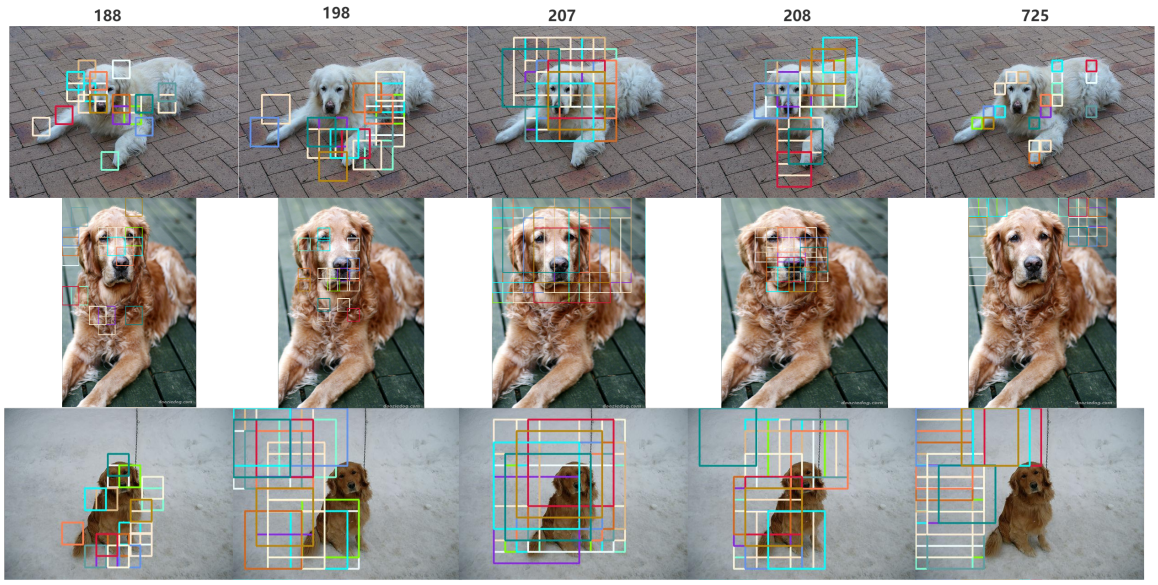

Figure 3.3: Multiple region proposals for object detection

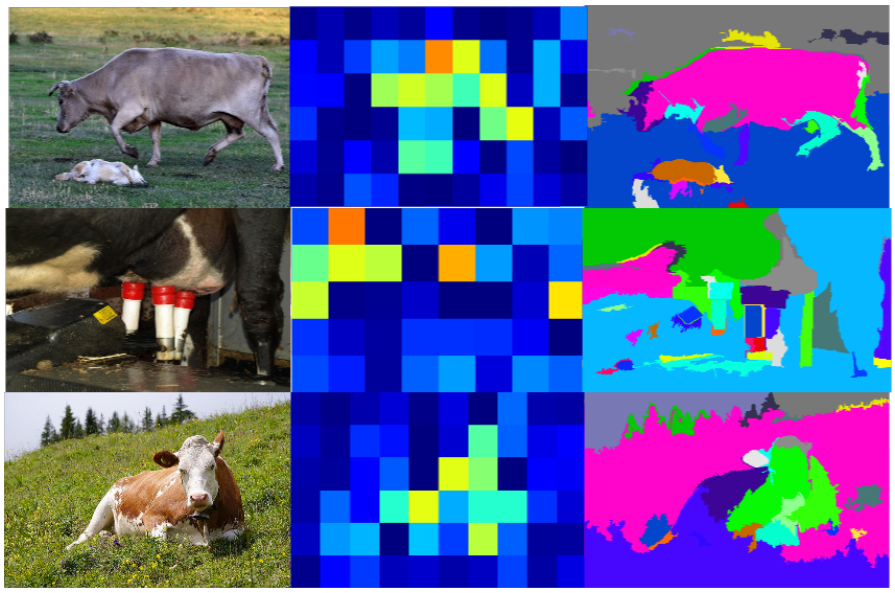

Figure 3.4: Input images, heatmaps, similarity segmentation images from left to right

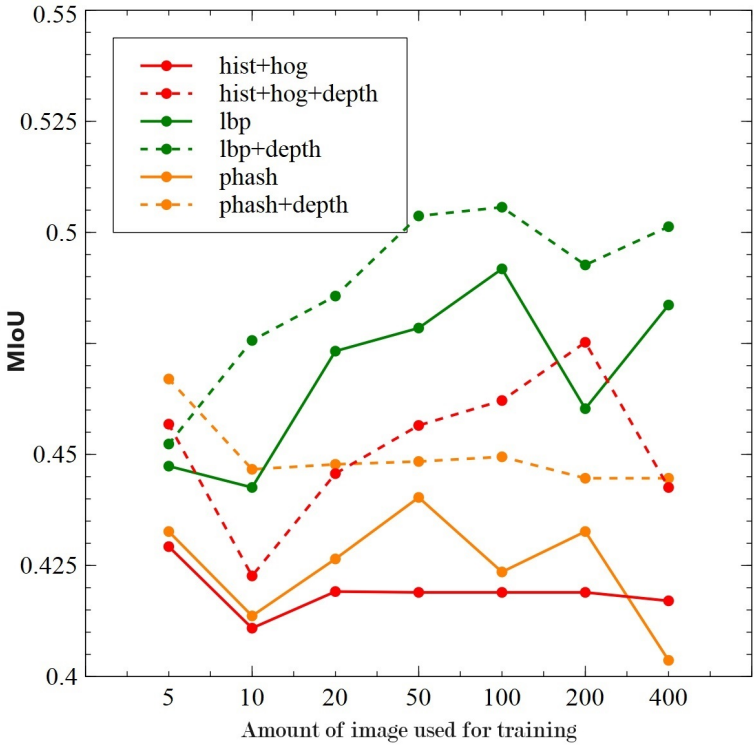

Figure 3.5: Impact of heatmap similarities on object detection results

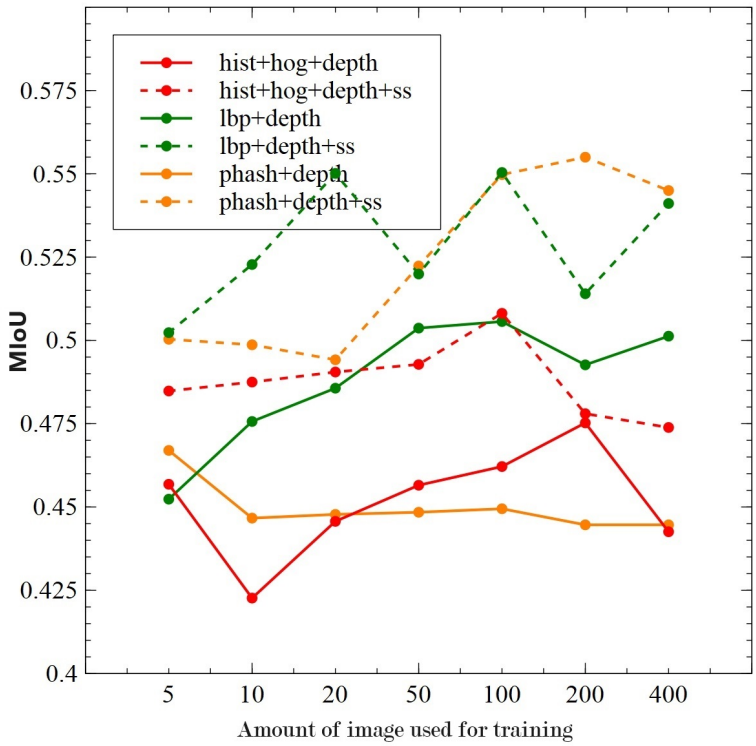

Figure 3.6: Impact of similarity segmentation on object detection results

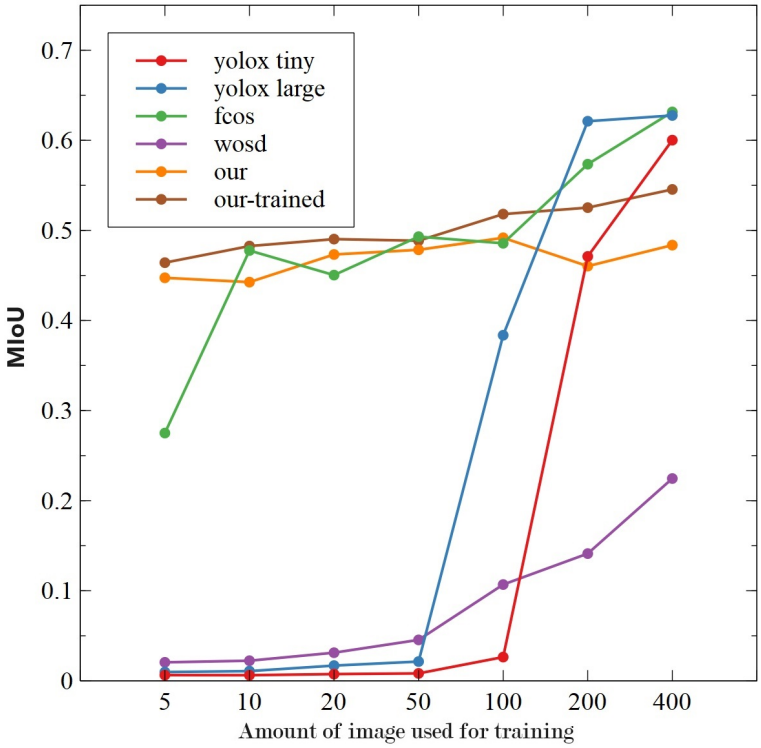

Figure 3.7: Performance of detecting Corgi with different algorithms on our dataset

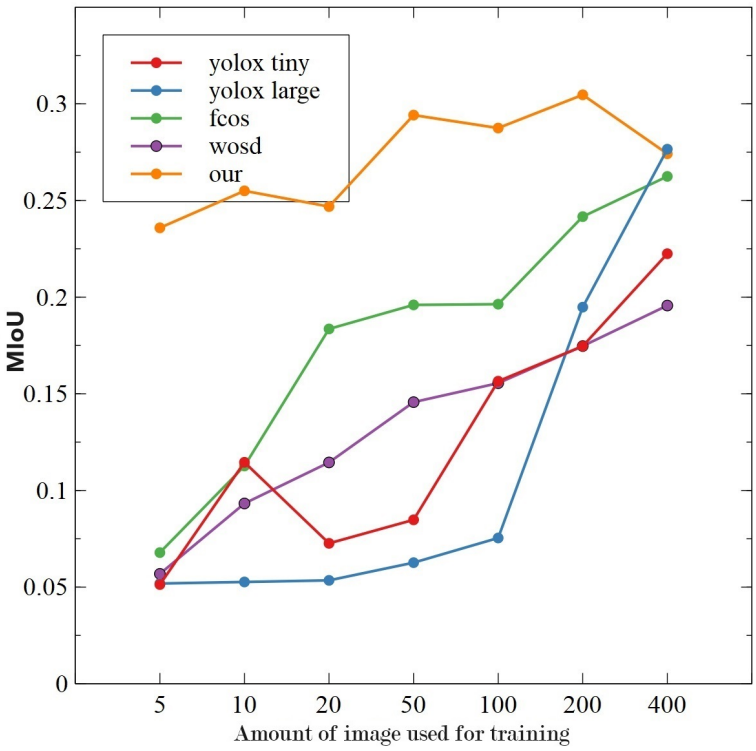

Figure 3.8: Performance of detecting cow with different algorithms on COCO dataset

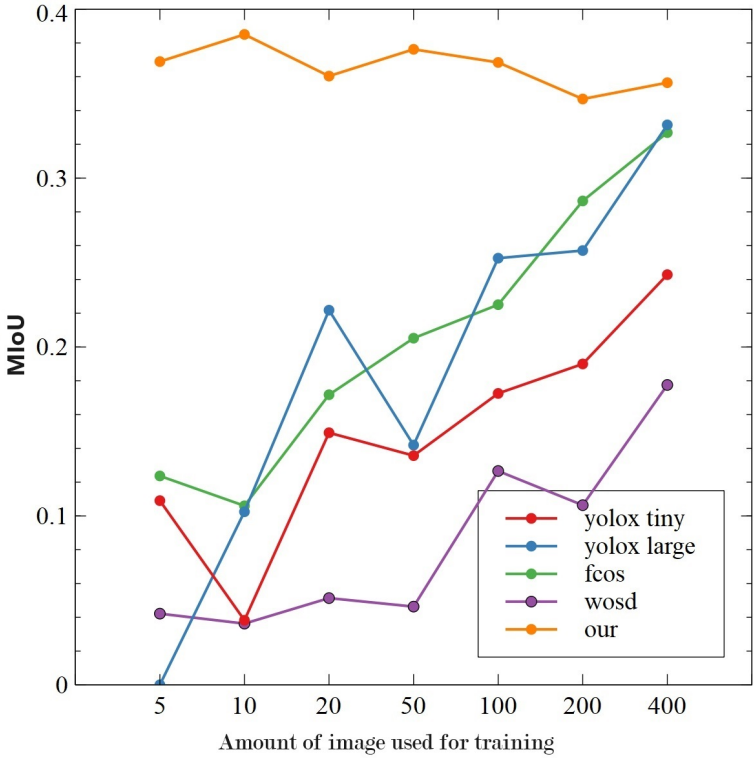

Figure 3.9: Performance of cow with different algorithms on VOC dataset

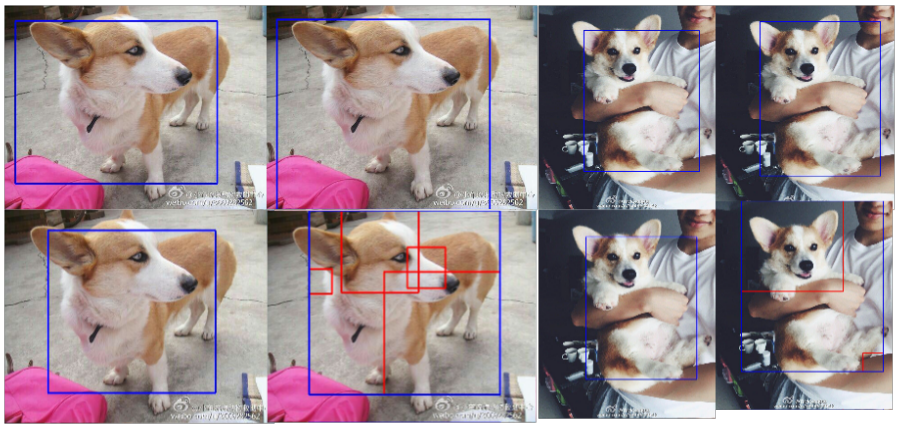

Figure 3.10: Performance of Corgi with different networks in our dataset, the top left image shows Yolox, the top right image shows Focus, the bottom left image shows WSOD, and the bottom right image shows our object detection results. Yolox, Focus, and WSOD were trained on 400 images

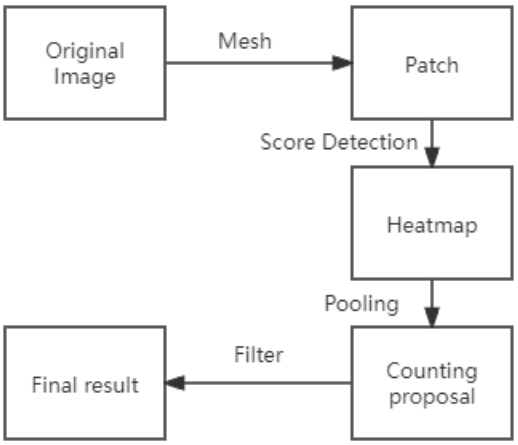

Figure 3.11: Structure of our counting algorithm

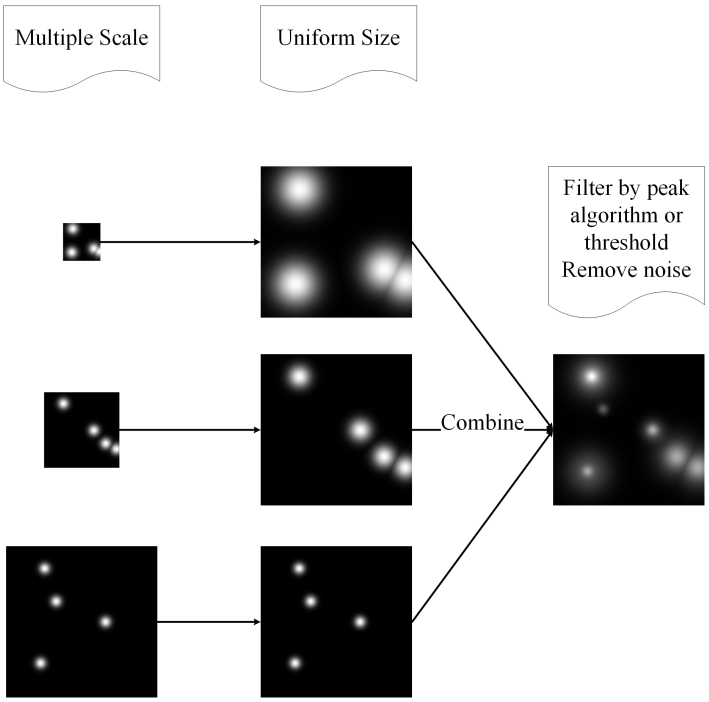

Figure 3.12: Structure of our counting algorithm

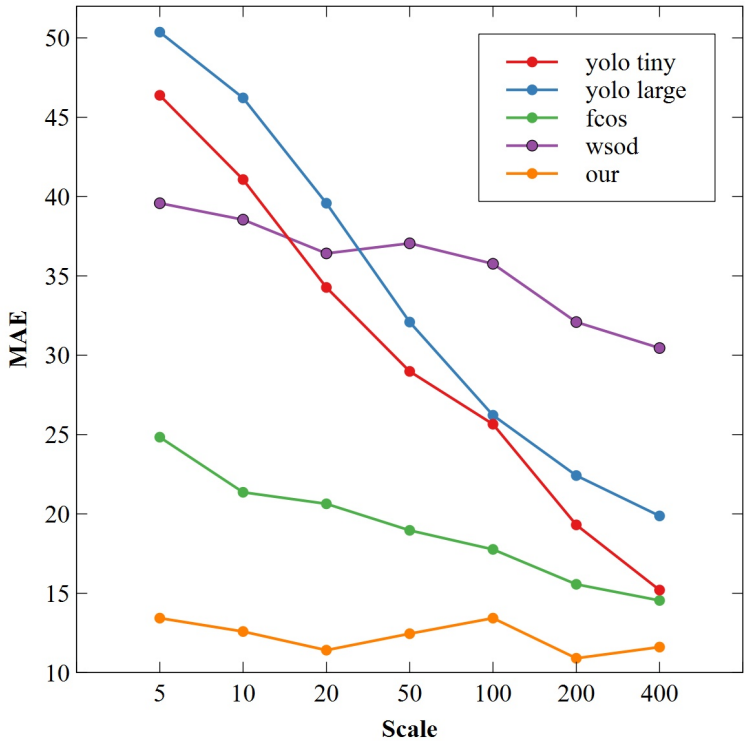

Figure 3.13: Counting performance of different algorithms when COCO is used as training data

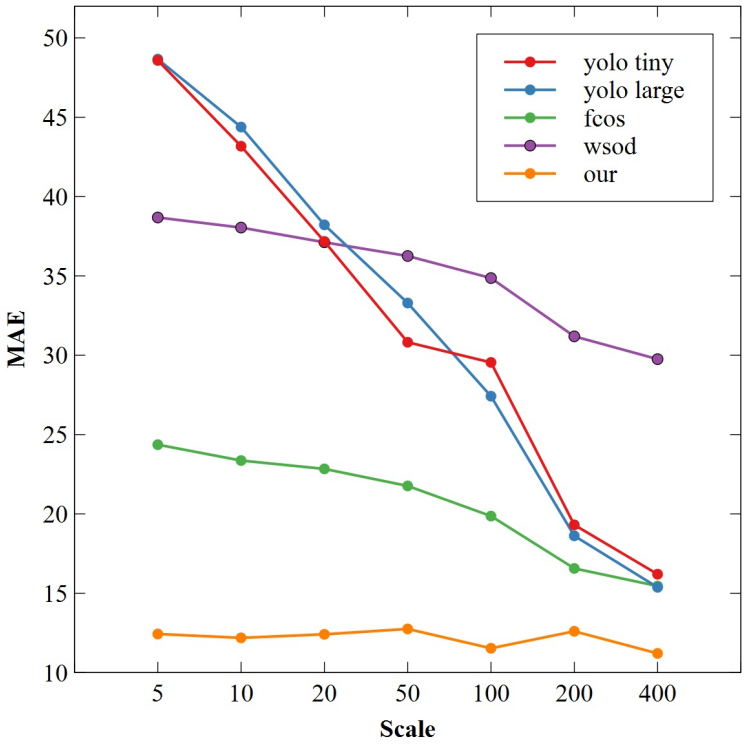

Figure 3.14: Counting performance of different algorithms when VOC is used as training data

Chapter 4

Deep Multi-task Learning for Animal Chest Circumference Estimation from Monocular Images

The applications of deep learning algorithms with images to various scenarios have attracted significant research attention. However, application scenarios in animal breeding managements are still limited. In this chapter we propose a new deep learning framework to estimate the chest circumference of domestic animals from images. This parameter is a key metric for breeding and monitoring the quality of animal in animal husbandry. We design a set of feature extraction methods based on a multi-task learning framework to address the challenging issues in the main estimation task. The multiple tasks in our proposed framework include object segmentation, keypoint estimation, and depth estimation of cow from monocular images. The domain-specific features extracted from these tasks improve upon our main estimation task. In addition, we also attempt to reduce unnecessary computations during the framework design to reduce the cost of subsequent practical implementation of the developed system. Our proposed framework is tested on our own collected dataset to evaluate its performance.

4.1 Introduction

In animal breeding management systems, the administrator usually needs to regularly check the animal to obtain the physiological parameters of animal in time, so that the feeding

strategy could be optimised accordingly. However, the collection of various physiological parameters adds a lot of complex daily workloads to the administrator. The purpose of this chapter is to propose a new deep learning framework with images to simplify the process of animal monitoring and management.

There are many physiological parameters of animal. In this chapter, we only focus on the use of a monocular camera as a low-cost device to be applied for this application to estimate the chest circumference of a dead cow lain on the ground. The workflow in our proposed framework is to capture a monocular RGB image of the cow from camera as input, then generate the intermediate results (depth, keypoints, and segmentation), finally output the regression result (chest circumference). The overall framework is illustrated in Fig. 5.1. The parameter (chest circumference) to be estimated is shown in Fig. 4.2. The intermediate results from our multi-task learning framework include the bounding box (the first image), the keypoints (the second image), the semantic segmentation mask (the third image), and the depth map (the fourth image) as shown in Fig. 4.3. This framework can be implemented by using a monocular camera and a GPU computer in real time, which greatly reduces the waste of human resources in the animal breeding management systems.

Traditional algorithms for object detection [121], object segmentation [46] and key point detection [150] are mostly based on 2D images. If we only analyse 2D images, it is difficult to retrieve the distance between two points in an image. Some 3D modeling methods [151] could consume a lot of computational power, making it difficult to implement the algorithm in real time. In order to balance the computational power and implementation demand, we choose to use the monocular depth estimation from RGB image to capture 3D information. We also estimate the keypoints of animal from segmented region. By fusing the depth and keypoint information, we are able to estimate the chest circumference of animal from the multi-task learning framework. Through our testing and evaluation results from the images we captured in real life, we found our proposed system is effective and performs well in various lightning conditions.

4.2 Related Works

There are some existing estimation methods for object length from images relevant to this work. They are based on object segmentation [46] and keypoint detection [122]. They

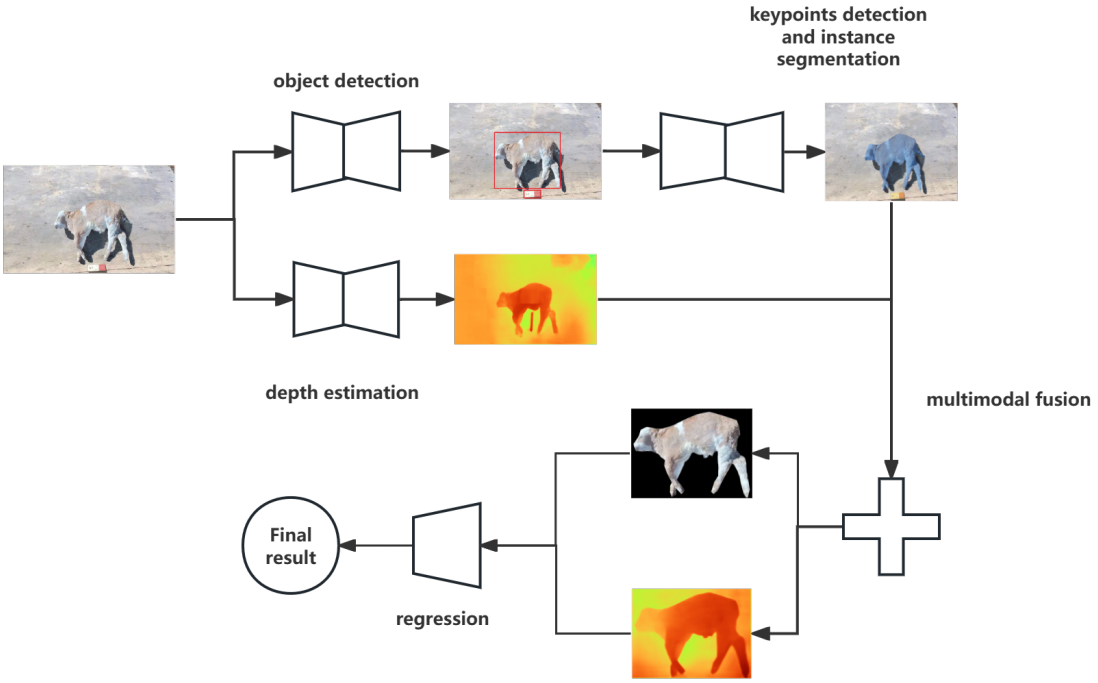

Figure 4.1: Overview of our proposed framework, including object detection, object segmentation, keypoint detection, monocular depth estimation and regression network.

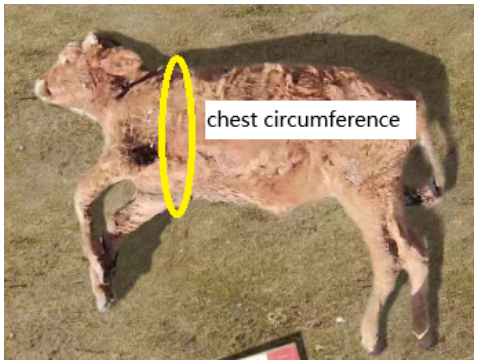

Figure 4.2: The chest circumference of cow

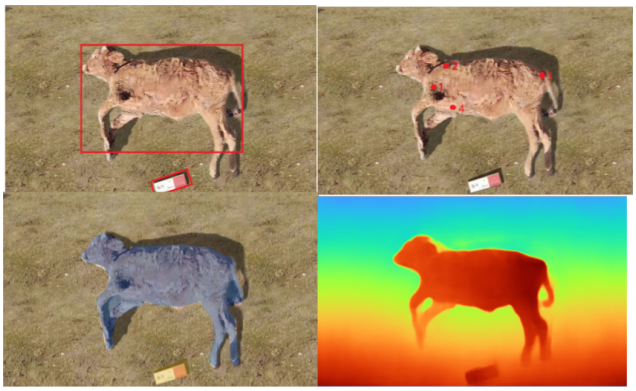

Figure 4.3: The intermediate results from our multi-task learning framework, from left to right: bounding box, keypoints, semantic mask, and depth map generated from our framework.

estimate the length on 2D images, but do not use the depth information. In our work, we use a multi-task learning framework, which combines the information from multiple tasks, including depth estimation, object segmentation, and keypoint estimation, to implement the main parameter estimation task. In the following, we review some existing works on object detection, instance segmentation, key point detection and monocular depth estimation.

Object detection: Object detection is a common task in deep learning. The output of object detection is generally composed of three parts: category label, detection confidence, and object bounding box. They can be roughly divided into two directions: object detection based on anchor frame, and object detection independent of anchor frame. The former proposes anchor boxes or uses traditional computer vision technology to search for potential anchors, matches the proposed anchor with the possible ground truth box, and trains to correct the input proposal box to complete the prediction of the bounding box [152] [6]. The latter [122] [7] is mainly divided into two categories: dense prediction based and key point estimation based [153]. Our object detection network is based on the Yolo network architecture [152] with a reduced number of layers for network efficiency.

Instance segmentation: The instance segmentation [104] task aims to obtain the category information of specified objects in image. They work in end-to-end fashion, or perform the detection first and then segmentation [154]. Their outputs include mask, object label and confidence. They can be used to eliminate the influence of background on feature extraction areas [122]. In recent years, there are also some semi-supervised [40] and unsupervised [39]

based results. The current mainstream object segmentation models can be roughly divided into three categories:

- End to end: it directly outputs the segmentation result. By using a regression network and the ground truth information, a segmentation network is directly trained.
- Bottom up: The idea is to perform the semantic segmentation at pixel level first, and then different instances are distinguished by clustering, metric learning or other means.
- Top down: The idea is to estimate the bounding box of the instance first, and then perform the semantic segmentation inside the box. The representative is the well-known Mask-RCNN [46].

Keypoint detection: In recent years, keypoint detection has been widely used in human pose estimation [150] [123] and face recognition tasks [155]. It outputs a sequence of 2D keypoints. They are also divided into two ideas: top down and bottom up. The former is to detect the object first, and then carry out the keypoint regression [156]. On the contrary, the latter is to regress multiple groups of keypoints first, and then group them together [150]. At present, most keypoint detection tasks are based on heatmap [122], and the improvement is observed by optimising the information loss in [123]. Our network uses the same feature extraction layers for both segmentation and keypoint selection, which can share the representation and help each other.

Monocular depth estimation: At present, most monocular depth estimation methods are based on the Structure from Motion (SfM) [87] framework, which relates the depth estimation with camera pose estimation together. In many cases [100] [84] [85], monocular depth estimations are trained on the Kitti dataset [81] and CityScape dataset [82]. The scale problem is hard to be solved. The accuracy improvement in dynamic scenes has been the research focus for a period of time [95] [91]. Our depth is based on the pre-trained network Midas [100] and an alignment method, which is able to produce an aligned estimated depth map.

Multi-Task Learning: Recently, multi-task joint learning has been increasingly applied in more scenarios, such as optimising network structures and incorporating multi-modal information to optimise the results of object pose estimation in space [157], using multi-task

joint learning for text correction in the field of text recognition [158], and applying multi-task joint learning in the field of genetics [159]. Inspired by these applications, our proposed framework is a multi-task joint learning paradigm.

4.3 Methods

In this section, we will describe our algorithm framework in detail. We use a monocular RGB image containing a dead cow lain on the ground as input. Our final goal is to estimate the chest circumference of the cow in the image. We employ a multi-task learning framework, which includes the use of results from multiple estimation tasks, such as depth map, keypoints, and object mask of the cow, to regress the final parameter. The framework is designed to facilitate the practical application, and focus on the design of light-weighted network models.

Fig. 5.1 shows the overview of our proposed framework. The input is a monocular image. It uses an object detection network to detect the object and produce a bounding box from the input image. Then it uses a network to perform the keypoint detection and instance segmentation. At the same time, it uses a monocular depth network to estimate the depth map from the input image. The keypoints, object mask, and depth map are then used as input to the final regression network. The regression network outputs the chest circumference parameter.

4.3.1 Object detection

Our object detection network is to estimate the bounding box of a cow from input image. It can effectively remove the background, and also filter out some images that do not meet the requirements. In the subsequent process, the cow to be detected needs to appear completely and clearly in the image. By filtering out poor images, the system only needs to detect the images that meet the requirements.

Our object detection network is based on the Yolo network architecture [152]. As shown in Fig.4.4, we have modified the multi-scale output of Feature Pyramid Network (FPN) [54] by reducing the number of high-resolution layers in FPN (please see the deleted section). This effectively reduces the number of network parameters, ensures the effective output of large target results, and reduces the impact of incomplete small objects on detection results.

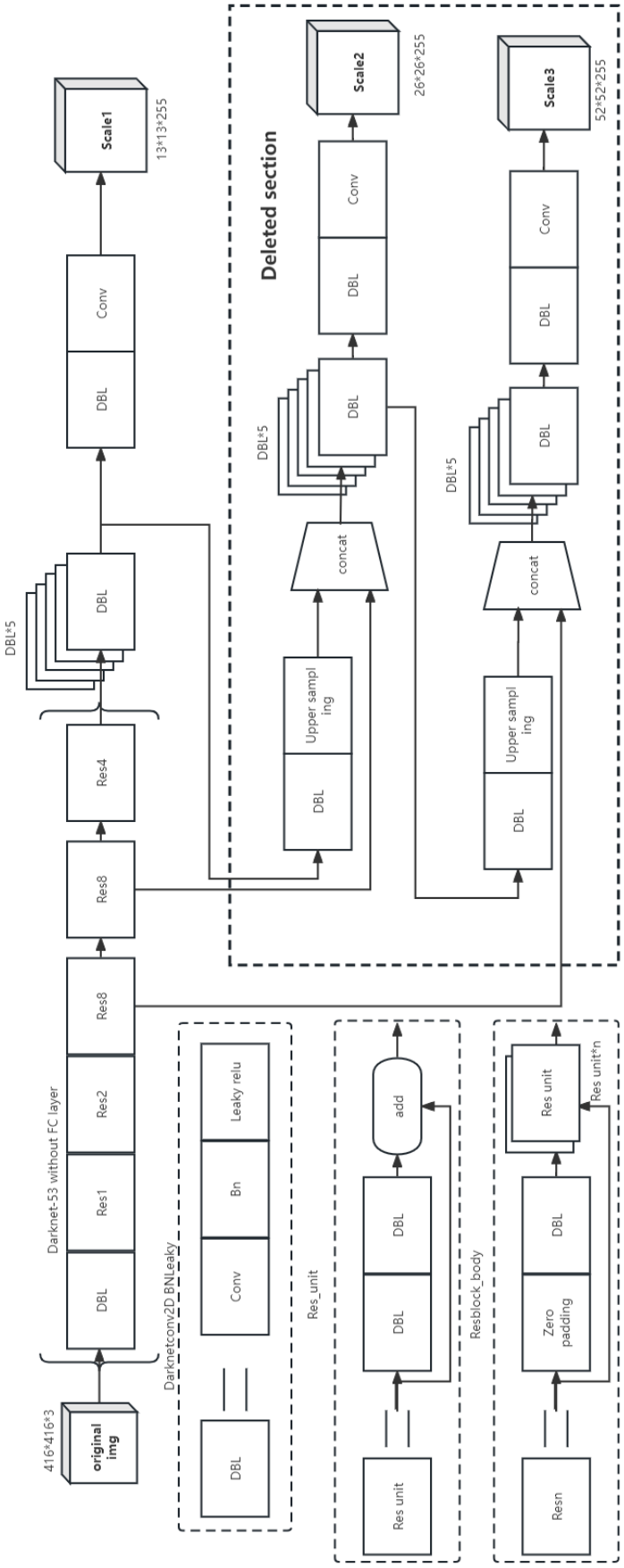

Figure 4.4: The object detection network structure

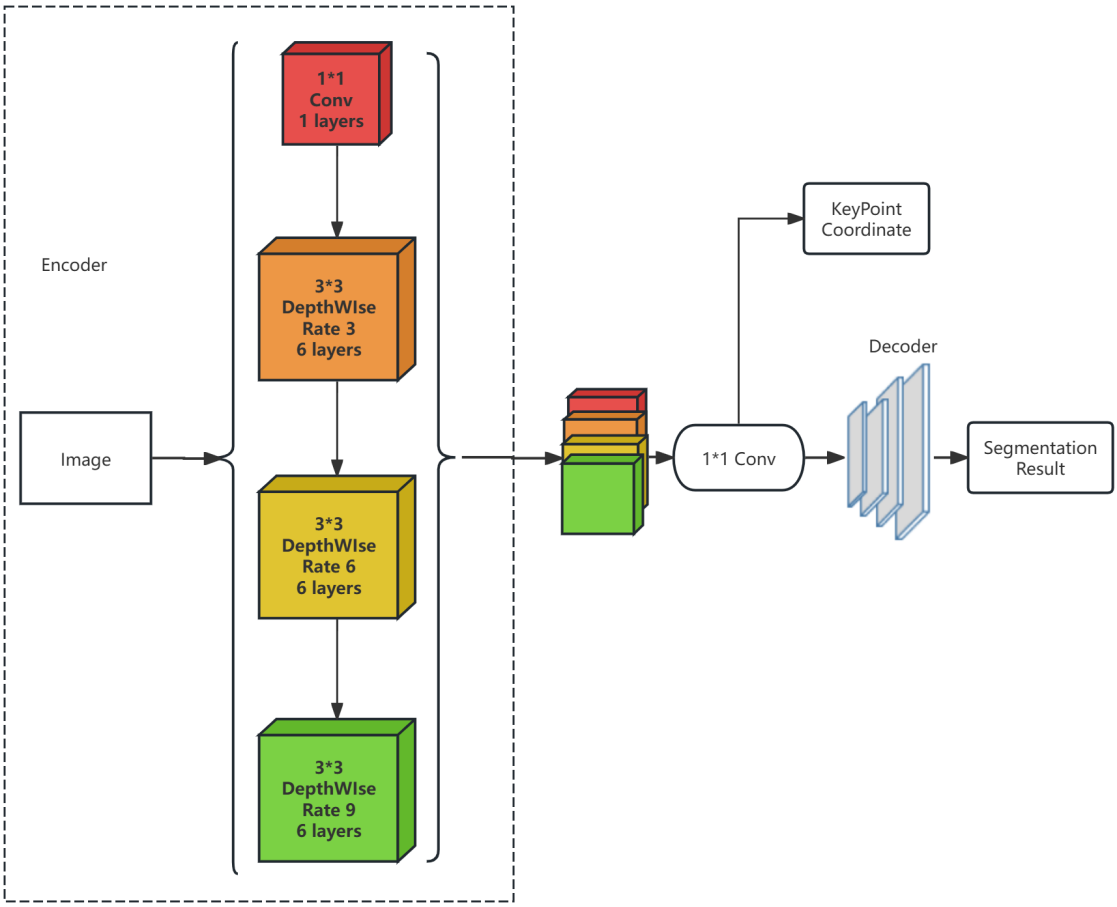

Figure 4.5: The object segmentation and keypoint detection network structure. There are 19 layers in total in the feature extraction part

4.3.2 Keypoint detection and object segmentation

The keypoints we select are distributed around the edge of the segmented object (see the second image in Fig. 4.3). As they are located around the outline of cow body, they provide important information for the parameter to be estimated. The keypoint detection result is to ensure that the cow in the image has an unified position and posture for next step processing.

The keypoint selection provides an opportunity to share the feature extraction parts of the network for the keypoint detection and object segmentation tasks as both tasks share similar features. This selection also makes the features of the two tasks interrelated, so that the two tasks can optimise each other in the feature extraction process. This could potentially improve the performance for both tasks, and also simplifies the network complexity.

Our object segmentation and keypoint detection network is shown in Fig.4.5. It is built up on the top of the Deeplab-V3 [154]. We inherit the feature extraction part of Deeplab-V3 [154]. After the last convolution of feature extraction is completed in the network, the convolution result is divided into two branches. One branch produces the object segmentation result after performing the deconvolution operations in the decoder part. The other produces the keypoint detection result after performing a linear operation to stretch the feature extraction results into a one-dimensional tensor and passing a fully connected network.

Our object segmentation task has the following features:

- Single object: the object we need to segment is a single cow object and a reference object (cigarette box). The reference object is used for comparing the distances between keypoints with the length of the reference object (cigarette box) in the image. This is important for dealing with the scaling problem. See the regression subsection below for more details.

- The proportion of the object in the image is large: in general, the proportion of cow in the image should be more than half in the application scenario.

We choose an end-to-end method for the tasks mentioned above. This can also reduce the computational power required in the process.

4.3.3 Aligned depth and scale estimation

Our depth estimation network uses the pre-trained network Midas [100] to estimate the depth for each pixel of input image. The depth from this network is the estimated distance $\hat{d}_i$ between a point i of the cow and the image plane. There are two problems to be solved before it can be used to estimate the chest circumference:

1. When the angle between the cigarette box plane and the image plane is small, the scales in the x and y directions are basically the same ($s_x = s_y$). But the scale in the z direction (s_z) is different with the x scale. We have to estimate them separately. When the angle between the cigarette box plane and the image plane is not small, we need to calculate each scale individually.

2. The ground plane is not parallel to the image plane. We have to estimate the ground plane and align the ground plane with the image plane. A new aligned depth estimate is required for the main parameter estimation.

These two problems are solved by using the reference object (cigarette box) in image. Given the known lengths of the cigarette box, we can retrieve the scales. By detecting the corner 3D coordinates of the cigarette box in image, we can retrieve the ground plane.

To solve problem 1, we use the results of object segmentation and depth estimation. We can obtain the estimated 3D coordinates of four corners of the cigarette box in image. When the scales in x and y directions are very different, we can assume the scales satisfy the linear relationship $s_x = js_y = ks_z$. We can obtain s_x from the estimated coordinates of four corners and the ground truth lengths of the cigarette box. Further,we can obtain k and j from the ratio of ground truth lengths of long and short sides of the cigarette box.

To solve problem 2, we assume that some areas in the image are the ground. We need to estimate the ground plane. The ground plane equation is $ax+by+cz+d=0$. By detecting sufficient points in the image that are the ground and using the least squares method, we can obtain the equation parameters a,b,c,d. The normal vector of the ground $\vec{n_g}$ is (a,b,c).

From the estimated 3D coordinates of four corners of the cigarette box in image, we can obtain the direction vector l of any two points (x_1,y_1,z_1),(x_2,y_2,z_2). The angle between the

straight line and the image plane is:

$$\cos\theta = \frac{\vec{n_i} \cdot \vec{l}}{|\vec{n_i}| \left|\vec{l}\right|} \tag{4.1}$$

where $\vec{n_i}$ is the normal vector of the image plane $(0,0,1)$, and the straight line is the segmentation result of the cigarette box in the ground plane.

Given the ground truth length L between two selected corners of the cigarette box, the depth difference between the two points is $d_{diff} = L\sin\theta$. In this way we can obtain the depth differences between any two points of the cigarette box. Then we can compute a set of depths d_i for selected points in the cigarette box using d_{diff}. Their corresponding estimated depths from the network Midas are $\hat{d_i}, i = 1,...,n$. Using a set of pairs $d_i, \hat{d_i}$, a least square method is used to compute the parameters D_{scale} and D_{shift}:

$$D_{scale} = \frac{\sum_{i=0}^{n} d_i \hat{d_i} - \frac{\sum_{i=0}^{n} d_i \sum_{i=0}^{n} \hat{d_i}}{n}}{\sum_{i=0}^{n} d_i^2 - \frac{(\sum_{i=0}^{n} d_i)^2}{n}} \tag{4.2}$$

$$D_{shift} = \frac{\sum_{i=0}^{n} \hat{d_i} - D_{scale} \sum_{i=0}^{n} d_i}{n} \tag{4.3}$$

Finally the aligned estimated depth $\bar{d_i}$ is:

$$\bar{d_i} = \hat{d_i} D_{scale} + D_{shift} \tag{4.4}$$

4.3.4 Regression network

Our multi-task learning framework generates the object mask, depth map and keypoints, which are fed into the regression network to estimate the parameter: chest circumference. Before they are fed to the regression network, they have to be normalised to have a unified size. As we need to preserve the aspect ratio of the object, we use the padding method to normalise the size.

As the distances between each image and the camera are different, we have to use the reference object (cigarette box) with known lengths to find the scale for our regression task. We choose the ratio of diagonal lengths between the bounding box and the reference box as the scale to calculate the parameter.

The loss function for our regression task is designed by using a Mean Absolute Percentage Error (MAPE).

$$loss = \tan \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \tag{4.5}$$

where n is the number of images, y_i is the ground truth, and $\hat{y}_i$ is the estimated parameter for image i. The loss value is in the range of $(0 < \frac{1}{n}\sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right| < \frac{\pi}{2})$.

The regression network is shown in Fig.4.6. We choose the A0 version of RepVgg [48], and use 3×3 convolution cores in the regression network. This design greatly improves the optimisation efficiency of CUDA for training speed. At the same time, the design of our residual blocks is similar to residual networks, which avoids the problems of gradient disappearance and gradient explosion in the training process. We add an attention network CBAM [53] in the input and output parts of the network, which enables the regression network to focus more on the effective region.

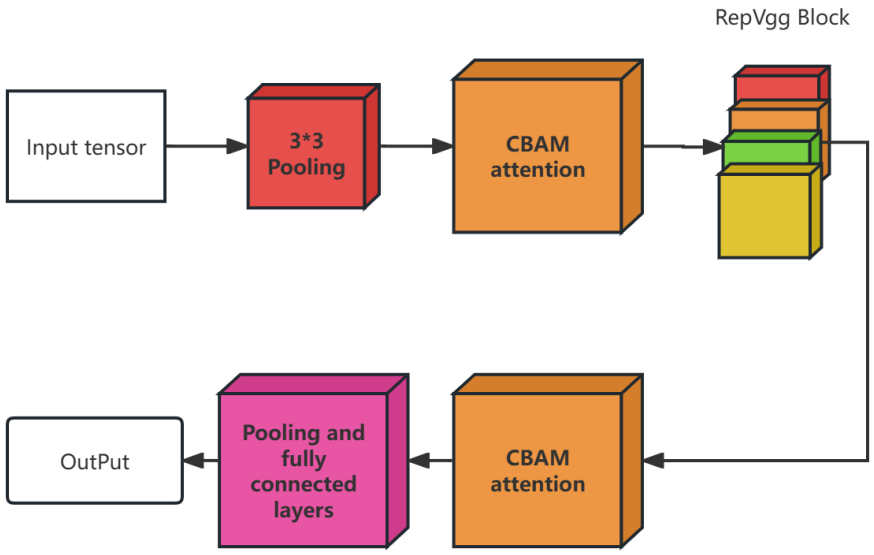

Figure 4.6: The structure of our regression network

4.4 Experiment Results

In this section, we will present the details of our experiments and results. We use the GeForce GTX1080Ti graphic card to train and test various parts of the framework. All of them are

implemented and tested on PyTorch.

4.4.1 Dataset

Our dataset is collected from real scenes where all the images contain a cow and a cigarette box lain on the soil and cement ground. The camera used is a short focus camera with a focal length of 3 mm. More than 200 cows are used for image collection, each of them is recorded with a video clip. From all the video clips, we obtain 3000 training images and 300 test images. We manually label the training set and the test set with the ground truth segmentation mask and keypoint coordinates. And the bounding box is generated using the segmentation results we labeled, without the need for additional annotations. The deep estimation network is not trained in our framework and there is no need for labeling.

4.4.2 Evaluation Metrics

Our estimated parameter $\hat{y}_i$ is the chest circumference of cow in image i. The corresponding ground truth is y_i. The average error represents the average percentage of error between the estimated value $\hat{y}_i$ and the true value y_i of all test samples.

$$AE = \sum_{i=1}^{n} \frac{\hat{y}_i}{y_i} \tag{4.6}$$

R2 score shows how well our regression model predicts the chest circumference of cow from observed images. It reflects the proportion of all variations of dependent variable y_i that can be explained through the regression model. It is computed as below

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} \hat{y}_i \tag{4.7}$$

$$R^2 = \frac{SS_{reg}}{SS_{tot}} = \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\hat{y}_i - \bar{y})^2} \tag{4.8}$$

If the R2 score (R^2) is close to 0, the model cannot estimate the parameter at all. If it is 1, the model prediction is perfect.

4.4.3 Training

Our keypoint detection and object segmentation network is developed based on the Deeplab-V3 [154]. The input of the segmentation and keypoint network is a colour image with a

side length of 300 and 3 channels. The output tensor shape of the target segmentation part is (1,300,300), and the tensor shape of the keypoint detection part is (4,2). The height and width of the input tensor of the regression network for the final output (chest circumference) are 224 pixels. The tensor shape of the final output of the network is 1, and the result of the network output will be multiplied by the scale obtained during the preprocessing of the segmented network to obtain the final output in centimetres. During the training process, we used AdamW [160] as the optimiser. In order to perform the effective data augmentation while preserving the effective information in the image, we only used three data augmentation methods: random rotation of no more than 45 degrees left and right, random flipping of image up, down, left and right, and adding a small amount of noise during the data augmentation process. We train this network by using our training dataset with labels.

Table 4.1 displays the testing results for the segmentation network. The object size in the table represents different object sizes in images. Due to the fact that the objects we detected are usually large, we only count the objects with a length and width greater than 32 pixels. When the detection object is between 32 and 96 pixels in length and width, the average precision and recall rate can reach 80%. When the detection object is over 96 pixels in length and width, the average precision can reach 87.8%, and the average recall rate can reach 90.6%.

At the same time, we separately compute the MIOU (Mean Intersection over Union) of the segmentation network for different objects. In order to verify the performance of the segmentation network in the boundary part, we include the testing on the Boundary IOU [161] when testing the segmentation network. Table 4.2 shows the testing results. It can be seen that all the values are above 94%, which indicates the network performs well in our dataset.

In terms of keypoint detection, our average OKS (Object Keypoint Similarity) is 0.97.

Our monocular depth estimation network is developed based on the MidasV2 [100]. We do not need to train the network, only apply the alignment scale to the output of monocular depth estimation.

After training the above two networks individually, the regression network shown in Fig.5.1 is trained on our dataset with the loss function in (4.5). The training process consists of 50 epochs. During the training process, we select AdamW [160] as the optimiser to

Table 4.1: Segmentation results for different object sizes

Indicator Name	IoU range	Target size	Value
Average Precision	0.50:0.95	all	0.878
Average Precision	0.50	all	1
Average Precision	0.75	all	1
Average Precision	0.50:0.95	32:96	0.800
Average Precision	0.50:0.95	96	0.878
Average Recall	0.50:0.95	all	0.906
Average Recall	0.50:0.95	32:96	0.800
Average Recall	0.50:0.95	96	0.906

Table 4.2: Segmentation results for different objects

Indicator Name	Target	Value
MIoU	Cow	0.984471
MIoU	Cigarette	0.978546
MIoU	Macro	0.981509
MIoU	Micro	0.985037
Boundary IoU	Cow(15 Pixels)	0.946831
Boundary IoU	Cigarette(5 Pixels)	0.953415
Boundary IoU	Macro	0.950123
Boundary IoU	Micro	0.949210

achieve rapid convergence of model parameters.

4.4.4 Main results

The main results are shown in Table 4.3. It shows the testing results for different network structures used in the regression network (see Fig. 4.6), including Res18, Res50, Res101, Ghost-ResNet-56 [86], FasterNet-T0 [162], RepVggB2 [48], RepVggB3 [48], and RepVggB2 +CBAM. The second and third columns show the network parameters and flops for different regression network structures.

The fourth column shows the percentage of the test samples whose estimation error is less than 10%. It can be seen that when RepVggB3 [48] is used, there are 47.72% of the test samples with an estimation error of less than 10%.

The fifth column of the table is the average error which represents the average percentage of error between the estimated value and the true value of all test samples. When using RepVggB2+CBAM as the regression backbone network, we obtain a minimum average error of 16.28%, which is better than the RepVggB2 without the attention module CBAM.

Table 4.3: Evaluation results on own dataset

Structure	Parameters	Flops (GFlops)	Sample percentage ($< 10\%$)	Average Error (AE)	R2 score
Vgg19	143667240	19.67	0.4696	0.1726	0.9566
Res18	11689512	1.82	45.17%	18.65%	0.9256
Res50	25557032	4.12	43.54%	19.33%	0.9034
Res101	44549160	7.84	39.54%	21.50%	0.7956
Ghost-ResNet-56 [86]	4342924	0.63	32.32%	20.28%	0.7624
FasterNet-T0 [162]	3942924	0.34	30.43%	20.14%	0.7524
RepVggB2 [48]	89023016	20.47	43.35%	16.66%	0.8234
RepVggB3 [48]	123085928	29.18	47.72%	18.30%	0.8527
RepVggB2+CBAM	89842924	20.47	44.32%	16.28%	0.8934

The last column is the R2 score, in which Res18 performs the best with a value of 0.9256. This shows that the network we designed can fit the data efficiently in this task.

A number of samples are shown with their heatmaps (see Fig. 5.8). We can see that our algorithm has successfully noticed the effective regions in images, such as boundary regions. This demonstrates our network model is effective for the estimation task. This improvement can be attributed to two factors. First, the addition of mask effectively eliminates the interference of background information, making the subsequent feature extraction network more focused. Second, the addition of CBAM [53] attention in the regression network further improves the network's ability to focus on key areas. This is because CBAM [53] is composed of spatial attention and channel attention, with spatial attention improving the network's sensitivity to important regions.

We also test the impact of the depth estimation network on the estimation results. Fig. 4.10 shows the estimated chest circumferences with and without the depth estimating network. The abscissa is the ground truth chest circumference. The vertical axis is the error between the estimated value and the true value. By comparing the results in these two figures, we find that the addition of our monocular depth estimation network can significantly improve the accuracy of the system.

This is because the chest circumference itself needs to be measured in three dimensions and the image itself does not have enough stereo information, so it is difficult to directly estimate the chest circumference accurately. In our method, due to the introduction of the prior knowledge accumulated during the depth estimation model training process, the stereo information of the image is enriched, allowing for better results compared to methods that rely solely on the image's information to estimate the chest circumference. In addition, as

shown in the figure, the estimation result of small chest circumference has greatly improved compared to methods using image information alone for estimation. This is because it is difficult to control the photographer's shooting techniques when photographing cows with small chest circumference , and the instability of the distance-angle relationship between the shooting equipment and the cow will cause the size and shape of the cow in the image to change. These changes are difficult to capture directly in the image, and the addition of the depth estimation calibration method has greatly alleviated the impact of this situation, significantly improving the accuracy of small chest circumference estimation.

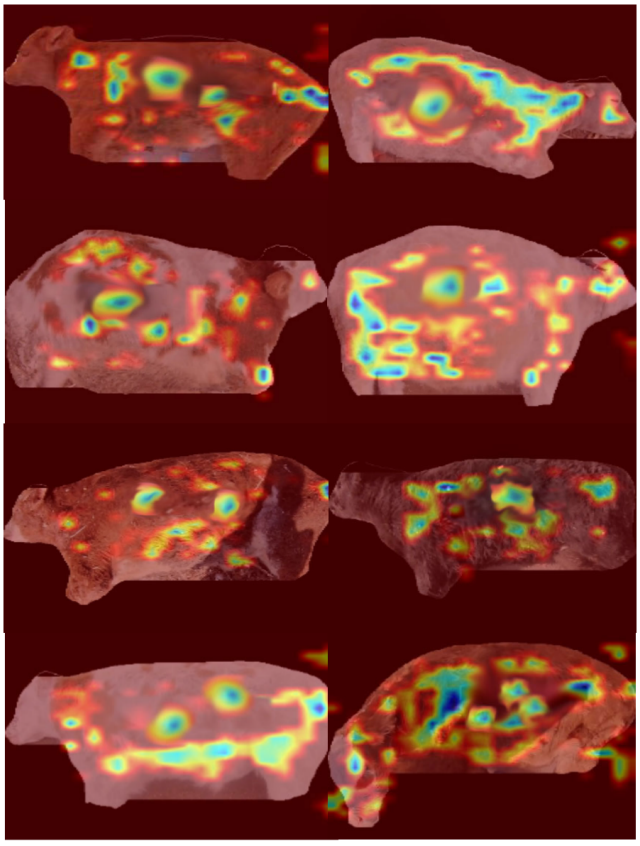

Figure 4.7: Heatmaps for our testing results

4.4.5 Ablation experiments

To demonstrate the performance of our algorithm, we conduct a series of ablation experiments. We conduct ablation experiments with different regression network structures and different input channels.

Fig. 4.8 shows the loss values during the training for the network structures: Res18, Res50, Res101, RepVggB2, and RepVggB3. During the 40 training epochs, we find the RepVggB2, RepVggB3, and RepVggB2+CBAM perform better than Res18, 50 and 101 in terms of convergent rate and loss value. This is mainly because the RepVgg block has more branches compared to the residual block in ResNet.

Fig. 4.9 shows the loss values during the training for different inputs: 1 channel, 2 channels, and 4 channels. We find that the training performance for 4 channels performs better than other cases. When the input is 1 channel, we only use the depth map as the input. When the input is 2 channels, we use the depth map and mask as input. When the input is 4 channels, we use the depth map and the original RGB image as input.

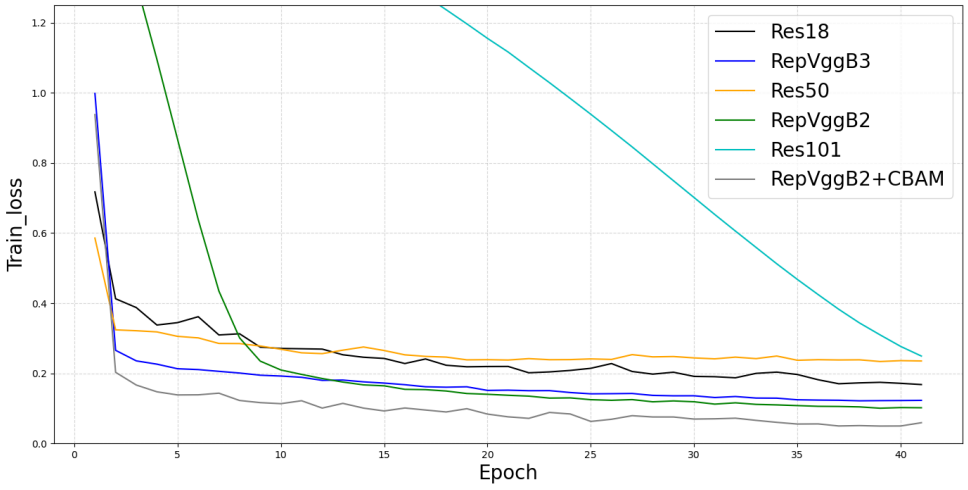

Figure 4.8: The training performance of different network structures

4.5 Conclusion

In this work, we proposed a deep learning framework to estimate the chest circumference of cow from monocular images. This framework is based on the multi-task learning scheme, which incorporates the networks for monocular depth estimation, keypoint detection, object segmentation, and object detection. By fusing the results from multiple tasks, we can regress the chest circumference of cow from monocular images. We also made the contributions to the depth estimation network, and keypoint detection and object segmentation network in order for them to be used in our framework. We collected our own dataset for training and

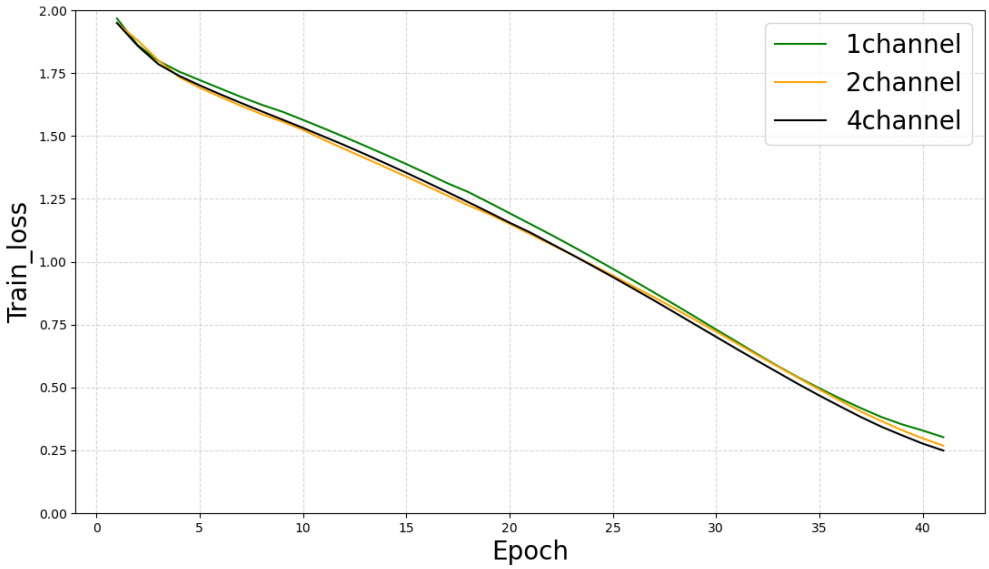

Figure 4.9: The training performance of different input channels: 1 channel for depth map input, 2 channels for depth map and mask inputs, and 4 channels for depth map and the original RGB image inputs

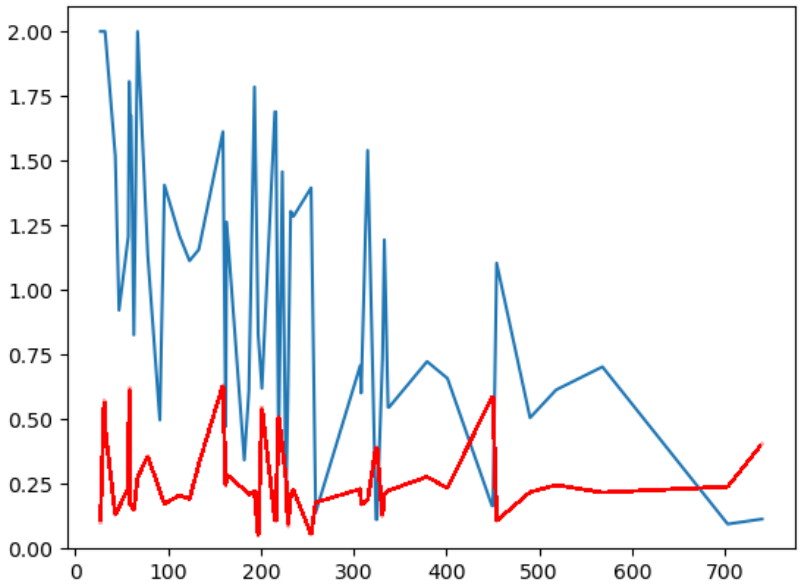

Figure 4.10: The blue one is the estimation results without the depth network and the red one is the estimation results with the depth network. The abscissa is the ground truth chest circumference. The vertical axis is the error between the estimated value and the true value.

testing our models. The evaluation results show our framework is effective and provides a practical solution to the parameter estimation task.

In the future, we would like to extend our framework to include stereo camera images, which could provide more accurate information for the depth estimation. We would like to estimate other parameters, such as the body length or age of animals to enhance the rapid modeling and digitization of animal. The generalisation of the proposed network to other animals, such as pigs, is an essential work in the next step.

Chapter 5

Multi-task Learning for Animal Face Recognition with Spatio-temporal 3D Convolution Network

With the emergence of 3D convolution over spatial and temporal domains, the performance of many visual tasks have been greatly improved. In this work, we use 3D convolution to extract spatial and temporal features from consecutive images for face recognition tasks, particularly for farm animal face recognition tasks in breeding management applications. We also apply a multi-task learning approach to the face recognition task by adding depth estimation and object segmentation networks into the system. By the use of this multi-task learning approach, the training time could be reduced, and the dependence of complex tasks on labeled data is reduced. We contribute a multi-scale 3D convolution network to extract spatio-temporal features from the sub-tasks, depth and mask, for the animal face recognition task. In addition, we also develop a method for generating an 3D point cloud for each detected face, making it possible to register a limited number of images from different angles. With the dataset collected by ourselves, and also public datasets on human faces, we test our network and the experiment result shows our proposed network outperforms some of the state of the art networks.

5.1 Introduction

The application of end-to-end deep learning often requires a large amount of labeled training data as a basis. In some scenarios where it is difficult to obtain sufficient training data, the application of deep learning is often based on multi-task learning [163] This chapter proposes a animal face recognition algorithm based on multi-task learning where the amount of labeled data is limited. It employs two sub-tasks: depth estimation and object segmentation, and one 3D convolution network to generate an animal face recognition result. The goal is to establish a learning system which requires a small amount of data for training.

Most face recognition methods are based on individual images. We propose to use consecutive images to implement our face recognition method as this will provide not only spatial features but also temporal features. In order to effectively extract and combine different features in multi-task learning, a "3D multi-scale network" is proposed, which uses 3D convolution to extract spatio-temporal features from consecutive images to increase the accuracy of the face recognition results.

In addition, we also use a key point detection network to detect the key points of an animal face in image. The consecutive input images, estimated key points, and estimated monocular depth are combined together to generate a point cloud as a byproduct for animal face recognition tasks.

5.2 Related Works

In our proposed face recognition system, we apply object segmentation [128] and monocular depth estimation [101] sub-tasks, and also perform point cloud registration [164] [165] based on key point detection. We do not need to train the monocular depth estimation and object segmentation network. However, we need to design an new 3D convolution [18] network to extract and fuse the features from consecutive input images. We also design a new point cloud registration algorithm based on monocular depth estimation, and key point detection. In this section, we separately review the state of the art work related to monocular depth estimation [101] and object segmentation [128], key point detection [123], and face detection [11].

5.2.1 Monocular depth estimation

Monocular depth estimation [166] is one of the fundamental tasks in computer vision. For a long time, the research on depth estimation [95] has focused on comparing the accuracy of some public data sets [82] [81], and some researchers [91] have focused on the accuracy of monocular depth estimation of dynamic objects in the scene. Recently more researchers [100] [85] focus on obtaining higher accuracy in monocular depth estimation for multiple data sets [167] [168]. However, this is limited by the representation capacity of traditional CNN networks [63]. In order to achieve consistent depth representation of multiple data sets, a consistency loss was proposed in Midas [100] and DPT [85] to improve the performance of monocular depth estimation in different scenarios.

5.2.2 Object segmentation

Object segmentation [46] has been an important task in computer vision. Some researchers [46] [104] focused on the accuracy of object segmentation [169]. However, with the large-scale application of object segmentation tasks, the traditional CNN network structure usually appears to be bloated. Recently many researchers have focused on light weighted models for object segmentation [170]. The research focus on object segmentation applications of large visual models [128] is to enable one model to accurately predict multiple objects in multiple scenes, which also enables multi-task learning to be implemented more efficiently.

5.2.3 Key point detection

Usually key point detection is used for human pose detection [171] [123] and some face recognition tasks [172] [173] that require affine transformation. It is difficult for large visual models to be directly and effectively applied to key point detection tasks. The emergence of more and more lightweight network models [64] [86] enables key point detection to be deployed on edge devices in many scenarios, which makes the application of related technologies [174] more conveniently.

5.2.4 Face recognition

Typically, face recognition consists of three steps: 1) face detection, 2) feature extraction, and 3) classification. Most algorithms for face recognition are based on pattern recognition

[11] [175]. With the emergence of the SVM classifier [176], the classification task was gradually replaced by SVM [177] [178]. In recent years, face recognition algorithms based on CNN networks have gradually become mainstream, such as end-to-end [179] [12] [11] or multi-stage [155] face recognition. However, there is a lack of research on multi-task face recognition in specific scenes. This chapter focuses on multi-task face recognition algorithms for animal face detection where training data is a major concern.

5.3 Methods

In this section, we will introduce our network in detail. Our ultimate goal is to achieve effective animal face recognition using consecutive images. In addition, a 3D point cloud is also reconstruction for a detected animal face. The overall process is divided into multiple subtasks, including monocular depth estimation, object segmentation, key point detection, point cloud registration, and 3D convolution feature extraction. The final output of the network is a similarity score and a 3D point cloud. From the similarity score, the face recognition result is deduced.

Different with end-to-end face recognition networks [12] [179], our network adopts a face recognition method based on multi-task learning, which has the following advantages.

1. Compared with the traditional end-to-end deep learning method, multi-task learning can produce more effective additional results. our network can detect animal face from three consecutive images taken from different view angles, and produce a point cloud for the detected face. Our multiple tasks include depth estimation, object segmentation, and key point detection.

2. Multi-task learning has a clear learning path, which can effectively improve the learning speed. In this chapter, depth estimation and object segmentation are used as subtasks to clarify the learning path of face recognition and improve the learning speed.

3. Sub-tasks can be supervised in multi-task learning, making the learning process more interpretable. The addition of depth estimation and object segmentation in this chapter allows us to effectively supervise the intermediate results of face recognition.

4. Sub-tasks can be replaced by pre-trained models with better generalization capabilities without the need for additional training, thereby reducing unnecessary computing

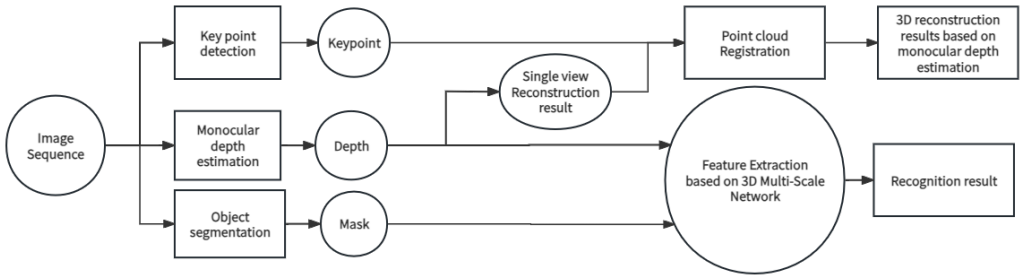

Figure 5.1: Our network structure

power and data requirements. The monocular depth estimation and object segmentation in this chapter directly adopt from existing pre-trained models, thus reducing the training requirements.

Fig 5.1 shows the overview of our proposed framework. After obtaining multiple consecutive images, each of them will be sent to a monocular depth estimation network, an object segmentation network, and a key point detection network to obtain the depth, mask and key points in the image respectively. The mask result is then used to filter out the background information in the image. The filtered image sequence and their depths are then regularized and fed into our 3D multi-scale network. Finally, 3D multi-scale features are extracted and used to output the final face recognition result. In addition, after completing the monocular depth estimation, object segmentation, and key point detection, we also use the depths and masks to generate a local point cloud of the face, and use the key points to align the point cloud to obtain the final point cloud result.

5.3.1 Monocular depth estimation and object segmentation

Spatial information is a very important part of face recognition. Before performing face recognition, it is necessary to obtain the depth information of image to extract spatial information. The addition of depth information allows us to generate a point cloud of the face. We use depth anything [101] pre-trained model for monocular depth estimation. This method greatly improves model's capacity by adding transformer to the model structure, and it uses 62 million labeled images as training data, so that effective monocular depth estimation results can be obtained in different scenarios at the same time.

The object segmentation network also uses the existing pre-training model. We use seg-

mentation anyting [128] for object segmentation, which is a pre-trained model for target detection based on the transformer architecture. The model is trained on a dataset containing more than 11 billion masks, making it capable of performing good zero-shot transfer learning. The role of the segmentation here is to filter out invalid background information in the image.

5.3.2 Key point detection and point cloud registration

Unlike monocular depth estimation and object segmentation, key points are task dependent. It is difficult to directly use pre-trained models for key point detection. CNN-based network structures can usually complete training faster. So our key point detection uses a network architecture HrNet [123]. Compared with the traditional CNN-based key point detection, HrNet cleverly integrates high-resolution features during the decoding process, so it has better performance for key point detection in high-resolution images. We train it with our own collected dataset for key point detection.

The key points are used to align and register the point cloud of detected animal face. After completing the monocular depth estimation and object segmentation, we can obtain animal facial point clouds from multiple angles. We need to align them based on key point detection. Unlike traditional point cloud registration tasks [164], our point cloud registration task has two features.

- It contains fewer corresponding points. This is because in order to reduce the amount of computation for 3D reconstruction, we use two animal facial images we obtained, but they have large difference in view angles. There are fewer overlap areas in the two images registered in our task. Therefore, we use the key point detection network to determine the overlap area between the two images, and then only rely on the point cloud information in the overlap area for point cloud registration by using ICP.

- Since the point cloud data is generated by the results of monocular depth estimation, the point cloud is very sparse in the area where the depth values have abrupt changes. This causes the challenge to point cloud registration. In our key point detection, we selected a group of key points that are relatively isolated in space (corner tips, ear tips, and nose tips), which could effectively alleviate the problem caused by the sparse areas.

Our point cloud registration include three steps:

1. First, we use a set of relatively isolated key points in space to obtain the spatial relationship between the two point clouds and the depth scale difference between the two images.

2. Use a set of key points to determine the overlap area between the two point clouds, and use the points in the area to perform ICP-based registration to further refine the spatial relationship between the two point clouds.

3. Based on the density and sparsity in the two point clouds, delete the relatively sparse parts of the two point clouds, retain the relatively dense parts, and complete the final point cloud registration.

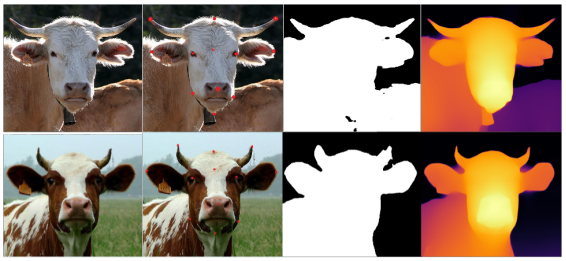

Figure 5.2: Sub-tasks for two images from left to right: original image, key points, segmentation, depth

Fig 5.2 shows the key point detection results for two images. When selecting key points, we took into account both the point cloud registration and segmentation tasks. We selected isolated points in space such as the ear tip and the horn tip, as well as some key points on the edge of the cow's face.

Fig 5.3 shows the registration result for two scenes, each of them has two images taken from different angles. The left and right ends of the figure are the original images used in the point cloud generation. With our registration algorithm, two point clouds obtained from the two images can be effectively integrated.

5.3.3 Feature extraction based on 3D convolution

After completing the depth estimation and object segmentation, the image needs to be feature extracted. For this purpose, we designed a feature extraction network based on 3D

Figure 5.3: Two point cloud registration samples

convolution. Compared with traditional 2D convolution, 3D convolution has the following advantages.

1. 3D convolution is better at extracting temporal information between consecutive images. The input of 3D convolution is usually a sequence of images with spatial or temporal correlation. In our task, 3D convolution can more effectively extract the facial features of animals in the image sequence.

2. 3D convolution has more parameters and therefore has a higher representation capacity. Generally, more network parameters mean that the network can accommodate more training data. The network parameters can be increased by increasing the depth or width of the network. The increase in network depth means that the network can extract higher-dimensional features, but it could cause gradients to disappear or explode during training. The increase in network width will not increase the network's ability to extract high-dimensional features, but the branches added for convolution allow the network to extract rich details in the image. In our task, the invalid background has been filtered out by mask before facial feature extraction, and the depth map is provided as input feature. Thus this task does not require a very complex feature extraction step. The use of 3D convolution in this task is to enrich the detailed features and increase the capacity of the deep learning network, which makes it possible to effectively extract features from some facial images that are not coherent in sequence.

3. Compared with 2D convolution, 3D convolution can better fuse the features of each channel. Fig 5.4 shows the difference between 2D convolution, 3D convolution and

the 3D convolution we use. Normally, it is not possible to use 3D convolution kernels to perform convolution on 2D images. When using 2D convolution to convolve a multi-channel image, we need to use multiple 2D convolution kernels to convolve each channel separately, and then add the feature maps obtained by convolution of each channel to obtain the final convolution result. Therefore, the feature extraction of each channel is relatively independent during the convolution process (please see 2D convolution with a sequence length of 1 in Fig 5.4).

Usually when using 3D convolution to convolve an image, three dimensions of a convolution kernel in 3D convolution correspond to the height, width and sequence length of the image. After convolution, different channels are fused together, and then the features are merged like 2D convolution (please see 3D convolution with a sequence length of 3 in Fig 5.4).

In our 3D feature extraction process on a single image, we treat the image as 3 single channels composed of R, G, and B information, and the size of the convolution kernel used is 3*3*3. The operation process of 3D convolution is that the convolution kernel performs dot product operations on all pixels in the area covered by the convolution kernel in the three adjacent channels. This operation allows 3D convolution to be effectively applied when the image sequence length is 1, while allowing information from different channels to be better integrated during the convolution process, thereby further improving the accuracy of image feature extraction (please see 3D convolution with a sequence length of 1 in Fig 5.4).

Fig 5.5 shows the structure of our 3D convolution feature extraction network. The image first passes through a pooling layer. Then the image is input into a CBAM module. The CBAM module consists of a spatial attention module and a channel attention module. The CBAM module has two main functions:

1. Help the model to better focus on important local areas and enhance spatial and channel attention.

2. Filter out unimportant features early and weight important features. Placing the CBAM block [53] before formal feature extraction can filter out some unimportant features

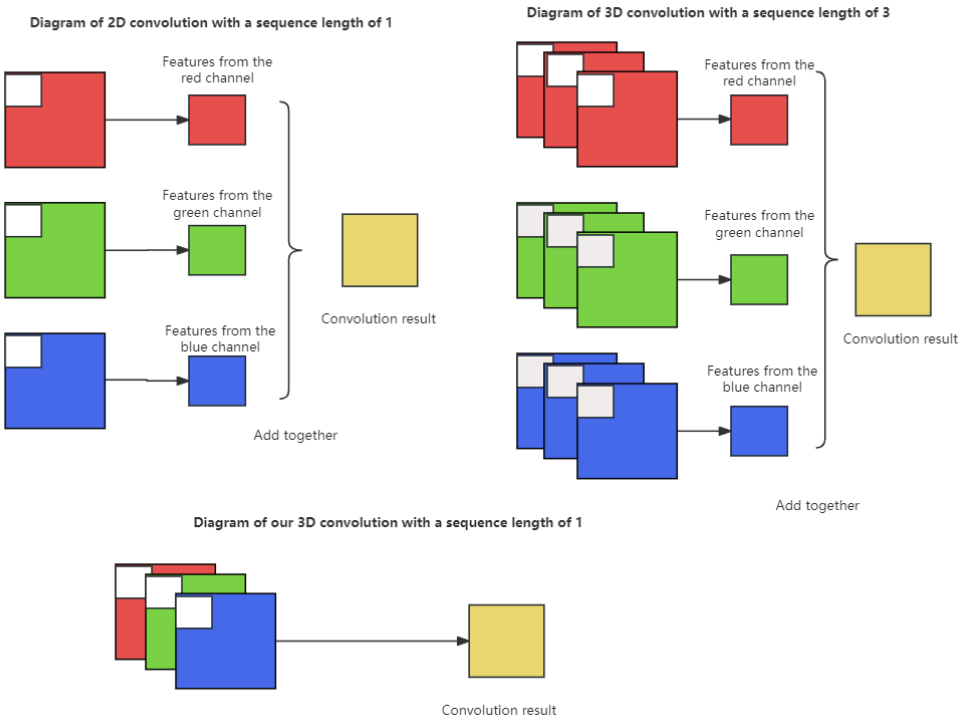

Figure 5.4: 2D convolution (top left), 3D convolution (top right), and our 3D convolution (bottom)

and improve the efficiency and effect of the model in the subsequent feature extraction process.

After the CBAM block [53], the feature map will be produced and go through four feature extraction stages. After completing the feature extraction of stage 4, we merge the features of stage 2 and stage 3 in the feature extraction process. The feature map outputs by stage 2 and stage 3 are convolved with two layers and one layer respectively, in order to unify the size of the feature map to the size of the feature map of stage 4, and concatenate the feature maps of the three parts. Then a convolution with a kernel size of 1 is used to integrate the information between channels of the merged feature map. This structural design allows the final feature extraction result to fuse deep and shallow features while also fusing information of different scales to further improve the feature extraction effect of important detail information in the image.

After completing all the feature extraction and multi-scale feature fusion stages, we obtain an 1024 feature map with a side length of 4.

We slice the feature map into 4 feature maps with a side length of 64 in sequence. Then we use SSIM(Structural Similarity Index) [180] to compute the similarity of the four feature maps to obtain the final recognition result. Traditionally, when the feature map is flattened, pooled and fully connected, each feature vector contains the features of the entire image, which makes the local features of the image easily invalid. Using sliced feature maps for similarity comparison can better avoid the problem of local feature invalidation. This allows the network to retain as many local features in the image as possible, improving the accuracy of the network in specific scenarios.

We also perform human face recognition tests during our experiment in order to show the generalization capability of our proposed network. After obtaining an 1024 feature map, the features are flattened to stretch the feature maps into one-dimensional tensors, and then further feature extraction is performed on the obtained one-dimensional tensors through the fully connected layer, and finally an one-dimensional tensor with a length of 1024 is obtained. The cosine distance is used for similarity calculation to obtain the final human face recognition result.

5.4 Experiments

In this section, we will present the details of our experiments and results. We used the GeForce GTX2080Ti graphics card to test various parts of our face recognition system. All of them were implemented and tested on PyTorch.

5.4.1 Datasets

We conducted tests on three datasets for different application scenarios.

1. For animal face recognition scenarios with consecutive images, we collected our own dataset, which contains video clips of 600 cows. We cut frames from the video clips and obtained 600 image sequences containing cow heads from different angles. The length of each sequence is 3. Through padding and resizing, we unified the size of the image to 600×600. All the images were annotated with key points as required. We divided the dataset into 500 sequences as training set and 100 sequences as test set.

2. Our network was also trained to detect human faces for evaluating its performance. For face recognition scenarios with multiple separate frames (they are not consecutive images), we used CelebA(CelebFaces Attributes Dataset) [10] as the training and test set. The CelebA [10] dataset was released by the Multimedia Laboratory of the Chinese University of Hong Kong and contains 202,599 facial images of 10,177 different celebrities. Each image was annotated with 40 attribute labels. The coordinates of key points including left and right eyes, nose, and mouth were also annotated.

3. We also tested our network for human face recognition scenario with one single image. We used LFW(Labeled Faces in the Wild) [9] as the test set. The LFW [9] dataset was released by the Department of Computer Science at the University of Massachusetts Amherst. The dataset contains 13,233 faces from 5,749 people. The images were collected from news photos on the Internet, and all the images were labeled with the identity information of the people. The size of all images is 250×250.

5.4.2 Evaluation metrics

Accuracy and *Std* (Standard Deviation) are used as evaluation indicators on the LFW dataset. *TAR@FAR*(True Acceptance Rate at given False Acceptance Rate) is used to evaluate datasets,

such as CelebA where there are multiple images of one target. Below we introduce the computing methods of *Accuracy* and *TAR@FAR* respectively.

First, we need to use the confusion matrix to calculate TP(True Positive), FP(False Positive), TN(True Negative), FN(False Negative) respectively.

Accuracy

The formula of *Accuracy* is as follows:

$$Accuracy = \frac{TP+FN}{TP+FP+TN+FN}$$

TAR@FAR

TAR@FAR is a commonly used metric in the field of biometric systems and authentication technologies. This metric is used to evaluate the performance of a verification system by balancing the trade-off between security (low FAR) and usability (high TAR).

Firstly, we introduce *TAR* and *FAR* separately.

- *TAR* **(True Acceptance Rate)**: The proportion of genuine user access attempts that are correctly accepted by the system. It is defined as:

$$TAR = \frac{TP}{TP+FN} \tag{5.1}$$

- *FAR* **(False Acceptance Rate)**: The proportion of impostor access attempts that are incorrectly accepted by the system. It is defined as:

$$FAR = \frac{FP}{FP+TN} \tag{5.2}$$

TAR@FAR is a performance measure that indicates the TAR at a specific FAR level. To find TAR at a given FAR, one must adjust the system threshold to achieve the desired FAR and then measure the corresponding TAR.

$$TAR@FAR = TAR|_{FAR=\alpha} \tag{5.3}$$

where α is the specified FAR level.

5.4.3 Main results

We evaluated the results on three datasets, CeLebA [10], LFW [9], and our own datasets. During the evaluation process, we compared the results with GhostFaceNetV2 [12], FaceNet [11], x2softmax [13], sphereface2 [14], and sface [15]. Table 5.1 shows the results when both the training set and the test set are CelebA [10]. Table 5.2 shows the test results when the training set is CelebA and the test set is LFW. Table 5.3 shows the test results when both the training set and the test set are our own datasets. To more intuitively show the difference between the networks, we plotted the ROC(Receiver Operating Characteristic Curve) curve when evaluating on the CelebA [10] dataset and our own dataset. We used *TAR@FAR* to evaluate the test results on the CelebA [10] dataset.

Table 5.1: Main result on CelebA

Method	TAR@FAR=0.01%	Flops	Parameter
Our network	0.5701	1.62	34.54
Our network for three images	0.9475	4.87	68.09
Our network with multi-scale	0.5720	1.89	87.65
Our network with multi-scale for three images	**0.9668**	5.43	78.96
GhostFaceNetV2 [12]	0.5763	**0.06**	**14.05**
FaceNet [11] (ResNet18)	0.1503	0.68	19.59
FaceNet [11] (ResNet50)	0.1467	1.46	57.09
X2softmax [13] (ResNet18)	0.7294	2.64	46.27
X2softmax [13] (ResNet50)	0.7924	6.34	65.83
Sphereface2 [14] (ResNet18)	0.7352	2.64	46.27
Sphereface2 [14] (ResNet50)	0.7723	6.34	65.83
Sface [15] (ResNet18)	0.7310	2.64	46.62
Sface [15] (ResNet50)	0.3549	6.34	66.06

From experiment results in Table 5.1, we found that when evaluating with the CelebA [10] dataset, our experimental results show a better performance when the sequence length of image was 1. When the sequence length of image was 3, our test result (0.9668) for *TAR@FAR* is significantly better than other face recognition networks used in the comparison. As the network is large, it consumes more time (flops) and uses large number of parameters.Through a comparison of the table, it can be found that with the addition of multi-scale information, while the input image sequence remains at 1, the reduction in the

high-resolution information in the feature extraction process leads to a slightly improved face recognition result. By increasing the number of input images, the advantages of 3D convolution are demonstrated. Compared to using a single image as input, the recognition accuracy significantly improves when the sequence length is 3. Eventually, with input sequence at 3 and the addition of multi-scale information, the optimal result of 0.9668 is obtained.

The sequence length of image in the LFW [9] is 1. The test results are shown in Table 5.2. From the experiments, we found that our test results still have higher accuracy and lower Std than other networks when evaluated with the LFW [9] dataset. Our network has higher flops and larger number of parameters.

Table 5.2: Main result on LFW

Method	Accuracy	Std	Flops	Parameter
Our network without CBAM	0.8470	0.01311	0.55	65.03
Our network	0.8575	0.01607	0.55	65.03
Our network with 2D convolution and multi-scale	0.8593	0.01677	0.55	65.03
Our network with multi-scale	**0.8670**	**0.01085**	0.55	65.03
GhostFaceNetV2 [12]	0.8300	0.01501	**0.06**	**14.05**
FaceNet [11](ResNet18)	0.7856	0.01847	0.68	19.59
FaceNet [11](ResNet50)	0.7881	0.02279	1.46	57.09
X2softmax [13](IResNet18)	0.8511	0.01505	2.64	46.27
X2softmax [13](IResNet50)	0.9110	0.14967	6.34	65.83
Sphereface2 [14](IResNet18)	0.8665	0.01347	2.64	46.27
Sface [15](IResNet18)	0.8600	0.01345	2.64	46.62
Sface [15](IResNet50)	0.7425	0.01128	6.34	66.06

Table5.2 presents the evaluation results using the LFW [9] dataset. The input sequence of all networks is 1. On LFW [9], due to the stable conditions such as the consistent lighting of the input image, the addition of the CBAM module [53] failed to improve the performance of network. However, the inclusion of the multi-scale method still effectively reduced the loss of high-resolution information during the convolution process and improves the final result of the network. Therefore, by using 3D convolution and multi-scale strategies, we achieved an accuracy of 0.867 on LFW. Compared with other existing face recognition algorithms, our method not only improves the accuracy, but also has a stable recognition result with a variance of only 0.01085.

When evaluating with our own dataset, we used both the sequence length 1 and sequence length 3. *TAR@FAR* is used as the evaluation metric.

From experiment results in Table 5.3, we found that our network can achieve good results for single input image and has the highest accuracy (0.9812) among the tested networks for 3 consecutive input images. Both the number of flops and the number of parameters in our network are higher. The results are consistent with previous experiment results. The use of 3D convolution and multi-task networks are the reasons for high metric scores.

Fig 5.8 shows the heatmap of the intermediate output of the feature network. It can be seen that with the addition of mask and CBAM, the feature extraction process focuses more on the face area.

The results of our 3D reconstruction are shown in the Fig 5.9. The top two rows from left to right. They are the original images of the cow from two angles, the monocular depth estimation results, and the reconstruction results of a single image. The point cloud results after registration are shown in the bottom two rows of images. As can be seen from the figure, through reconstruction, some details of the animal's facial point cloud are enriched, providing a data basis for farm animal monitor.

5.4.4 Ablation experiments

Table 5.4 shows the results of our ablation experiments on the CelbeA. By comparing the first and second rows, it is proved that the addition of CBAM improves the performance of network feature extraction. By comparing the third and fourth rows, we found that using 3D convolution with a three-frame sequence can improve the accuracy of the results. After comparing the fifth and sixth rows and the seventh and eighth rows, it can be proved that the addition of depth estimation results further improves the accuracy of the results. Then in the experiments of the ninth and tenth rows, we tested whether adding multi-scale information can further optimize the results. However, after attempting to add depth information following the inclusion of multi-scale information and trying to replace the traditional feature comparison method with SSIM, we found that these two methods cannot continuously improve our multi-task learning performance. This may be because the face itself does not contain a large amount of stereo information. The addition of multi-scale monocular depth estimation results increases the effective information while reducing the proportion of effec-

tive information in the input information, which in turn affects the effectiveness of the final feature extraction.

Fig 5.10 shows the ROC curves corresponding to the different experimental methods in table 5.4. From Fig 5.10, we can see that when the experimental method uses 3D convolution to extract features from three frames of images and adds CBAM and multi-scale methods, better feature extraction effects can be obtained on CelebA, with an AUC of 0.99921.

Table 5.5 shows the results of our ablation experiments on our own dataset, and Fig 5.11 shows the ROC curves corresponding to the different experimental methods in the table. Experiments on our dataset have proven that adding depth is effective when performing animal face recognition, as animal faces have more three-dimensional information than human faces. When performing animal face recognition, using SSIM to perform feature map similarity comparison instead of traditional tensor comparison can further optimize the results.

5.5 Conclusions

In this chapter, we proposed a new network for animal face recognition tasks from images. Human face datasets are also used to demonstrate its generalization capability over various domains. our network adopted a multi-task learning approach and a 3D convolution method for single image or consecutive image inputs. Our multi-task learning includes sub-tasks: monocular depth estimation, object segmentation, and key point detection. Our 3D convolution method extracts the spatial and temporal features from input images. They work together to improve the accuracy of face recognition and speed up the learning process, specially for animal face recognition. Our testing results demonstrate our proposed network performs very well on public datasets and our own dataset, and outperform some existing networks. We also used the key point detection and an ICP based registration method to reconstruct the 3D point clouds of detected face. In the follow-up work, we will try to use some zero-shot methods to implement sub-tasks to further simplify the network implementation process, and use the network for practice evaluation in animal breeding management applications.

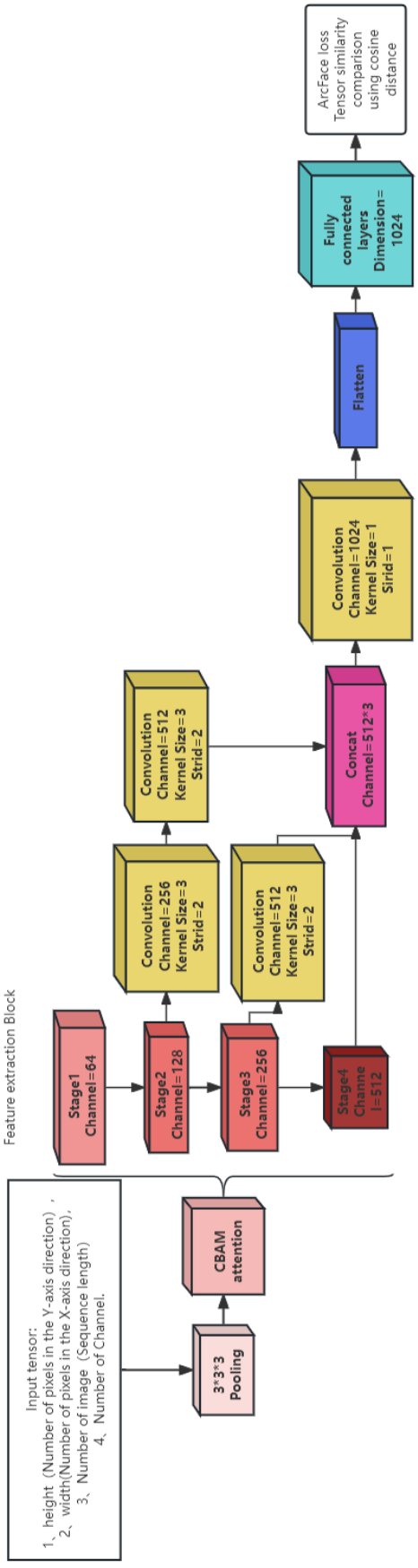

Figure 5.5: Network Structure of feature extraction network

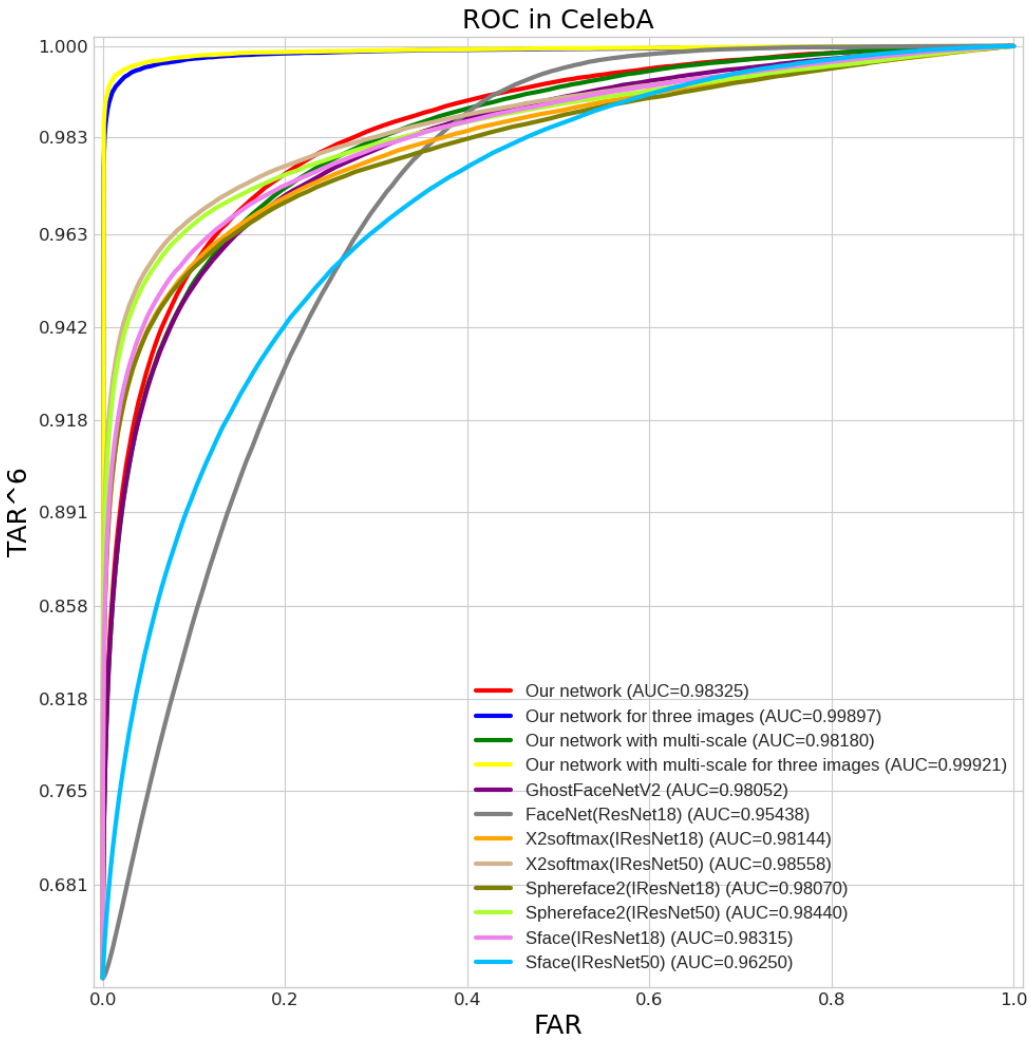

Figure 5.6: ROC curve on CelebA

Table 5.3: Main result on our dataset

Method	TAR@FAR=0.01%	Flops	Parameter
Our network with multi-scale, depth map and SSIM	0.8835	3.67	87.25
Our network with multi-scale, depth map and SSIM for three images	**0.9812**	10.75	78.85
GhostFaceNetV2 [12]	0.8448	**0.06**	**14.05**
FaceNet [11](ResNet18)	0.6552	0.68	19.59
FaceNet [11](ResNet50)	0.7378	1.46	57.09
X2softmax [13](IResNet18)	0.8265	2.64	46.27
X2softmax [13](IResNet50)	0.8946	6.34	65.83
Sphereface2 [14](IResNet18)	0.8336	2.64	46.27
Sphereface2 [14](IResNet50)	0.9294	6.34	65.83
Sface [15](IResNet18)	0.8311	2.64	46.62
Sface [15](IResNet50)	0.9113	6.34	66.06

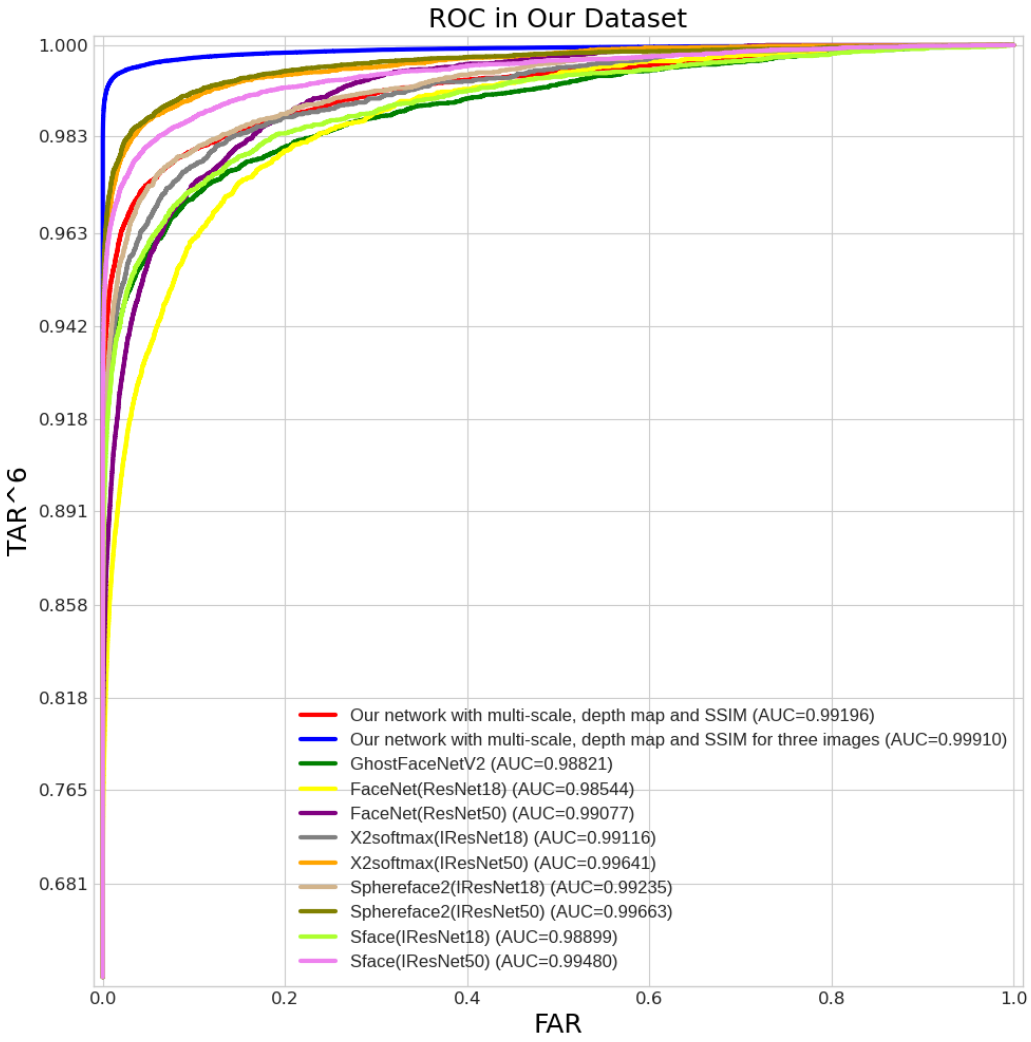

Figure 5.7: ROC curve on our dataset

Figure 5.8: Heatmap of feature extraction network

Figure 5.9: Our 3D Reconstrucion result

Table 5.4: Impact of different backbone on face recognition task on CelebA

Method	*TAR@FAR*	Flops	Parameter
Our network based on 2D convolution without CBAM for image from single view	0.5261	**0.68**	**28.98**
Our network based on 2D convolution for image from single view	0.5606	0.69	29.07
Our network without CBAM for image from single view	0.5533	2.01	51.01
Our network without CBAM for images from three views	0.9025	6.03	67.79
Our network for image from single view	0.5701	2.01	51.19
Our network for images from three views	0.9475	6.04	67.97
Our network for image from single view based on depth map	0.5711	4.02	59.58
Our network for images from three views based on depth map	0.9497	12.07	93.13
Our network with multi-scale for image from single view	0.5720	1.89	87.65
Our network with multi-scale for images from three views	**0.9668**	5.43	78.96
Our network with multi-scale for image from single view based on depth map	0.5685	3.67	87.25
Our network with multi-scale for images from three views based on depth map	0.9243	10.75	78.85
Our network with multi-scale for images from three views based on depth map and SSIM	0.9277	10.75	78.85

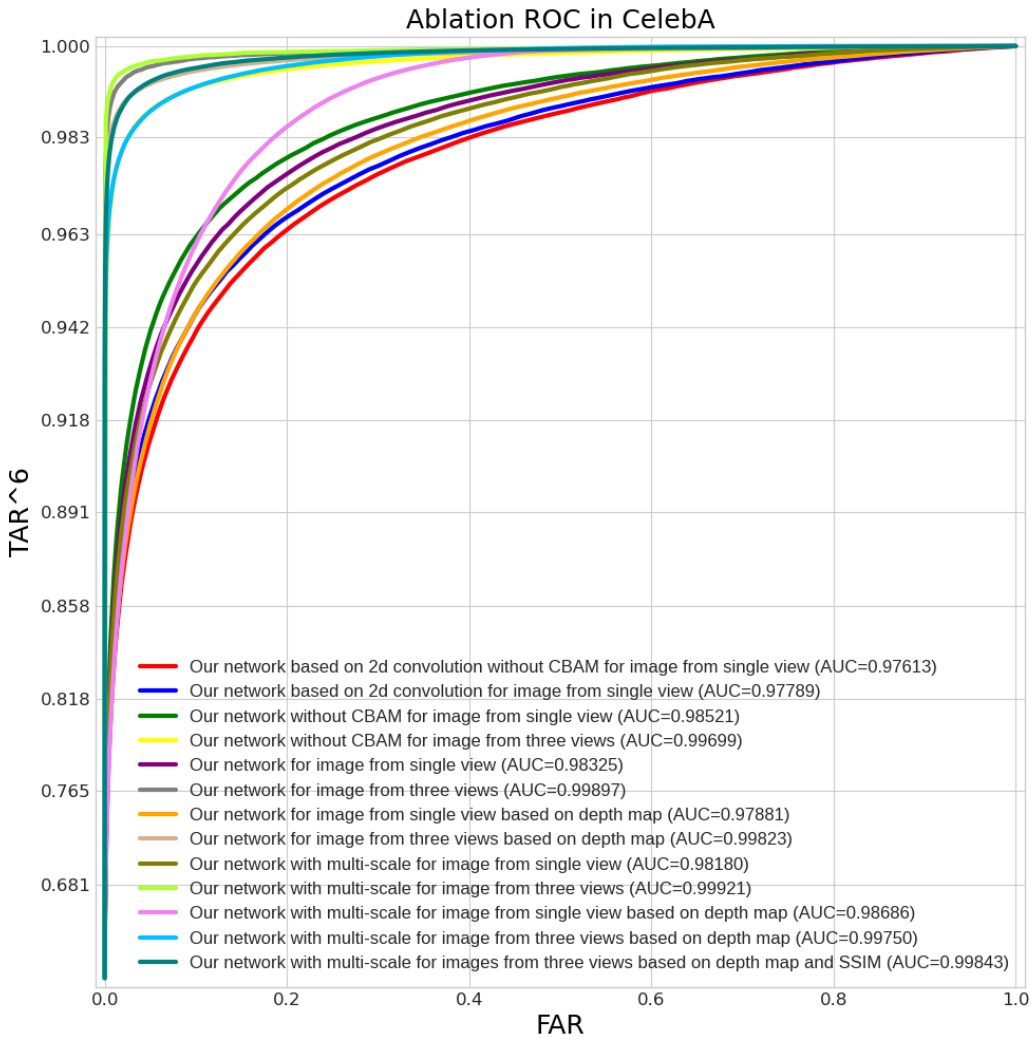

Figure 5.10: Ablation experiment ROC curve on CelebA

Table 5.5: Impact of different backbone on face recognition task on our dataset

Method	*TAR@FAR*	Flops	Parameter
Our network based on 2D convolution without CBAM for image from single view	0.7826	**0.68**	**28.98**
Our network based on 2D convolution for image from single view	0.7931	0.69	29.07
Our network without CBAM for image from single view	0.8156	2.01	51.01
Our network without CBAM for images from three views	0.9154	6.03	67.79
Our network for image from single view	0.8345	2.01	51.19
Our network for images from three views	0.9357	6.04	67.97
Our network for image from single view based on depth map	0.8564	4.02	59.58
Our network for images from three views based on depth map	0.9554	12.07	93.13
Our network with multi-scale for image from single view based on depth map	0.8808	3.67	87.25
Our network with multi-scale for images from three views based on depth map	0.9702	10.75	78.85
Our network with multi-scale for images from three views based on depth map and SSIM	**0.9812**	10.75	78.85

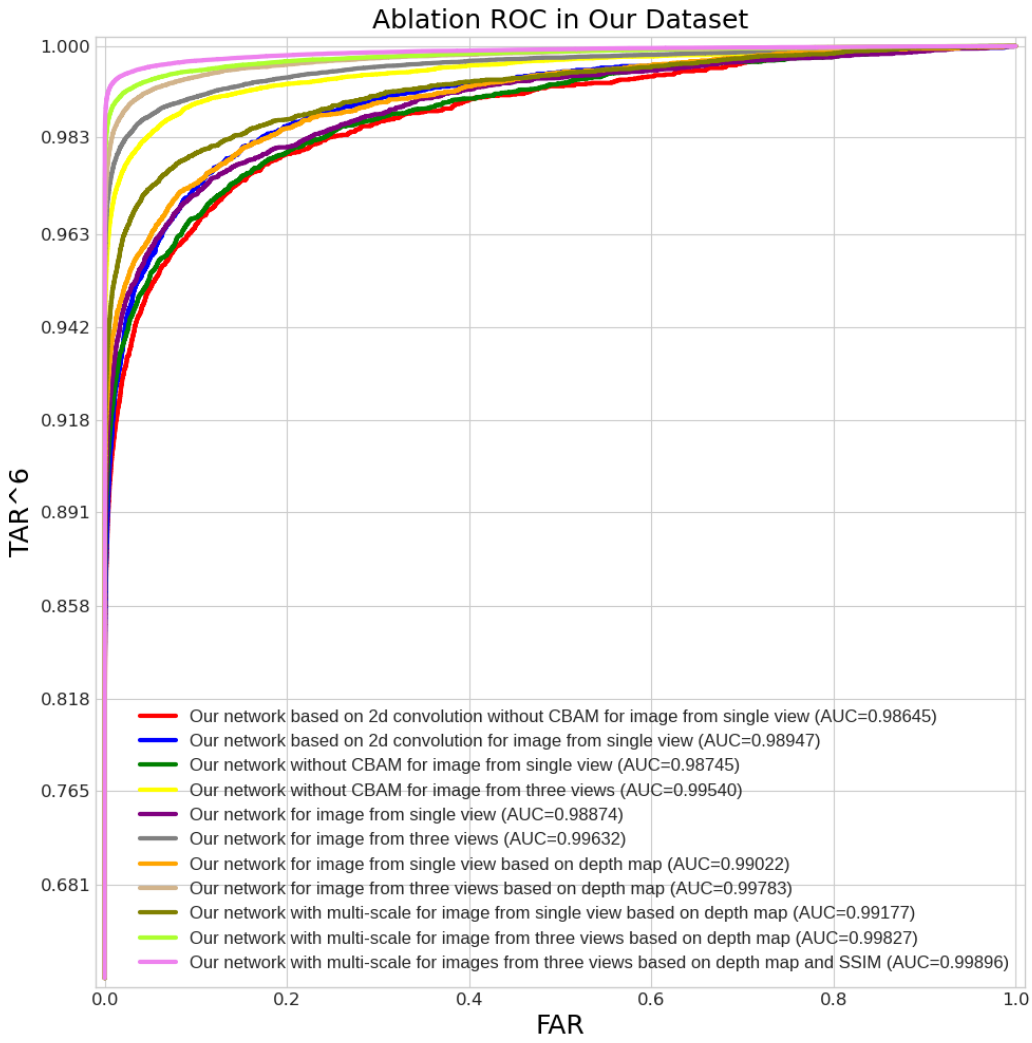

Figure 5.11: Ablation experiment ROC curve on Our dataset

Chapter 6

Conclusions and Future Work

This chapter summarizes my research work and outlines the important research contributions. In addition, this chapter lists the journal and conference publications of my research work and extends the discussion of future work.

6.1 Research Summary

My Ph.D. research focused on farms as task scenarios and used computer vision-related technologies as tools to build multi-task solutions that can solve practical problems encountered in the process of farm animal monitoring. To reduce the difficulty of farm animal monitoring, my research used images or image sequences as input.

The traditional process of farm animal supervision is made difficult by the large number of operations that require human intervention. At the same time, this non-automated manual monitoring process struggles to ensure the continuity and reliability of information flow. Due to human intervention, it is difficult to ensure the safety of both humans and animals during the monitoring process. Furthermore, the varied conditions of the environment and information collection hardware also pose challenges in the establishment of automated monitoring systems for farm animals. Therefore, the focus of this paper is to explore the possibility of automating farm animal supervision using deep learning multi-task learning methods. The application of deep learning greatly reduces the safety issues of manual operations and also allows monitoring results to be accumulated continuously and effectively for a long period of time. In addition to reducing the cost of data collection and improving the accuracy of results, the application of multi-task learning methods provides a modular

processing method that also enables the expansion of subsequent tasks in farm animal monitor. The accumulation of sub-task model data in the monitoring process can not only meet the optimization needs of existing tasks, but also become a solid foundation for solving new tasks in the future.

In the breeding animal detection method from images, traditional object detection is used as the task objective. Considering the practical difficulties in data collection in the process of farm animal monitor, the application of heatmap improves the efficiency of transfer learning using the classification network pre-training model and greatly reduces the data requirement for the object detection task. The addition of the post-processing step of Similarity Segmentation based on pattern recognition improves the accuracy of the model without adding additional data.

In order to solve the practical problem of parameter measurement in the process of farm animal monitoring, this thesis proposes a deep multi-task learning method for animal chest circumference estimation from monocular images. During process design, the architecture of deep learning multi-task learning was adopted, and monocular depth estimation was added as a subtask to the task process, which provided stereo information for feature extraction. By adding calibration objects of known size in the image, the scale of monocular depth estimation as well as the spatial relationship between the viewing plane and the measurement target were corrected. In order to deal with the problems of scene clutter and complex lighting conditions in farm animal monitoring scenarios, histogram equalization was used to mitigate the impact of lighting, and additional data augmentation steps were incorporated in the model training process. Object segmentation was also used as a subtask to remove the background, further weakening the impact of background clutter and complex scenes on the measurement results.

Finally, to address the problem of farm animal recognition in the process of farm animal monitoring, a 3D convolution-based animal face recognition method is proposed in multi-task learning for animal face recognition with spatio-temporal 3D convolution network based on the characteristics of animal faces. By modifying the 3D convolution, it has been adapted to accommodate different image input modes, including single-frame and multi-frame image input. In addition, by using object segmentation to eliminate the background, the robustness of the model is improved. The addition of monocular depth estimation and key point de-

tection increases the stereoscopic information in the process while achieving effective normalization of images at different angles. In addition to its success on animal face data, the method has also been tested on lfw and CelabA for single-frame and multi-frame image facial recognition results.

6.2 Research Contributions

The significant contributions of my research are briefly outlined as follows:

(1) **A Breeding Animal Detection Method from Images With Grided Heatmaps (Chapter 3)**

- A new object detection method was proposed for breeding animal images that have not been seen by the network before. The method is based on the pre-trained network model for classification tasks. There is no training or labelled datasets required in this method.
- A multi-scale gridded heatmap generation method was proposed for selecting region proposals and used for testing various feature similarities. In addition, a similarity segmentation technique was utilized to refine the bounding box generated from the heatmaps.
- The proposed method is compared on COCO [17] and VOC [16] as well as our own collected data. When the amount of labeled data is less than 200, the evaluation results are significantly better than YOLO [6], Fcos [7] and WSOD [8].

(2) **Deep Multi-task Learning for Animal Chest Circumference Estimation from Monocular Images (Chapter 4)**

- A new deep-learning framework for estimating the chest circumference of the cow from monocular images was proposed. This is a multi-task learning framework that can extract the key points, masks, and depth information of a cow in an image and estimate the parameter of the cow.
- A new alignment method for depth estimation was developed to align the ground plane with the image plane. The alignment method is based on the ground plane estimation and the reference object estimation (the cigarette box in the image).

- The object segmentation and key point detection were based on the same network, which is built up on top of the Deeplab-V3 [154]. They share the same features extracted from the input image.
- The proposed network framework was tested and evaluated on real images collected in the field. The results demonstrated framework is effective and efficient. The average absolute error in measuring chest circumference is less than 30 %.

(3) **Multi-task Learning for Animal Face Recognition with Spatio-temporal 3D Convolution Network (Chapter 5)**

- A multi-task-based facial recognition method was proposed, which includes three sub-tasks: object detection, depth estimation, and key point detection.
- A new 3D multi-scale network was proposed for extracting and fusing spatio-temporal features from consecutive images.
- The experiment results showed that facial recognition accuracy is improved with proposed multi-task learning and 3D multi-scale network on our collected datasets. On the LFW [9] dataset, the performance of our method was on par with advanced methods. When using three images as input sequences for facial recognition on the CelebA [10] dataset, our method achieved a 30% improvement in accuracy compared to the advanced methods.

6.3 Future Work

Based on current research in the multi-task learning area, three main directions are worthy of further exploration: robustness and real-time.

- **Robustness** For multi-task learning, the robustness of the subtasks determines the data requirements to achieve the task goals. Although transformer-based models have made deep learning tasks, such as object detection and object segmentation, more robust in different scenarios and for different task targets, deep learning such as depth estimation, which has scale issues, still needs to align scenes with different depth distributions through scale factors. In addition, in some tasks that require manual definition, such as key point detection, achieving one-shot is still very challenging.

- **Real-time** The application of multi-task learning allows tasks that were originally centralized to be dispersed into various subtasks. This feature makes the model deployment method more flexible, so it can reasonably allocate tasks with different computing power requirements to different computing power devices. This operation speeds up the efficiency of the model application. However, the collaboration between different tasks in multi-task learning makes the information transmission between subtasks an important factor affecting the efficiency of multi-task learning applications. In addition, transformer-based subtasks, while providing highly robust subtask selection for various scenarios of multi-task learning, also greatly increase the number of model parameters, thereby significantly reducing the running speed. Although technologies such as knowledge distillation, model pruning, mixed precision, half-precision can improve the running efficiency of the model to a certain extent, a contradiction still exists between robustness and running speed during the multi-task learning process.

In summary, this paper proposes some innovative multi-task learning strategies to solve practical problems encountered in farm animal monitoring. It has achieved advanced results in related tasks, among which the animal face recognition algorithm has achieved advanced performance compared with similar algorithms. It contributes to further advancing research in the field of multi-task learning.

Bibliography

[1] C. B. Vennerød, A. Kjærran, and E. S. Bugge, "Long short-term memory rnn," 2021. [Online]. Available: https://arxiv.org/abs/2105.06756

[2] N. B. Erichson, S. H. Lim, and M. W. Mahoney, "Gated recurrent neural networks with weighted time-delay feedback," 2022. [Online]. Available: https://arxiv.org/abs/2212.00228

[3] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[4] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," 2024. [Online]. Available: https://arxiv.org/abs/2404.19756

[5] T. Xiao and J. Zhu, "Introduction to transformers: an nlp perspective," 2023. [Online]. Available: https://arxiv.org/abs/2311.17633

[6] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," *ArXiv*, vol. abs/2107.08430, 2021.

[7] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9626–9635, 2019.

[8] M. Awan and J. Shin, "A robust context-aware proposal refinement method for weakly supervised object detection," *IEEE access : practical innovations, open solutions*, vol. 8, pp. 199 768–199 780, 2020, wsod class activation map heatmap .

[9] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007, http://vis-www.cs.umass.edu/lfw/.

[10] Y. Zhang, Z. Yin, Y. Li, G. Yin, J. Yan, J. Shao, and Z. Liu, "Celeba-spoof: Large-scale face anti-spoofing dataset with rich annotations," 2020.

[11] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2015. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2015.7298682

[12] M. Alansari, O. A. Hay, S. Javed, A. Shoufan, Y. Zweiri, and N. Werghi, "Ghostfacenets: Lightweight face recognition model from cheap operations," *IEEE Access*, vol. 11, pp. 35 429–35 446, 2023.

[13] J. Xu, X. Liu, X. Zhang, Y.-W. Si, X. Li, Z. Shi, K. Wang, and X. Gong, "X2-softmax: Margin adaptive loss function for face recognition," 2023.

[14] Y. Wen, W. Liu, A. Weller, B. Raj, and R. Singh, "Sphereface2: Binary classification is all you need for deep face recognition," 2022.

[15] F. Boutros, M. Huber, P. Siebke, T. Rieber, and N. Damer, "Sface: Privacy-friendly and accurate face recognition using synthetic data," 2022.

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[17] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[18] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," 2015. [Online]. Available: https://arxiv.org/abs/1412.0767

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016. [Online]. Available: https://arxiv.org/abs/1506.01497

[20] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 160–167. [Online]. Available: https://doi.org/10.1145/1390156.1390177

[21] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," 2016. [Online]. Available: https://arxiv.org/abs/1511.06114

[22] Z. Chen, S. Watanabe, H. Erdogan, and J. R. Hershey, "Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks," in *Interspeech*, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:13051257

[23] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data: A case study on user response prediction," 2016. [Online]. Available: https://arxiv.org/abs/1601.02376

[24] Y. Liu, P.-H. C. Chen, J. Krause, and L. Peng, "How to Read Articles That Use Machine Learning: Users' Guides to the Medical Literature," *JAMA*, vol. 322, no. 18, pp. 1806–1816, 11 2019. [Online]. Available: https://doi.org/10.1001/jama.2019.16489

[25] Y. Chen, D. Zhao, L. Lv, and Q. Zhang, "Multi-task learning for dangerous object detection in autonomous driving," *Inf. Sci.*, vol. 432, pp. 559–571, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:45437456

[26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[27] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” 2016. [Online]. Available: https://arxiv.org/abs/1504.00702

[28] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” 2011. [Online]. Available: https://arxiv.org/abs/1103.0398

[29] A. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese, “Taskonomy: Disentangling task transfer learning,” 2018. [Online]. Available: https://arxiv.org/abs/1804.08328

[30] A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” 2018. [Online]. Available: https://arxiv.org/abs/1705.07115

[31] X. Liu, P. He, W. Chen, and J. Gao, “Multi-task deep neural networks for natural language understanding,” 2019. [Online]. Available: https://arxiv.org/abs/1901.11504

[32] T. Wei, J. Qi, and S. He, “A flexible multi-task model for bert serving,” 2022. [Online]. Available: https://arxiv.org/abs/2107.05377

[33] H. Benmeziane, K. E. Maghraoui, H. Ouarnoughi, S. Niar, M. Wistuba, and N. Wang, “A comprehensive survey on hardware-aware neural architecture search,” 2021. [Online]. Available: https://arxiv.org/abs/2101.09336

[34] Y. Gandelsman, A. Shocher, and M. Irani, “”double-dip”: Unsupervised image decomposition via coupled deep-image-priors,” 2018. [Online]. Available: https://arxiv.org/abs/1812.00467

[35] H. Li, P. Xiong, H. Fan, and J. Sun, “Dfanet: Deep feature aggregation for real-time semantic segmentation,” 2019. [Online]. Available: https://arxiv.org/abs/1904.02216

[36] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2014. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.214

[37] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," 2016. [Online]. Available: https://arxiv.org/abs/1603.06937

[38] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," 2018. [Online]. Available: https://arxiv.org/abs/1711.07319

[39] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800–810, 2001.

[40] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. L. Yuille, "Deep co-training for semi-supervised image recognition," *ArXiv*, vol. abs/1803.05984, 2018.

[41] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014. [Online]. Available: https://arxiv.org/abs/1311.2524

[42] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[43] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "Yolov6: A single-stage object detection framework for industrial applications," 2022. [Online]. Available: https://arxiv.org/abs/2209.02976

[44] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," 2018. [Online]. Available: https://arxiv.org/abs/1804.06208

[45] A. Sengupta, F. Jin, R. Zhang, and S. Cao, "mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns," *IEEE Sensors*

Journal, vol. 20, no. 17, p. 10032–10044, Sep. 2020. [Online]. Available: http://dx.doi.org/10.1109/JSEN.2020.2991741

[46] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[47] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, 2018.

[48] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style convnets great again," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13 728–13 737, 2021.

[49] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," 2017. [Online]. Available: https://arxiv.org/abs/1707.01083

[50] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2020. [Online]. Available: https://arxiv.org/abs/1905.11946

[51] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," 2020. [Online]. Available: https://arxiv.org/abs/1910.03151

[52] Z. Qin, P. Zhang, F. Wu, and X. Li, "Fcanet: Frequency channel attention networks," 2021. [Online]. Available: https://arxiv.org/abs/2012.11879

[53] S. Woo, J. Park, J.-Y. Lee, and I.-S. Kweon, "CBAM: Convolutional block attention module," in *European Conference on Computer Vision*, 2018.

[54] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2016.

[55] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," 2017. [Online]. Available: https://arxiv.org/abs/1710.09829

[56] R. M. Schmidt, "Recurrent neural networks (rnns): A gentle introduction and overview," 2019. [Online]. Available: https://arxiv.org/abs/1912.05911

[57] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[58] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014. [Online]. Available: https://arxiv.org/abs/1406.2661

[59] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[61] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," 2021. [Online]. Available: https://arxiv.org/abs/2104.10972

[62] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," 2021. [Online]. Available: https://arxiv.org/abs/1812.08434

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[64] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[65] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2019. [Online]. Available: https://arxiv.org/abs/1801.04381

[66] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018, pp. 7132–7141.

[67] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[68] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. kin Wong, and W. chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," 2015. [Online]. Available: https://arxiv.org/abs/1506.04214

[69] L. Sifre and S. Mallat, "Rigid-motion scattering for texture classification," *ArXiv*, vol. abs/1403.1687, 2014.

[70] B. Liao, X. Wang, L. Zhu, Q. Zhang, and C. Huang, "Vig: Linear-complexity visual sequence learning with gated linear attention," 2024. [Online]. Available: https://arxiv.org/abs/2405.18425

[71] X. Qi, Z. Liu, R. Liao, P. H. S. Torr, R. Urtasun, and J. Jia, "GeoNet++: Iterative geometric neural network with edge-aware refinement for joint depth and surface normal estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 969–984, 2020.

[72] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris, and Y. Aksoy, "Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9680–9689, 2021.

[73] V. Patil, C. Sakaridis, A. Liniger, and L. V. Gool, "P3Depth: Monocular depth estimation with a piecewise planarity prior," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1600–1611, 2022.

[74] J. Watson, O. M. Aodha, V. Prisacariu, G. Brostow, and M. Firman, "The temporal opportunist: Self-supervised multi-frame monocular depth," 2021. [Online]. Available: https://arxiv.org/abs/2104.14540

[75] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. D. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Neural Information Processing Systems*, 2019.

[76] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," 2016. [Online]. Available: https://arxiv.org/abs/1603.04992

[77] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," 2017. [Online]. Available: https://arxiv.org/abs/1609.03677

[78] A. R. Zamir, A. Sax, T. Yeo, O. F. Kar, N. Cheerla, R. Suri, Z. Cao, J. Malik, and L. J. Guibas, "Robust learning through cross-task consistency," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11 194–11 203, 2020.

[79] M. Dehghani, J. Djolonga, B. Mustafa, P. Padlewski, J. Heek, J. Gilmer, A. Steiner, M. Caron, R. Geirhos, I. Alabdulmohsin, R. Jenatton, L. Beyer, M. Tschannen, A. Arnab, X. Wang, C. Riquelme, M. Minderer, J. Puigcerver, U. Evci, M. Kumar, S. van Steenkiste, G. F. Elsayed, A. Mahendran, F. Yu, A. Oliver, F. Huot, J. Bastings, M. P. Collier, A. Gritsenko, V. Birodkar, C. Vasconcelos, Y. Tay, T. Mensink, A. Kolesnikov, F. Pavetić, D. Tran, T. Kipf, M. Lučić, X. Zhai, D. Keysers, J. Harmsen, and N. Houlsby, "Scaling vision transformers to 22 billion parameters," 2023. [Online]. Available: https://arxiv.org/abs/2302.05442

[80] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021. [Online]. Available: https://arxiv.org/abs/2103.14030

[81] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[82] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset," in *CVPR Workshop on The Future of Datasets in Vision*, 2015.

[83] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision (ECCV)*. Springer, 2012, pp. 746–760.

[84] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 1623–1637, 2019.

[85] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12 159–12 168, 2021.

[86] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1577–1586, 2019.

[87] C. Godard, O. M. Aodha, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3827–3837, 2018.

[88] H. Zhan, C. S. Weerasekera, J. Bian, R. Garg, and I. D. Reid, "DF-VO: What should be learnt for visual odometry?" *ArXiv*, vol. abs/2103.00933, 2021.

[89] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," 2017. [Online]. Available: https://arxiv.org/abs/1704.07813

[90] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[91] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1983–1992, 2018.

[92] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3d packing for self-supervised monocular depth estimation," 2020. [Online]. Available: https://arxiv.org/abs/1905.02693

[93] Q. Dai, V. Patil, S. Hecker, D. Dai, L. Van Gool, and K. Schindler, "Self-supervised object motion and depth estimation from video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[94] Y. Wang, Z. Yang, P. Wang, Y. Yang, C. Luo, and W. Xu, "Joint unsupervised learning of optical flow and depth by watching stereo videos," 2018. [Online]. Available: https://arxiv.org/abs/1810.03654

[95] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, "Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8976–8985, 2019.

[96] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," 2018. [Online]. Available: https://arxiv.org/abs/1802.05522

[97] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," 2018. [Online]. Available: https://arxiv.org/abs/1811.06152

[98] Y. Zou, Z. Luo, and J.-B. Huang, "Df-net: Unsupervised joint learning of depth and flow using cross-task consistency," 2018. [Online]. Available: https://arxiv.org/abs/1809.01649

[99] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow

and motion segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[100] S. Bhatia, B. Hooi, M. Yoon, K. Shin, and C. Faloutsos, "Midas: Microcluster-based detector of anomalies in edge streams," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 3242–3249, 2020.

[101] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," 2024. [Online]. Available: https://arxiv.org/abs/2401.10891

[102] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," 2023. [Online]. Available: https://arxiv.org/abs/2302.12288

[103] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," 2023. [Online]. Available: https://arxiv.org/abs/2302.05543

[104] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "SOLOv2: Dynamic, faster and stronger," *ArXiv*, vol. abs/2003.10152, 2020.

[105] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," 2023. [Online]. Available: https://arxiv.org/abs/1605.06409

[106] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018. [Online]. Available: https://arxiv.org/abs/1708.02002

[107] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016. [Online]. Available: https://arxiv.org/abs/1506.02640

[108] R. Girshick, "Fast r-cnn," 2015. [Online]. Available: https://arxiv.org/abs/1504.08083

[109] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016. [Online]. Available: https://arxiv.org/abs/1506.01497

[110] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," 2016. [Online]. Available: https://arxiv.org/abs/1612.08242

[111] R. Joseph and F. Ali, "YOLOv3: An Incremental Improvement," *arXiv.org*, pp. 1–6, Apr. 2018. [Online]. Available: http://arxiv.org/abs/1804.02767v1

[112] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 73–80.

[113] S. J. F. Guimarães, J. Cousty, Y. Kenmochi, and L. Najman, "An efficient hierarchical graph based image segmentation," 2012. [Online]. Available: https://arxiv.org/abs/1206.2807

[114] Z. Tian, C. Shen, X. Wang, and H. Chen, "Boxinst: High-performance instance segmentation with box annotations," 2020. [Online]. Available: https://arxiv.org/abs/2012.02310

[115] J. Lin, Y. Shen, B. Wang, S. Lin, K. Li, and L. Cao, "Weakly supervised open-vocabulary object detection," 2023. [Online]. Available: https://arxiv.org/abs/2312.12437

[116] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2017. [Online]. Available: https://arxiv.org/abs/1612.03144

[117] P. Lu, T. Jiang, Y. Li, X. Li, K. Chen, and W. Yang, "Rtmo: Towards high-performance one-stage real-time multi-person pose estimation," 2024. [Online]. Available: https://arxiv.org/abs/2312.07526

[118] S. Kreiss, L. Bertoni, and A. Alahi, "PifPaf: Composite fields for human pose estimation," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11 969–11 978, 2019.

[119] H. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2353–2362, 2016.

[120] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022. [Online]. Available: https://arxiv.org/abs/2207.02696

[121] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European Conference on Computer Vision*, 2015.

[122] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *ArXiv*, vol. abs/1904.07850, 2019.

[123] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5686–5696, 2019.

[124] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2016. [Online]. Available: https://arxiv.org/abs/1511.00561

[125] Y. Yang, W. Zhang, J. Wu, W. Zhao, and A. Chen, "Deconvolution-and-convolution networks," 2021. [Online]. Available: https://arxiv.org/abs/2103.11887

[126] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," 2016. [Online]. Available: https://arxiv.org/abs/1603.06937

[127] Z. Tian, T. He, C. Shen, and Y. Yan, "Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation," 2019. [Online]. Available: https://arxiv.org/abs/1903.02120

[128] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023. [Online]. Available: https://arxiv.org/abs/2304.02643

[129] T. Cao, L. Du, X. Zhang, S. Chen, Y. Zhang, and Y.-F. Wang, "CaT: Weakly supervised object detection with category transfer," *CoRR*, vol. abs/2108.07487, 2021, full-annotation image-level .

[130] A. Howard, M. Sandler, G. Chu, L. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," *CoRR*, vol. abs/1905.02244, 2019. [Online]. Available: http://arxiv.org/abs/1905.02244

[131] S. Guo, Y. Wang, Q. Li, and J. Yan, "DMCP: differentiable markov channel pruning for neural networks," *CoRR*, vol. abs/2005.03354, 2020, . [Online]. Available: https://arxiv.org/abs/2005.03354

[132] C. Hu, H. Wu, X. Li, C. Ma, X. Chen, J. Yan, B. Wang, and X. Liu, "Less or more from teacher: Exploiting trilateral geometry for knowledge distillation," 2024, .

[133] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," *CoRR*, vol. abs/2103.13630, 2021, . [Online]. Available: https://arxiv.org/abs/2103.13630

[134] C. Luo, "Understanding diffusion models: A unified perspective," 2022, vAE.

[135] Z. Chen, Z. Fu, R. Jiang, Y. Chen, and X.-s. Hua, "SLV: Spatial Likelihood Voting for Weakly Supervised Object Detection," https://arxiv.org/abs/2006.12884v1, Jun. 2020, spatial likelihood voting (SLV) mil proposal spatial probability proposal voting proposal .

[136] T. Zhu, B. Ferenczi, P. Purkait, T. Drummond, H. Rezatofighi, and A. van den Hengel, "Knowledge combination to learn rotated detection without rotated annotation," 2023.

[137] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," *CoRR*, vol. abs/1808.01974, 2018. [Online]. Available: http://arxiv.org/abs/1808.01974

[138] F. Shao, Y. Luo, L. Chen, P. Liu, W. Yang, Y. Yang, and J. Xiao, "Counterfactual co-occurring learning for bias mitigation in weakly-supervised object localization," 2024.

[139] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," 2018.

[140] V. Thangaraj and S. Esakkirajan, *Digital Image Processing*, Sep. 2009, hist .

[141] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1, hog .

[142] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002, lbp .

[143] C. Zauner, "Implementation and benchmarking of perceptual image hash functions," 2010, phash .

[144] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *CoRR*, vol. abs/1709.01507, 2017. [Online]. Available: http://arxiv.org/abs/1709.01507

[145] S. Mehta and M. Rastegari, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," *CoRR*, vol. abs/2110.02178, 2021. [Online]. Available: https://arxiv.org/abs/2110.02178

[146] W. Wang, E. Xie, X. Li, D. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pvtv2: Improved baselines with pyramid vision transformer," *CoRR*, vol. abs/2106.13797, 2021. [Online]. Available: https://arxiv.org/abs/2106.13797

[147] K. Tian, Y. Jiang, Q. Diao, C. Lin, L. Wang, and Z. Yuan, "Designing bert for convolutional networks: Sparse and hierarchical masked modeling," 2023.

[148] J. Wang, S. Zhang, Y. Liu, T. Wu, Y. Yang, X. Liu, K. Chen, P. Luo, and D. Lin, "Riformer: Keep your vision backbone effective while removing token mixer," 2023.

[149] F. Scholkmann, J. Boss, and M. Wolf, "ampd: An algorithm for automatic peak detection in noisy periodic and quasi-periodic signals," *Algorithms*, vol. 5, no. 4, pp. 588–603, 2016.

[150] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1302–1310, 2016.

[151] R. Deng, G. Zeng, G. Rui, and H. Zha, "Image-based building reconstruction with manhattan-world assumption," in *Pattern Recognition*, 2011.

[152] P. Soviany and R. T. Ionescu, "Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction," *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pp. 209–214, 2018.

[153] R. A. Güler, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos, "DenseReg: Fully convolutional dense shape regression in-the-wild," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2614–2623, 2016.

[154] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *ArXiv*, vol. abs/1706.05587, 2017.

[155] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-stage dense face localisation in the wild," *ArXiv*, vol. abs/1905.00641, 2019.

[156] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time," 2022. [Online]. Available: https://arxiv.org/abs/2211.03375

[157] T. Chen and D. Gu, "CSA6D: Channel-spatial attention networks for 6d object pose estimation," *Cognitive Computation*, 2021.

[158] F. Wang and Z. Xie, "An adversarial multi-task learning method for chinese text correction with semantic detection," 2023.

[159] P. Xu, J. Cai, Y. Gao, Z. Rong, and H. Xin, "MIRACLE: Multi-task learning based interpretable regulation of autoimmune diseases through common latent epigenetics," 2023.

[160] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *ArXiv*, vol. abs/1711.05101, 2017.

[161] B. Cheng, R. Girshick, P. Dollár, A. C. Berg, and A. Kirillov, "Boundary IoU: Improving object-centric image segmentation evaluation," 2021.

[162] J. Chen, S. hong Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S. H. G. Chan, "Run, don't walk: Chasing higher flops for faster neural networks," 2023.

[163] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, p. 1499–1503, Oct. 2016. [Online]. Available: http://dx.doi.org/10.1109/LSP.2016.2603342

[164] C. Lv, W. Lin, and B. Zhao, "Kss-icp: Point cloud registration based on kendall shape space," *arXiv preprint arXiv:2211.02807*, 2022.

[165] P. J. Besl and N. D. Mckay, "A method for registration of 3-d shapes," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 14, no. 3, pp. 239–256, 1992.

[166] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: Learning of structure and motion from video," 2017.

[167] N. Liu, N. Zhang, L. Shao, and J. Han, "Learning selective mutual attention and contrast for rgb-d saliency detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[168] J. Cho, D. Min, Y. Kim, and K. Sohn, "Diml/cvl rgb-d dataset: 2m rgb-d images of natural indoor and outdoor scenes," 2021.

[169] X. Liu, J. Hu, H. Wang, Z. Zhang, X. Lu, C. Sheng, S. Song, and J. Nie, "Gaussian-iou loss: Better learning for bounding box regression on pcb component detection," *Expert Systems with Application*, no. Mar., p. 190, 2022.

[170] G. Xu, J. Li, G. Gao, H. Lu, J. Yang, and D. Yue, "Lightweight real-time semantic segmentation network with efficient transformer and cnn," 2023.

[171] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," 2020.

[172] S. Shi, "Facial keypoints detection," 2017.

[173] M. Kim, Y. Su, F. Liu, A. Jain, and X. Liu, "Keypoint relative position encoding for face recognition," 2024.

[174] P. Dileep, B. K. Bolla, and S. Ethiraj, "Revisiting facial key point detection: An efficient approach using deep neural networks," 2022.

[175] M. Abdullah, "Optimizing face recognition using pca," *International Journal of Artificial Intelligence amp; Applications*, vol. 3, no. 2, p. 236–31, Mar. 2012. [Online]. Available: http://dx.doi.org/10.5121/ijaia.2012.3203

[176] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Microsoft, Tech. Rep. MSR-TR-98-14, April 1998. [Online]. Available: https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/

[177] T. H. Le and L. Bui, "Face recognition based on svm and 2dpca," 2011.

[178] S. M. Valiollahzadeh, A. Sayadiyan, and M. Nazari, "Face detection using adaboosted svm-based component classifier," 2008.

[179] J. Deng, J. Guo, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4685–4694, 2018.

[180] J. Nilsson and T. Akenine-Möller, "Understanding ssim," 2020.