Contents lists available at ScienceDirect

Displays

journal homepage: www.elsevier.com/locate/displa

A review of visualisation-as-explanation techniques for convolutional neural networks and their evaluation[☆]

Elhassan Mohamed [*], Konstantinos Sirlantzis, Gareth Howells

Intelligent Interactions Research Group, Kent Assistive RObotics Laboratory (KAROL), School of Engineering, University of Kent, UK

A R T I C L E I N F O

Keywords:
Activation heatmaps
Architecture understanding
Black-box representations
CNN visualisation
Convolutional neural networks
Explainable AI
Feature visualisation
Interpretable neural networks
Saliency maps
XAI

A B S T R A C T

Visualisation techniques are powerful tools to understand the behaviour of Artificial Intelligence (AI) systems. They can be used to identify important features contributing to the network decisions, investigate biases in datasets, and find weaknesses in the system’s structure (e.g., network architectures). Lawmakers and regulators may not allow the use of smart systems if these systems cannot explain the logic underlying a decision or action taken. These systems are required to offer a high level of ’transparency’ to be approved for deployment. Model transparency is vital for safety–critical applications such as autonomous navigation and operation systems (e.g., autonomous trains or cars), where prediction errors may have serious implications. Thus, being highly accurate without explaining the basis of their performance is not enough to satisfy regulatory requirements. The lack of system interpretability is a major obstacle to the wider adoption of AI in safety–critical applications. Explainable Artificial Intelligence (XAI) techniques applied to intelligent systems to justify their decisions offers a possible solution. In this review, we present state-of-the-art explanation techniques in detail. We focus our presentation and critical discussion on visualisation methods for the most adopted architecture in use, the Convolutional Neural Networks (CNNs), applied to the domain of image classification. Further, we discuss the evaluation techniques for different explanation methods, which shows that some of the most visually appealing methods are unreliable and can be considered a simple feature or edge detector. In contrast, robust methods can give insights into the model behaviour, which helps to enhance the model performance and boost the confidence in the model’s predictions. Besides, the applications of XAI techniques show their importance in many fields such as medicine and industry. We hope that this review proves a valuable contribution for researchers in the field of XAI.

1. Introduction

The significant success of Convolutional Neural Networks (CNNs) in image and video-based tasks such as image classification [1–3], object detection [4–6], and semantic segmentation [7–9] is bounded by their inherent inability of explaining their behaviours. Consequently, the adoption and deployment of CNN-based systems in safety–critical real-life applications, such as medical, automation and assistive robotics, is limited. These industries require reliable and explainable systems that integrate trust in the decision process with no or very low error tolerance. Besides, when a system failure occurs, it should be possible to justify and interpret its source to avoid it in the future. Thus, it is challenging to trust a black box system without understanding the intuitions behind its predictions.

The complex nature of CNNs makes the interpretation process challenging because it is difficult to identify the relations between the activities of individual neurons and the outcome of the neural network. That is why the explanations of CNNs predictions need to be considered in a wider frame of connections between several neurons or layers to attain a comprehensive understnading of the final result. CNN-based models can perform significantly better than conventional computer vision algorithms in terms of accuracy. However, the intractability of failures in CNN-based systems, when they occur, is a critical flaw.

Model interpretation through visualisation, or any other means of analysis, is an overlooked step in many systems, even though it can greatly help in improving the systems’ robustness if appropriately utilised. It can provide insights into how the network operates at each time step. This can explain the effort to understand and verify powerful deep network methods, not only to improve reliability for real-life application

[☆] This paper was recommended for publication by Prof G Guangtao Zhai.

[*] Corresponding author.
E-mail address: enrm4@kent.ac.uk (E. Mohamed).

https://doi.org/10.1016/j.displa.2022.102239
Received 28 November 2021; Received in revised form 30 March 2022; Accepted 15 May 2022
Available online 20 May 2022

Abbreviations

ACoL	Adversarial Complementary Learning
AI	Artificial Intelligence
AM	Activation Maximisation
AGG-Mean	Aggregating Mean
AGG-Var	Aggregating Variance
AUC	Area Under Curve
CAM	Class Activation Map
CNN	Convolutional Neural Network
CCAM	Common Class Activation Map
CLEAR	CLass-Enhanced Attentive Response
CLRP	Contrastive Layer-wise Relevance Propagation
DCNN	Deep Convolutional Neural Network
DeepLIFT	Deep Learning Important FeaTures
DeSaliNet	Deconvolutional Salient Network
DGN	Deep Generator Network
DNN	Deep Neural Networks
DTD	Deep Taylor Decomposition
FC layer	Fully Connected layer
FGVis	Fine-Grained Visual explanation
FullGrad	Full-Gradient
GAIN	Guided Attention Inference Network
GI	Gradient $\odot$ Input
GAP	Global Average Pooling
GBP	Guided Back Propagation
Grad-CAM	Gradient Class Activation Map
HoG	Histogram of Gradients
IG	Integrated Gradients
LBP	Local Binary Pattern
LIME	Local Interpretable Model-agnostic Explanation
LRP	Layer-wise Relevance
MSE	Mean Square Error
PCC	Pearson Correlation Coefficient
R-CNN	Regions with CNN features
ReLU	Rectified linear unit
RISE	Random Input Sampling for Explanation
SDR	Smallest Destroying Region
SENN	Self-Explaining Neural Network
SGD	Stochastic Gradient Descent
SGLRP	Softmax-Gradient Layer-wise Relevance Propagation
SHAP	SHapley Additive exPlanation
Smooth-Grad	Smooth-Gradient
SPG	Self-Produced Guidance
SSIM	Structure Similarity Index
SSR	Smallest Sufficient Region
t-SNE	t-Distributed Stochastic Neighbor Embedding
XAI	Explainable Artificial Intelligence
XRAI	Better Attributions Through Regions

deployment but also to gain a general understanding of the model's components and operation. Explainable AI is an active and quickly growing research area that needs more investigation and consensus [10].

The definition of model interpretability in the context of image analysis and understanding is not well-established [11]. Also, there is a discrepancy between researchers in defining the concept of explaining the motive of a prediction. Some studies define it as the ability of the model only to highlight the important regions or features that contribute to the output predictions. Others assess the model's interpretability by its ability to highlight the entire object of interest in an input image. Consequently, Lipton [11] argued that model interpretability might have different definitions that reflect different ideas or applications. For example, semantic segmentation can be argued as a visualisation approach to explaining a model's predictions because it assigns each pixel in an image to a specific class. Nevertheless, segmentation outputs focus on the whole object without revealing which parts of the image are relevant for the outcome. Consequently, the annotated output is not sufficient to justify the model's decision. On the other hand, input perturbation methods highlight only the important features or regions used by a model to support its decision. This study presents and discusses methods that follow both definitions for operation and output understanding.

We mainly focus on the visual explanation of pre-trained CNNs. Though, Cynthia [12] argued that some explanation methods do not provide enough evidence or details. It is also suggested that building inherently interpretable models is a better approach than explaining the model's decisions [12]. It is argued that self-explaining models, such as Self-Explaining Neural Networks (SENN) [13], can construct a highly complex and interpretable model without limiting the performance. This makes inherently interpretable systems more immune to adversarial noise. On the other hand, many post-hoc explanation methods are unstable because they produce different explanations for the same input when noise is introduced [13]. It is a significant challenge to persuade policymakers to define metrics and procedures to ensure the safety of complex non-self-explaining deep network models because some measurements and evaluation methods can be easily misinterpreted and exploited [12]. However, self-explaining systems are beyond the scope of this review.

Unlike Seifert et al. [14], Guidotti et al. [15], and Zhang et al. [16], whose surveys are extended to other analysis techniques such as confusion matrices, histograms, explanatory graphs [17], and decision trees for model analysis [18], this review is mainly focused on visualisation methods, because CNN visualisation is the direct way to explore network decisions and representations [16]. Also, unlike other surveys [19–21], which present trends, statistics, and prospective applications of XAI, we conducted a technical-oriented review with in-depth comparisons and evaluations. We present the methods which justify their predictions visually and disregard the reasoning methods that describe the process of how a CNN makes its decision, such as image dissection techniques [22]. Model-agnostic methods such as Shapley values [23,24] and Anchors [25] are beyond the scope of this review because they are well-covered by Molnar [26] and Samek et al. [27]. Also, model approximation methods are beyond the scope of this review as we are interested in the direct explanation of pre-trained models. Mainly, we focus on visualising heatmaps (saliency maps), reconstructed images (synthesized images), and hidden layers' features. The main task for the systems being visualised is image classifications using the architecture of CNNs.

First, we want to draw the reader's attention that terms like visualisation, explanation, and attribution methods are used interchangeably. Relevance maps, attribution maps, saliency maps, sensitivity maps, and activation heatmaps are used in different contexts to refer to the visual contribution of each feature to the overall prediction. 'Saliency' can have two meanings depending on the context. It either signifies the gradient approach or the sensitivity map.

Second, we organise the visualisation techniques into three main categories depending on which part of the CNN is being visualised (Fig. 1). We follow a different categorisation technique to the one presented by Grun et al. [28], at which the authors proposed a taxonomy for feature visualisation methods consisting of three main classes: Input Modification, Deconvolutional, and Input Reconstruction methods. Input Modification methods, such as Occlusion [29,30], modify the input by occluding patches and measure the resulting changes in the

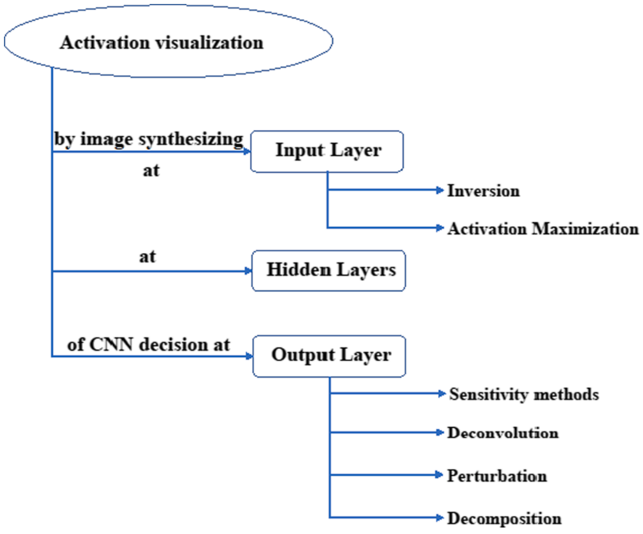


Fig. 1. Visualisation taxonomy.

output score [28]. Deconvolution methods [28] measure the contribution of a pixel in the input image by backpropagating its activation in the higher layer through the network until the input layer is reached. As the contribution of each pixel is measured, the group contribution can build up a visualisation map of features relevant to the object of interest [28]. Methods that goes under this class are DeconvNet [29], Backpropagation [31], Guided Backpropagation [32], Layer-wise Relevance Propagation (LRP) [33] and Class Activation Maps (CAM) [34]. Input Reconstruction methods [31,35,36] generate input images that can maximally activate a specific network's unit of interest.

Fig. 1 shows the proposed categorisation chart. Different visualisation methods are split based on the architecture position where the features are being visualised (input layer, hidden layer, or output layer). First, visualising feature maps at the input layer (equivalent to Input Reconstruction class [28]) by mathematically synthesizing images either by using Activation Maximisation (AM) to maximally activates a particular neuron or by using representations inversion. Second, visualising features and patterns learned by hidden layers. Last, visualising features that activate a network to make a decision with respect to the output class of interest or so-called post-hoc activation visualisation (visualise network decision). This includes gradients-based sensitivity analysis, decomposition techniques, and occlusion methods. The fashioned heatmaps from this section can be overlaid with the input image to reflect the salient features. Although some methods use the same approach to visualise hidden and output layers, we prefer to separate them into two different categories as the visualised features are different. Thus, the application and utilisation of the visualised features differ. The proposed categorisation organises the investigation process and suggests a possible framework for discussion.

Last, we present an in-depth analysis of state-of-the-art explanation methods, as many other techniques are built upon them. The paper's contribution can be summarised as follows: a technical review of different attribution methods focusing on post-hoc visualisation is presented in section 2. Applications and potential uses of visualisation techniques are highlighted in section 3. Different evaluation techniques (sanity checks) to assess the robustness of different explanation techniques are discussed in section 4. Finally, the review is concluded. Besides, the gaps and the future directions are highlighted in section 5.

2. Visualisation methods

In this section, the visualisation techniques of CNNs are explored. Subsection 2.1 (corresponding to activation visualisation at the input layer) presents AM, representation inversions, and combined approaches. Subsection 2.2 (corresponding to activation visualisation at the hidden layer) investigates features and patterns learned by hidden layers. Whereas perturbation, deconvolution, decomposition, and sensitivity analysis methods are discussed in subsection 2.3 (corresponding to activation visualisation at the output layer). For each primary method in subsection 2.3, network architecture, training details, conceptional approach, information extracted, pros, and cons are discussed. The generated heatmaps by these methods assign each pixel an importance value according to some function that depends on the output score.

2.1. Input image synthesizing 'Activation Maximisation'

Erhan et al. [37] constructed images that maximally activate a specific neuron using gradient ascent optimisation in the input image space. The AM problem is simplified to an optimisation problem at which an optimal input that can maximally activate a specific neuron using gradient ascent is sought. Starting with an initial input, the activation's gradients of the unit of interest is computed w.r.t the initialised input. Then gradient ascent is used to take steps in the input space to synthesize inputs that cause the highest activation for this unit (gradient ascent is also used by Nguyen et al. [38] for the same purpose). The process stops when an optimal input is obtained that can maximally stimulate this neuron. The optimal input can be displayed for interpretation and debugging purposes. Moreover, it helps to understand the nature of the functions learned by the network. However, as the produced input is mathematically synthesized, it looks artificial and far from natural images with high-frequency patterns and extreme pixel values in a random arrangement. The output is mainly scattered image parts that may represent what activates a particular unit.

The techniques of maximising the activation by modifying inputs can be applied to correctly classified images to manipulate them by some unrecognisable pixels' changes, to push the model to output different predictions (i.e. to deceive the network from classifying an object as class A to class B) [39]. Another approach is to mathematically produce unrecognisable images for humans that do not show any specific object. However, state-of-the-art CNNs still produce a high confidence score to a recognisable class which means some non-robust network architectures 'discriminative models' can be fooled [38]. The behaviour of discriminative models can be attributed to their linear nature and high-dimensional input space, while generative models are more robust to adversarial noise [40].

L2 regularisation can be used to numerically generate input images representing an output class of a model [31]. Generating such an image is similar to the backpropagation technique used to optimise layers' weights. However, in the case of image synthesizing, the trained weights are kept fixed, and optimisation is performed in the input space. The process starts with a zero-initialised image. After that, the mean of the training dataset images is added to the result. The optimisation process will continue until the optimal image that can maximally activate a specific unit is reached. This approach helps to reduce the effect of extreme pixels domination. Mitigating the impact of these pixels is beneficial as they are not useful for visualisation.

Gradient descent is used to optimise an objective function that inverts deep representations using image priors [36]. Image priors, such as total-variation normalisation, help to recover the statistics of low-level images. This information is useful for visualisation. However, the representation may remove them due to their non-usefulness for high-level tasks. Also, the technique helps to visualise the representation learned at each layer of a CNN. Mahendran et al. [41] extended their previous work [36] by introducing a unified formulation to visually investigate image features and CNNs. Visualisation of different representation types such as AM, inversion, and caricaturization are merged into a common framework. Thus, the visualisation problem is formulated as a regularised energy minimisation problem. The main aim is to produce natural-looking images by restricting image reconstruction to a set of natural images or so-called 'natural pre-images'. The analysis of CNN visualisation shows some interesting results, such as lower layers that contain representations of simple structures, e.g., lines and edges (local invariance). At the same time, deeper layers capture object-specific information and learn complex compositions.

Unlike [36], the inversion method learns image priors implicitly [42]. The method trains a CNN that invert a given feature vector into an expected pre-image using deconvolution networks. The trained network reconstructs images from the feature representation of different layers. It has been noticed that reconstructing from convolutional layers of AlexNet [1] produce high-resolution images. However, the quality degrades as representations of higher layers are being used, especially representations from Fully Connected layers (FC layers), which produce blurred images. Visualising representations from FC layers using the gradient descent approach [36] shows that they cannot preserve colours or locations. This contrasts with the trained network for inversion [42], which can retrieve some colour and location information from higher layers' representations. The inversion method [42] can be applied to non-differentiable features such as Local Binary Patterns (LBP) [43] and is significantly faster, in contrast to the gradient-based method [36].

Yosinski et al. [44] introduced new regularisation techniques that help to visualise the learned features. The regularisation operator is introduced to map an input to a regularised version of itself. The process starts from an initial value while taking gradient steps in the direction specified by the operator until the version of input that maximally activates a specific neuron is reached. Four regularisation techniques are used. Combined, they produce more effective results compared to individual utilisation. L2 and Gaussian blur regularisations are used to suppress high-frequency components and extreme pixel values, which are not useful for visualisation. At the same time, clipping pixels with small norms and small contributions to the output score helps to remove unnecessary values and only highlights the object of interest [44]. Many other image priors techniques are introduced to enhance the produced image quality, such as data-driven patches [45], jitter [41], initialisation from mean images [46], and centre-bias regularisation [46].

To produce more realistic visualisations, the DGN-AM technique [47] extended activation maximisation methods by introducing a deep generator network (DGN) that is trained as a prior to take a vector of scalars (feature representations) and produce a synthetic image. The synthetic image achieves two properties: it resembles real images from the ImageNet dataset [48], which means it is human interpretable (ImageNet is the same dataset used to train the CNN). Besides, it activates the neuron of interest. Experiments show that the produced synthetic images by DGN-AM reflect the learned features by neurons independently from priors (i.e., it shows neurons' prefers, not priors' prefers). Image synthesizing techniques are important in deep learning applications. They can be used to visualise features evolving during the training process, which may help to understand and debug DCNNs.

2.2. *Hidden layers feature visualisation*

Zhang et al. [49] proposed a method to modify a CNN to be more interpretable by training high convolutional layer filters to be able to represent a specific part of an object without any additional object-specific data annotation. High-layer filters in traditional CNNs can describe a mixture of patterns that might negatively impact the network interpretability. On the other hand, the proposed interpretable CNN pushes high layer filters to be more component-specific. This may help to identify object parts responsible for a specific prediction. The proposed method can be achieved by adding a loss for each filter's output to boost the filter towards a specific representation of an object part. The added loss helps to reduce the entropy of inter-category activations and spatial distributions of neural activations.

Zhou et al. [50] introduced a framework for network dissection that interprets the network's representations and quantifies their interpretability. The process involves three steps: identifying visual concepts in a dataset, measuring hidden units' response to the visual concepts, and quantifying alignments of hidden unit activations with visual concepts. The study also examines the impact of using different datasets and regularisation techniques [51,52] on the interpretability of a model. The introduced framework has some limitations, such as the inability to identify the contribution of joint units' that might represent one visual concept.

A software tool [44] is introduced to enable the visualisation of the channel's activations of convolutional layers in the same spatial layout as the input, where each filter is activated by a specific feature or pattern such as edges, faces, eyes, etc. Layers such as pooling and normalisation can be visualised using the proposed software, reflecting their impact on the model's behaviour. Real-time visualisation of all filters of a specific layer on one screen is a very informative approach as it shows the propagating data through a CNN.

Methods presented in [29,32,44,53] can be adapted to visualise the units of hidden layers. Filters in hidden layers are activated by patches or shapes captured by their receptive field. Krizhevsky et al. [1] directly visualised filters learned by the first layers to assess the learned features by a trained CNN. As multichannel layers are hard to visualise, Yu et al. [54] used dimension reduction (t-SNE) [55] to visualise patches in representation space constructed by filters of hidden layers. Besides, the DeconvNet approach [29] has been used to visualise the layer's information by reconstructing activations layer-by-layer successively until input space is reached (DeconvNet approach is described in detail in subsection 2.3). Also, image patches that maximally activates a filter in a hidden layer have been used by non-parametric methods to visualise that filter [56,57].

2.3. *Visualisation of output layer activations 'post-hoc visualisation'*

Explanation methods aim to define the contribution of each input feature to the output prediction. The output neuron associated with the correct prediction is the neuron of interest. The generated heatmap regarding the target object has red and blue regions corresponding to positive and negative evidence, respectively.

2.3.1. *DeconvNet [29]*

Conceptional Approach: Input patterns can cause a given activation in the feature maps. Deconcoultional networks are used by Zeiler et al. [58] to map the activations back to the input pixel space. The process can be explained as follows: an input image is presented to the CNN, whereas the features are computed through the networks' layers. To analyse a given activation, all other activations in that layer are set to zero. Then the feature maps are passed to the attached deconvolutional layer. Finally, the input pixel space is reached through successive unpooling, rectifying, and filtering operations to reconstruct the layer's activity.

To examine a convolutional network, a deconvolutional network is attached to its layers to show the input pattern that causes a given activation in the feature maps. The used approach can help to observe the features' progression during training and diagnose potential problems with the model. A disadvantage of this approach is that it can only visualise a single activation and not the joint activity presented in a layer.

Implementation details: Zeiler et.al [29] used a similar architecture to AlexNet [1] with some modifications. For instance, the sparse connections used in AlexNet layers 3, 4 and 5 are replaced by dense ones. For training, images are pre-processed by resizing and cropping. Furthermore, they are normalised by subtracting the per-pixel mean. Finally, ten different sub-crops of size 224 × 224 are used. Stochastic Gradient Descent (SGD) with 0.9 Momentum is used to update the model's parameters. The training dataset is divided into mini-batches of 128 images. The learning rate starts at 10^{-2} and then decreased manually throughout the training process when the validation error plateaus. All weights are initialised to 10^{-2} and biases to zero. Data augmentation is used with different flips and crops. After 70 epochs, training is stopped. The system takes around 12 days to be trained on a single GTX580 GPU. The proposed architecture outperforms AlexNet results on ImageNet dataset [48].

Visualisation and localisation: The proposed system has proved its efficiency to visualise feature activations using a deconvolutional network. Visualising a trained model can help to select better architectures. For example, by visualising the first and second layers of AlexNet architecture [1], it is noticed that the filters of the first layer are a mixture of high and low-frequency information. At the same time, the second layer visualisation shows aliasing artefacts caused by a large stride ($s = 4$) that is used in the first convolutional layer. A new architecture is proposed to overcome these problems by reducing the filter size in the first convolutional layer from 11 × 11 to 7 × 7 and reducing the stride to 2 instead of 4. Consequently, the new system retains more information in the first and second convolutional layers and achieves better accuracy.

Occlusion sensitivity is introduced to make sure that the object itself is the element that activates the network and not the context or the background. It also shows the ability of the model to locate the object in an image. This can be attained by occluding different portions of the input image with a grey square in a sliding window manner and monitoring the classifier's output. The system clearly shows its ability to localise an object within an image as the correct class probability has dropped significantly when the object of interest is occluded.

Discussion: The DeconvNet approach recalls the position of the max-pooling layers' values during a forward pass by storing these values in switches. The activations are then copied into the positions indicated by these switches during the deconvolutional process, while other low layer activations are set to zero. Switches are introduced as the max-pooling operation is non-invertible.

Unlike Regions with CNN features (R-CNN) [56], which can demonstrate visualisation by identifying patches within a dataset responsible for activations at the model's high layers, the occlusion approach is a top-down projection that can reveal structures within each patch that stimulate a particular feature. Results show that as the network becomes deep, it can learn powerful features [29]. Generally, high layers produce more discriminative features. Visualisation can be used to identify models' problems and obtain better results by selecting or modifying models' layers. Also, it provides insights into the sensitivity of the classification model to local structures and not to contexts.

A drawback of the DeconvNet method is that the image-specific information comes from max-pooling layers (switches). The absence of pooling layers will result in non-image-specific explanations. Besides, negative pieces of evidence are discarded during the backpropagation process due to the ReLU units, resulting in less informative heatmaps [59].

Similar to the Occlusion approach, Zintgraf et al. [53] proposed a method to evaluate the impact of removing information from an image. The introduced approach is based on prediction difference analysis [60] with some improvements. In prediction difference analysis, each input feature is assigned a relevance value according to its importance to the output prediction. A large prediction difference indicates a high contribution of the corresponding features to the output prediction and vice versa. Two enhancements are introduced [53]: conditional sampling and multivariate analysis. Compared to marginal sampling, conditional sampling can improve feature approximation by suppressing redundant, easily predicted pixels. Whereas a robust model should not be significantly affected by the deletion of one feature at a time (univariate approach), removing patches with several features (multivariate approach) should produce more descriptive relevance. The proposed improvements have produced more refined heatmaps that concentrate on the object of interest. Methods based on input features perturbation by occluding, manipulation or masking are significantly slow [53]. It needs several forward propagations through the network to compute the output score after every input perturbation. Moreover, the results are biased by the number of occluded features at each iteration determined by the sliding window size [61].

2.3.2. *Saliency map (Gradients) [31]*

Conceptional Approach: Gradients approach, also called backpropagation or saliency, visualise the derivatives calculated during the model's training. Still, saliency maps are computed after network training and not during the training process (i.e., the networks' weights are constant). Backpropagation is the process of increasing or decreasing the networks' weights to minimise the loss function during the training process [62]. Saliency maps return the spatial locations of the discriminative pixels of a particular class in an image. Class weights can be used to visualise the discriminative regions of an image that activates the network to produce a specific prediction. This is valid for linear class score functions, but as the class score of CNNs is a non-linear function of the input image, an approximation of the class score function can be estimated using first-order Taylor expansion [31]. In this case, the magnitude of the class score derivatives w.r.t the input image can compute image-specific class saliency maps. The magnitude of the derivatives indicates the most discriminative pixels of an image. Changing these pixels can have a great impact on the predictions. Consequently, these pixels represent the location of the object of interest in the image.

Saliency maps can be computed as follows: the class score derivatives are calculated w.r.t the input image through backpropagation. Then, the saliency map values are arranged in the same order as the input image pixels, i.e., m × n derivatives matrix will have the same indices as m × n input image pixels where m and n represent rows and columns of a greyscale image, respectively. Suppose the input is a multi-channel image such as RGB images. In that case, the maximum derivative magnitude is selected across all the channels to produce a single class saliency value for each pixel. Finally, the derivatives matrix is plotted to produce the saliency map.

Implementation details: The proposed system used a similar CNN architecture to AlexNet [1] but less wide with the following structure: five convolutional layers with 64, 256, 256, 256, 256 filters, respectively, followed by three FC layers with 4096, 4096, 1000 output neurons respectively. The network is trained on the ImageNet dataset with 1.2 M training images for 1000 classes [48]. Image jittering and zeroing-out random parts of an image are employed as regularisation techniques.

Visualisation and localisation: Saliency maps need one backpropagation pass to be produced. They can be considered a weakly-supervised approach for object localisation. It localises an object that mainly activates the network to make a prediction unless the network is cheating. No other types of annotations, such as bounding boxes or segmentation masks, guide the technique to localise an object. A proposed system based on saliency maps [63] has been used for the localisation task of ImageNet 2013 [48] and achieved a 46.4% top-5 error on the test set. The system computes the object segmentation mask using an image and its corresponding saliency map. Object segmentation mask is computed using GraphCut colour segmentation [64]. Colour segmentation is selected since the saliency maps may highlight only the most discriminative region of an image representing a part of an object and not the whole object. The process sets the foreground and background to follow the Gaussian Mixture Models [65]. The foreground is estimated from the pixels with saliency higher than the 95% threshold of saliency distribution in the image. At the same time, the background is estimated from pixels with less than a 30% threshold. The estimated foreground and background are then used to set the object segmentation mask to the largest connected component of the foreground pixels.

Discussion: Baehrens et al. [66] introduced the local gradients method to explain the predictions of machine learning classifiers. The proposed approach assigns a unique explanation to individual data points, unlike conventional feature extraction methods that extract the relevant global features for all data points. The gradients method [31], on the other hand, is applied to CNNs. It highlights an object located in an image using the target object's score derivative w.r.t the input image. The weakly-supervised approach for localisation, using the proposed Gradient method, beats the author's previous submission to ImageNet 2012 [48] using a fully supervised model [67] and Fisher vector feature encoding [68] by achieving a 50% localisation error.

The backpropagation 'saliency' approach can be considered a generalisation of the DeconvNet approach [29] as it can be used to visualise any layers' features and not only the convolutional ones. FC layer neurons are visualised in this paper [31] using the Gradient approach. DeconvNet is equivalent to the gradient approach through a CNN except for the backpropagation through the ReLU layers.

Although Gradient heatmaps are computationally faster than Occlusion as it only needs one backward propagation through the network, they do not fully explain the output prediction. The calculated map measures pixels change that would make an image belong to a specific category. However, it does not explain the classifier decision as argued by [59] or the direct relation to the variation of the output [61,69].

2.3.3. Gradient related approaches

Many approaches based on Gradients (eq. (1)) are proposed, such as element-wise products of gradients and input (GI) [70] (eq. (2)), Integrated Gradients (IG) [71] (eq. (3)), Smooth Gradients (SmoothGrad) [72] (eq. (4)), etc. Gradients of the output score are calculated w.r.t input and then multiplied with the input to enhance heatmap resolution [70]. Moreover, GI can be used to address the gradient saturation problem [70]. Although this technique can visually enhance the produced maps, this may be attributed to the original image's quality rather than the visualisation technique [73].

$$\frac{\partial Y^c(x)}{\partial x} \tag{1}$$

$$x \odot \frac{\partial Y^c(x)}{\partial x} \tag{2}$$

$$(x-\bar{x})\int_{\alpha=0}^{1}\frac{\partial Y^c(\bar{x}+\alpha(x-\bar{x}))}{\partial x}d\alpha \tag{3}$$

$$\frac{1}{n}\sum_{1}^{n}M_c(x+\mathscr{G}(0,\sigma^2)) \tag{4}$$

$\bar{x}$: *Baseline*

x : *Input*

Y^c : *Output prediction for class c*

n : *Number of samples*

M_c : *Class activation map for class c*

$\mathscr{G}(0,\sigma^2)$: *Gaussian noise with standard deviation* σ

Full-Gradient (FullGrad) [74] approach expands the Gradient method [31] by aggregating the information obtained from GI [70] and the gradients of the intermediate layers of a CNN. Aggregating maps from many layers produces sharp heatmaps as neuron-wise maps can independently support each spatial location's importance [74]. However, FullGrad can only use the maps of convolutional layers as they can preserve the spatial locations. Similarly, CAMERAS [75] produces high-resolution saliency maps by accumulating and fusing multi-scale activation maps and backpropagated gradients.

The Integrated Gradients [71] approach accumulates gradients over scaled-up versions of the input that follow a baseline defined by the user, i.e. it integrates the gradients of all points that fall on the straight-line path from the baseline to the input. Based on IG, XRAI region-based attribution method [76] is introduced to enhance IG's performance. Firstly, XRAI segments the input image using different sets of parameters to many overlapping regions. Then, using IG with black and white baselines, the importance of each region is tested. Finally, regions are combined into a large segment based on their relevance score. Using segmentation, XRAI can outperform gradient-based methods as it can identify relevant regions and discard others. Moreover, it can detect many instances of the same class in a given input image. Also, it can measure the smallest sufficient region that positively contributes to the output prediction.

The Smooth Gradients [72] approach uses added noise to enhance heatmap sharpness by averaging the explanations of noisy copies of the input. As Gradient sensitivity maps tend to be noisy due to the noisy gradients, SmoothGrad reduces visual noise by sampling similar images with added noise and then averaging the resulting sensitivity maps. Two hyper-parameters can be adjusted for SmoothGrad: the noise level (determined by the standard deviation) and the number of samples (equation (4)). SmoothGrad shows better visual coherence (highlights object of interest) and discrimination (highlights which class in a multi-class image is responsible for the prediction) compared to vanilla Gradient, Integrated Gradients, and Guided Backpropagation [72]. Inspired by SmoothGrad, Bykov et al. [77] propose NoiseGrad and FusionGrad. Instead of adding noise in the input space like SmoothGrad, NoiseGrad introduces stochasticity in the weight parameter space, resulting in a perturbated decision boundary. In other words, SmoothGrad produces a heatmap using multiple noisy versions of the input, whereas NoiseGrad uses multiple versions of the model. SmoothGrad and NoiseGrad are combined to produce FusionGrad [77] by incorporating both stochasticity in the input and model spaces to gain the benefits of both techniques.

2.3.4. Decomposition related approaches

Layer-wise Relevance Propagation (LRP) [33,78] uses backpropagation to compute relevance. LRP can be seen as a biased gradient towards positive values [27]. It is a generalised approach to visualise the contributions of non-linear classifiers by a pixel-wise decomposition of each pixel's output prediction. Starting from the output layer, the algorithm assigns a relevance (importance score) to the target neuron equal to the neuron's output. At the same time, the relevances of all other neurons are set to zero. Recursively, the LRP technique redistributes relevance over layers' neurons according to some rules (ϵ-rule, for instance) until it reaches the input layer, where the attribution can be identified [33] (Fig. 2). An example of LRP maps is shown in Fig. 3.

LRP can miss-attribute input regions to the relevance as it only considers the target class in the relevance calculations [79]. Consequently, Contrastive Layer-wise Relevance Propagation (CLRP) is introduced to enhance the discriminative ability of LRP by subtracting the relevance for non-target classes from the relevance propagation [79]. This boosts the contribution of the target class and suppresses the contribution of other classes. However, equally penalising non-target class may cause wrong attributions because of the equal weighting of non-target nodes. Softmax-Gradient Layer-wise Relevance Propagation (SGLRP) [80] is proposed to overcome this problem. SGLRP uses the gradients of the softmax layer w.r.t the intermediate value of each output node to subtract the relevance from the non-target classes. Using the softmax layer gradients as the initial relevance from the output layer while backpropagating creates an LRP model where the propagating values are the probabilities of all classes, and the highest is the target class [80]. A great advantage of the SGLRP [80] approach is that it removes relevance corresponding to non-target class with high probability compared to the low probability non-target classes, unlike vanilla LRP [33], which ignore non-target classes, and CLRP [79], which penalise non-target classes equally.

Deep LIFT [81] is an improved version of LRP [33]. Like LRP, it decomposes the output score while backpropagating through the model

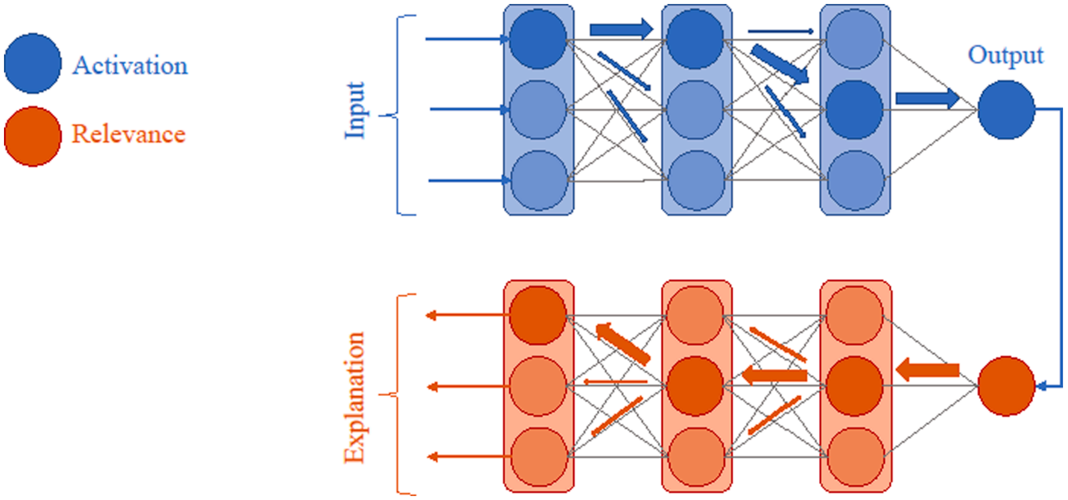


Fig. 2. LRP relevance redistribution technique (reproduced from [33]).

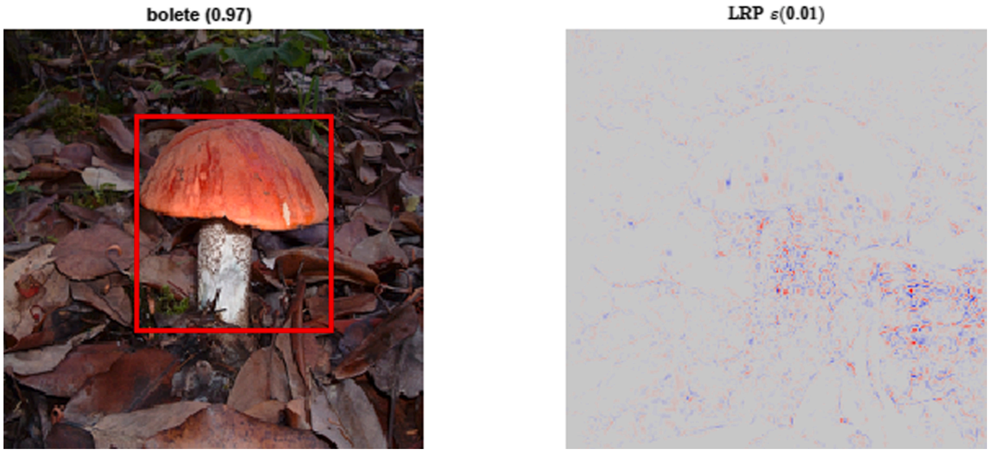


Fig. 3. An example of LRP map.

until input space is reached. However, DeepLIFT defines a reference point in the input space. Then, relevance is assigned according to the relative change (difference) in activations at the original input compared to the reference point [81]. Deep Taylor Decomposition (DTD) extends LRP using first-order Taylor expansion around a root point to decompose a neuron's activation in terms of contribution from its inputs [82]. Choosing the root point for DTD is challenging as many options are available. PatternAttribution extended DTD to solve this problem by learning the root point from the data (class 'signal' estimator is trained on the first half of the ImageNet training dataset) [69]. PatternAttribution acts as a root point estimator for DTD. PatternAttribution can be visualised as the neuron contribution of the estimated class to the output score. Using baseline (reference point) in decomposition approaches helps to overcome zero and saturate gradient problems in the gradient-based sensitivity analysis. However, decomposition techniques cannot satisfy the chain-rule property inherited in the gradient-based analysis [71]. Generally, sensitivity analysis measures local effects while decomposition measures global ones.

SHapley Additive exPlanation (SHAP) is a unified framework for interpreting predictions [23]. The proposed framework attempts to relate different methods that are based on assigning each feature an importance value. In addition, it helps to pick the best explanation method for a specific application.

2.3.5. Guided backpropagation (GBP) [32]

Conceptional Approach: The DeconvNet method proposed by Zeiler et al. [29] is used to analyse the All-CNN and to visualise the most discriminative regions of an image that contributes to the network's prediction. The DeconvNet approach is also used to investigate the impact of removing pooling layers. It is noticed that the DeconvNet approach fails to produce a reasonable explanation for the concepts learned by the networks' high layers. This behaviour is attributed to the absence of pooling layers. The DeconvNet approach relies on switches (position of maximum values as the max-pooling operation is a non-invertible operation) computed during a forward pass. Switches are then passed to the deconvolutional layers to reconstruct the image with the most discriminative regions. Without pooling layers, therefore, without switches, the DeconvNet approach will not be able to reconstruct the image.

All-CNN does not have any pooling layers. Consequently, the DeconvNet approach can be applied in the low layers without any need for switches. These layers learn general features such as Gabor filters. On the other hand, the DeconvNet method [29] fails to visualise high layers activations that learn more invariant representation.

Sprinenberg et al. [32] proposed an alternative way for visualisation by computing the activations' gradients w.r.t the input image through backpropagation. The main difference between DeconvNet and backpropagation approaches is how the backpropagation is handled through the ReLU units [31]. The backpropagation approach [31] is equivalent to the DeconvNet one except for gradients through ReLUs, which are computed based only on the top gradient values, and the bottom input is ignored. DeconvNet approach [29], which zeros negative values of top gradients, and backpropagation [31], which zeros negative values from bottom inputs, are then combined to produce Guided Backpropagation [32] which zeros both negative values. The signal from high layers guides the backpropagation; hence the name is derived. It works as the switches in the DeconvNet approach [29]. Doing so prevents negative gradients from flowing back, which can undesirably impact the visualisation.

Implementation details: Guided backpropagation uses only convolutional layers in its architecture. The proposed architecture investigates the simplest architecture based uniquely on convolutional layers. This architecture is intended to identify what specific components in a typical CNN is crucial to achieving state-of-the-art performance on deep learning tasks such as object recognition.

To achieve this, pooling layers, in a typical CNN used for classification, are replaced by convolutional ones with a stride equal to two. Convolutional layers with small filters are used to reduce the number of parameters (for example, 3 × 3 filter size). Lastly, FC layers are replaced by 1 × 1 convolutional layers with fewer parameters than FC ones [83].

Convolutional layers can compensate for pooling ones by removing pooling layers and increasing the stride of the convolutional layer before it. Besides, the pooling layer itself will be replaced by a convolutional one with a stride larger than one [32]. Increasing the stride of the convolutional layers can reduce the overlap between filters, which can negatively impact the network's accuracy. Also, replacing pooling layers with convolutional ones can increase network parameters. Consequently, the architecture is abstracted to only convolutional layers with subsampling, ReLU, Global Average Pooling (GAP), and softmax layers for output predictions.

Three datasets are used to evaluate the performance of the proposed models: CIFAR-10, CIFAR-100 [84,85], and ImageNet [48]. However, the focus is on CIFAR-10, as the training time is shorter than other datasets.

Detailed training parameters are as follows: SGD with 0.9 momentum has been used as the optimisation algorithm. The learning rate is multiplied by 0.1 when training epochs reaches 200, 250, and 300. Proposed systems (Strided-CNN, ConvPool-CNN, and All-CNN) are trained for 350 epochs. Strided-CNN removed pooling layers and increased the stride of the preceding convolutional layer. ConvPool-CNN kept the pooling layer but added a convolutional layer before it. All-CNN replaced the pooling layer with a convolutional one. Dropout [51,86] has been used as a regularisation technique. It is applied to the input layer with a 20% dropout probability and the newly introduced layers with a 50% probability. Also, a weight decay of 0.001 is introduced for further regularisation. Besides, data augmentation techniques are applied such as image flipping and random translation.

Visualisation: Guided Backpropagation provides high-resolution and clear activation maps compared to the DeconvNet approach on All-CNN. It can be used to visualise the intermediate and the output layers of the proposed network with or without switches. However, DeconvNet fails to produce clear activation maps on the All-CNN as it needs switches for deconvolution and un-pooling computations.

Discussion: The proposed simple architecture (All-CNN) has achieved state-of-the-art performance without complex design, normalisation, or pooling. All-CNN stabilises the performance with some improvements compared to the base model (that has pooling and FC layers). It can be concluded that pooling layers are not vital for CNNs, as removing them does not hurt the performance [32].

Guided Backpropagation and Occlusion approaches produce high-resolution maps, but their localisation ability is very poor compared to CAM [34] and Grad-CAM [87]. Guided Backpropagation's output is the fine-grained details of the features that activate a network to make a specific decision. In comparison, CAM and Grad-CAM are more region-based approaches.

Similar to Guided Backpropagation, DeSaliNet [88] combines both advantages of DeconvNet, which can accurately reproduce image boundaries, and the saliency method, which can localise objects efficiently. It can be noticed that DeconvNet [29], Backpropagation [31], and Guided backpropagation [32] use almost the same steps to produce the visualisation maps, although they are described in different ways. The main difference is in how they handle the gradients through ReLU layers. DeconvNet allows only positive derivatives to backpropagate (i.e., applying ReLU operation to the gradients). Backpropagation passes only the positive elements corresponding to the preceding feature map (from the lower layer). Guided backpropagation combines both techniques. Fig. 4 depicts the difference [32]. A qualitative comparison between the three gradient-based methods is shown in Fig. 5.

Kindermans et al. [69] proved theoretically using a linear model which mimics the simplest CNN that DeconvNet, Guided Backpropagation, and LRP do not produce the correct explanation. The limitations of these methods on a linear model hold for non-linear models. PatternNet and PatternAttribution [69] are introduced to tackle this problem. They produce qualitatively improved signal visualisation and attributions. PatternNet, as a signal estimator, visualises the explanation using the original colour channels, while PatternAttribution is visualised as a heatmap of pixel-wise contributions. Signal (class) visualisation and attribution can be attained using a signal estimator learned from data that is optimised to remove most of the information in the residuals (input minus estimator signals) [69]. PatternNet projects the estimated signal back to the input space. A process similar to gradients computation while backpropagation, but the network weights are replaced by the guiding directions determined by the signal estimator. Kindermans et al. [69] argued that the implicit signal estimator of DeconvNet, Guided Backpropagation, LRP, and DTD could not capture the true object (signal) in the input as the gradients do not provide an estimate for the signal in the data. In contrast, PatternNet can recover the signal effectively thanks to the optimised signal estimator [69].

Visually, Guided backpropagation seems to have an advantage over Backpropagation (Gradients) and DeconvNet approaches as it generates more human-interpretable visualisation (Fig. 5). Experimentally, this is not true. Guided Backpropagation is less class-sensitive than saliency maps (Gradients approach). In fact, Guided backpropagation acts as a simple edge detector. More details on visualisation techniques sanity checks are provided in section 4.

2.3.6. *Class activation map [34]*

Conceptional Approach: The term Class Activation Map (CAM) has been used to refer to the weighted activation maps generated for an image. GAP layer is introduced to generate accurate discriminative localisation. Though GAP is not a novel technique, its utilisation to produce heatmaps is a major contribution [34]. The intuition behind using GAP is that it helps the network to identify the whole area of the object [34], unlike global max pooling, where the localisation is limited to a point lying on the object's boundary [89].

CAM technique displays a heatmap representation that highlights image pixels which trigger the CNN to categorise an image to a specific class. Primarily, the approach maps the predicted class score back to the previous convolutional layer. GAP layer outputs the spatial average of the feature map of each unit at the last convolutional layer. A weighted sum of these values is used to generate the final output. The process can be summarized as follows: after the last convolutional layer of a typical CNN, the GAP layer takes the convolutional layer channels as an input and return their average as an output. Each output per category is assigned a weight. Then, a heatmap is generated per class output, and the weighted sum is calculated for all the heatmaps. Finally, the CAM is up-sampled to the image input size. The generation process of the heatmaps using the CAM technique is depicted in Fig. 6.

Implementation details: The proposed architecture resembles Network-in-Network [83] and GoogleNet [90] with mainly convolutional layers. GAP is performed on the convolutional feature maps

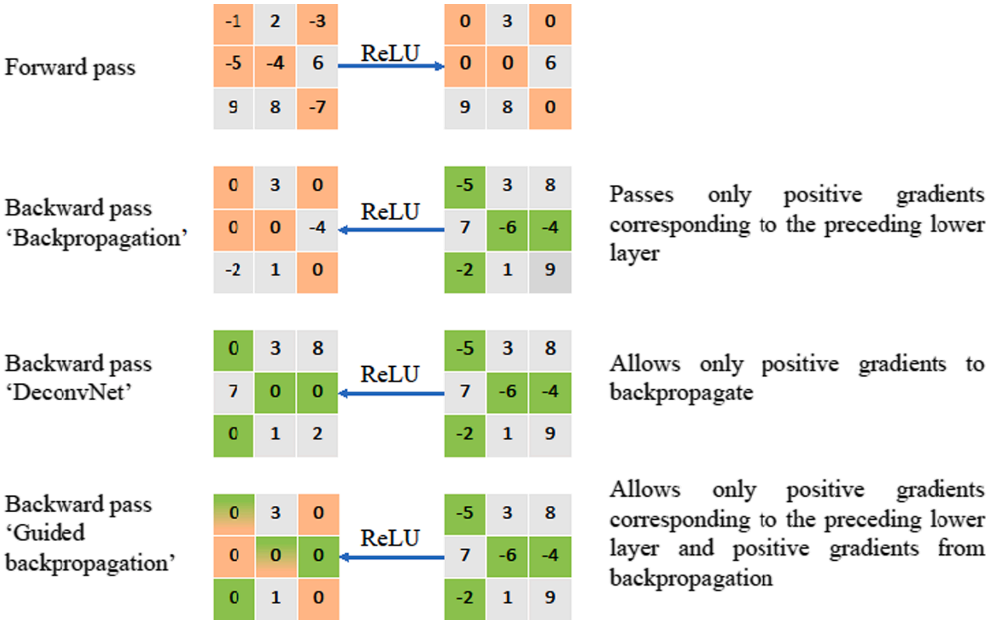


Fig. 4. Main differences between backpropagation, DeconvNet and Guided backpropagation approaches (reproduced from [32]).

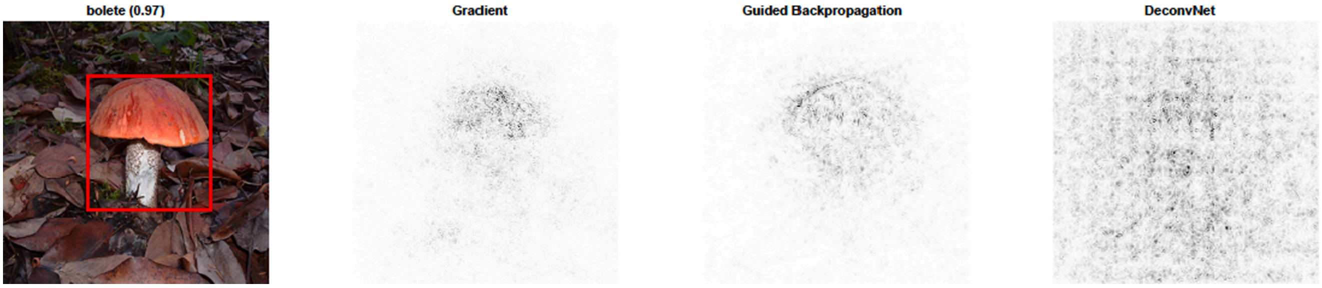


Fig. 5. Examples of gradient-based methods.

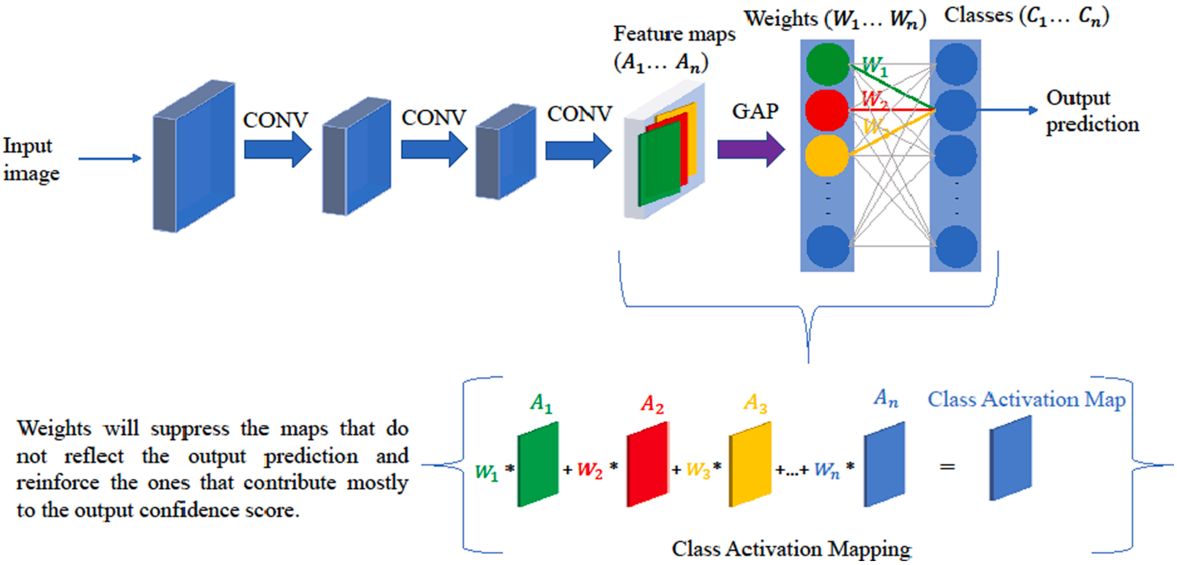

Input image
CONV
CONV
CONV
Feature maps (A_1… A_n)
GAP
Weights (W_1… W_n)
Classes (C_1… C_n)
W_1
W_2
W_3
Output prediction
Weights will suppress the maps that do not reflect the output prediction and reinforce the ones that contribute mostly to the output confidence score.
A_1
A_2
A_3
A_n
Class Activation Map
W_1 * + W_2 * + W_3 * +…+ W_n * =
Class Activation Mapping

Fig. 6. Class Activation Map (CAM) generation process (reproduced from [34]).

before the final output. Zhou et al. [34] evaluate the CAM technique on AlexNet, VGGNet, and GoogleNet. FC layers are replaced by GAP and softmax layers for class scoring. The number of learnable parameters significantly decreases by removing the FC layers with a side effect of a drop in the network's accuracy. It is found that increasing the spatial resolution of the last convolutional layer before the GAP layer results in improvements in the localisation ability [34]. Layers after conv5, conv5-3, and inception4e are removed from AlexNet, VGGNet, and GoogleNet, respectively. This results in a high spatial resolution for the last convolution layers of 13 × 13, 14 × 14, and 14 × 14, respectively, which improves the network's localisation ability [34]. Finally, a convolutional layer with 3 × 3 filter size, 1024 channels, one stride, and one padding followed by a GAP and a softmax layers are added to the mentioned architectures. Each proposed architecture is trained on the ImageNet dataset [48]. The details of the training options are not mentioned in the paper [34]. Similar results are obtained when these systems are trained from scratch or fine-tuning the newly added layers [34].

Visualisation and localisation: CAM can be seen as a weighted linear sum of specific visual patterns that activate some network units at different spatial locations. The proposed GAP-CNN network learned generic features similar to FC layers in AlexNet and VGGNet. Besides, it can identify discriminative image regions. Though, it has not been trained on a localisation task. In addition to heatmaps, CAM can be used to visualise class-specific units. Convolutional units act as visual concept detectors [30] that can identify low-level features such as edges and high-level features such as objects and compositions. A combination of individual class-specific units can guide the CNN to output predictions. Visualising this combination, besides heatmaps, gives insights into understanding CNN's behaviour and its approach to classifying an image. At the same time, it is challenging to track each unit contribution in FC layer networks, which can explain the intuition of using GAP.

Discussion: Many architectures tend to avoid FC layers to minimise the number of trainable parameters while maintaining high performance, such as SqueezNet [91], GoogleNet [90], ResNet [3] and MobileNet [92]. GAP layer is introduced as a regularisation technique to avoid overfitting [83]. It is found that the GAP layers can also enable CNNs to have localisation capabilities [30,34]. The proposed system has proved its efficiency by achieving top-5 errors for object localisation on ImageNet [48].

Both Occlusion maps [29] and CAM [34] analyse only convolutional layers. They ignore FC ones if they exist, which means some of the intuition behind the predictions are missing. Modifying CNN architectures to have a GAP layer then retraining the model is a limitation of the CAM technique. Also, CAM is constrained to visualise the final layer's heatmap and cannot be applied to visualise the middle layers.

2.3.7. *CAM-related approaches*

Many proposed studies for weakly-supervised object localisation are based on the success of CAM. Since CAM does not localise the entire object but rather a specific region that strongly contributes to the network's prediction, the Adversarial Complementary Learning (ACoL) [93] approach is introduced. Using weakly-supervised end-to-end training, ACoL can discover and localise the entire object of interest in an image by using an additional classifier for complementary object regions. Motivated by adversarial erasing [94], two classifiers are used. The first one is used to identify the most discriminative regions, which are then erased from the feature map. The erased feature maps are fed to the other classifier to extract new complementary object-related regions.

The strategy of the Hide-and-Seek [95] approach during training is to hide random patches in the training images. This prompts the network to search for other regions in an image that contributes to the network decision in the absence of the most discriminative regions. However, hiding patches randomly without any supervision might not help the network to discover new regions.

Zhang et al. [95] proposed a learning process called Self-Produced Guidance (SPG) to separate the foreground, mainly the object of interest, from the background to generate better visualisation and precise localisation of objects. SPG uses a classification network to generate an attention map (feature activation map) where the highlighted pixels represent the foreground, the low confident score regions represent the background, and the medium confidence areas remain undefined. Intermediate features are used to assign these undefined pixels to either the foreground or the background regions during the iteration process, using the upper layer's output as supervision for the lower layer to learn better object localisation. The foreground and background guidance masks are then used as supportive supervision to enable the network to learn better relations between pixels. Consequently, better visualisation maps can be attained.

Common Component Activation Map (CCAM) [96] uses CAMs as components instead of class-specific maps to localise unseen or unknown objects. Localising common objects of the same class among a set of images 'co-localising' is different from weakly-supervised object localisation as it is not limited to the predefined object categories. In CCAM, the output of the last FC layer of a typical convolutional network is used as an input object component vector instead of a categorical probability output. The average component vector is computed for a group of images to find the group common vector. The largest components from the group common vector are selected. Lastly, a weighted sum of the feature maps of the last convolutional layer is computed for each image to get the common component activation map according to the top components.

CLass-Enhanced Attentive Response (CLEAR) [97] is a multi-factor visualisation approach. It allows the visualisation of regions of interest that mainly contribute to the network's decisions and the predominant classes associated with these regions. It alleviates the ambiguity produced by binary-based heatmap approaches such as [33,82] by producing class-based heatmaps that are more readable and understandable. Binary-based heatmaps produce output that highlights positive and negative regions. Whereas CLEAR visualises the regions that contribute to the network predictions, besides to which classes are these regions belong.

CLEAR approach and architecture are similar to CAM [34] and Guided Backpropagation [32]. The process of CLEAR is as follows: activation maps are computed for each class of the last convolutional layer of the network. Two different types of maps are extracted from these activation maps, the predominant class activation map, which shows the highly contributed class for the network's prediction at each location, and the dominant response map, which shows the activation level for each location. Finally, they are combined to produce a CLEAR map. The architecture of CLEAR is built using only convolutional layers. The last convolutional layer of the network has a number of channels equal to the number of classes predicated by the network, which is then fed into a GAP layer then a softmax layer for output probabilities.

As CLEAR uses different colours to distinguish between different classes in class-based heatmaps, it is unfeasible to use this approach for more than ten classes, as shown in their paper [97]. Using different colours for big datasets such as ImageNet, which has more than 1000 classes, would be chaotic, making it difficult to visualise and interpret the decision outputs. This limits CLEAR applications for large datasets. Like CAM [34] and Guided Backpropagation [32], the proposed approach is applied only on fully convolutional networks. So, neural network modifications are applied to use VGG-16 [2], where the FC layers are replaced by convolutional ones that are fine-tuned on the training dataset.

2.3.8. *Gradient-weighted class activation map [87]*

Conceptional Approach: Unlike FC layers, convolutional layers can preserve spatial information. Although Grad-CAM is a general technique that can be applied to any layer of a CNN to examine its activations, this work [87] only focuses on explaining the output layer's decisions as its neurons can identify parts specific to the target object. Grad-CAM uses the gradient information passed to the last convolutional layer of a CNN to assign importance weights to each neuron for a specific decision 'class' of interest. The main difference between CAM [34] and Grad-CAM [87] is in the way of generating the weights for the feature maps. In CAM, heatmaps are generated by computing the weighted average sum of the activations of the last convolutional layer using the

FC layer’s weights. Whereas in Grad-CAM, the gradients of any layer are used to generate these weights.

The Grad-CAM approach can be summarised as follows: first, gradients of the score for a specific class are computed w.r.t. feature map activations of a convolutional layer. Then, the computed gradients are averaged to obtain the weights for each feature map. Finally, the forward activation maps are weighted and combined, followed by a ReLU operation (Fig. 7 grey shaded). ReLU is used to highlight the contributing features to the class of interest. Negative impact pixels usually belong to a different class. Consequently, they need to be suppressed using a ReLU function to obtain better localisation. The final result is a coarse heatmap of the same size as the final convolutional layer feature map.

Implementation details: Grad-CAM can be applied to any CNN architecture. Unlike its ancestor (CAM) [34], there are no restrictions on using specific layers such as GAP, and there is no need to retrain the whole system to adapt to the Grad-CAM approach. This means it can be applied to off-the-shelf CNN based architectures. It can be applied to CNNs with FC layers, CNNs with multimodal inputs, and reinforcement learning without architecture modification as it uses the gradients of any target class. Grad-CAM does not trade off architecture complexity or accuracy with interpretability. Thus, it can be applied to very deep architecture, such as ResNet [3].

Visualisation and localisation: Grad-CAM is a localisation technique that can produce a visual explanation for any CNN. To evaluate Grad-CAM localisation, pre-trained VGG-16 [2], AlexNet [1], and GoogleNet [90] have been used. Ramprasaath et al. [87] assume a model should achieve two factors to produce a high-quality visual explanation on a classification task. First, it should produce a class-discriminative output that localises a specific object in an image. Second, a high-resolution map can be attained with fine-grained details.

For visualisation evaluations, human studies and experiments have been conducted to understand the trade-off between interpretability and fidelity of Grad-CAM to model predictions. The main purpose of these studies is to show that Grad-CAM produces better quantitative and qualitative results than previous approaches. Besides, an end-user can trust the visualised model. Guided Grad-CAM, a combination between Guided backpropagation [32] and Grad-CAM [87], shows that it can help to improve human performance to better identify the object of interest (more class-discriminative) compared to Guided backpropagation [32]. They (humans) can also identify which model provides better results based on the produced visual explanations, which may help to build more trust in the model. Since VGG-16 [2] is better than AlexNet [1] in terms of accuracy on the PASCAL classification task [98], visualisation results show that humans can identify the more reliable model from prediction explanation, despite both models making the same predictions.

Grad-CAM shows a reasonable explanation for failure modes. Also, it is robust to adversarial noise. Adversarial examples can be created by optimising the input to maximise the prediction error [39]. Moreover, Grad-CAM can be used to identify and reduce biases in datasets.

Discussion: Grad-CAM technique shows that incorrect and unreasonable predictions can often have reasonable explanations. It can be used to identify dataset bias which can help to achieve model generalisation. Results show that it outperforms previous methods on weakly-supervised localisation tasks. It also helps users to distinguish between strong and weak deep neural networks, even if they produce the same predictions. Grad-CAM can be considered a generalisation of CAM, or CAM is a special case of Grad-CAM. Examples of CAM and Grad-CAM heatmaps are shown in Fig. 8.

On the other hand, Grad-CAM cannot highlight fine-grained details. Pixel space gradient-based visualisation methods such as Guided backpropagation [32] produce high-resolution visualisations than Grad-CAM [87]. To counter this problem Guided Grad-CAM technique is introduced. Grad-CAM and Guided backpropagation techniques are combined by point-wise multiplication to produce high-resolution (fine-grained pixel class-specific) and class discriminative (the class region is highlighted) maps. Guided Grad-Cam approach can be seen in Fig. 7.

The incapability of localising multiple occurrences of an object in an image can be considered a disadvantage of the Grad-CAM technique. Also, the partial derivatives hypothesis can cause inaccurate localisation.

Grad-CAM is used in Guided Attention Inference Networks (GAIN) [99] to generate online attention maps for training a network for the task of interest. Attention maps can be used as priors in weakly-supervised localisation and segmentation tasks. Although Grad-CAM can highlight the most discriminative regions in an image that contribute to the network’s prediction, GAIN is introduced to supervise the attention maps while training a network to produce complete and accurate maps. The produced map can cover the complete object of interest and enhance the overall performance of the system. This can be achieved using the two streams network of GAIN: the classification stream, which finds the regions that help to recognise the object of interest and the attention mining stream, which includes all the regions that contribute to the prediction in the attention map. Both streams share the same network parameters. The approach helps the network to

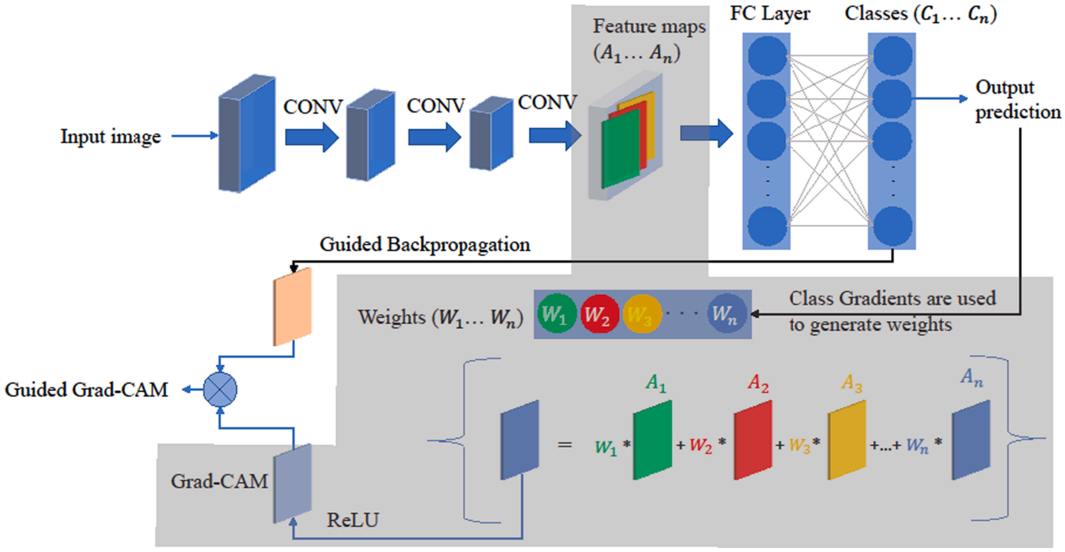

Input image
CONV
CONV
CONV
Feature maps ($A_1 \dots A_n$)
FC Layer
Classes ($C_1 \dots C_n$)
Output prediction
Guided Backpropagation
Weights ($W_1 \dots W_n$)
W_1
W_2
W_3
W_n
Class Gradients are used to generate weights
Guided Grad-CAM
A_1
A_2
A_3
A_n
$= W_1 * + W_2 * + W_3 * + \dots + W_n *$
Grad-CAM
ReLU

Fig. 7. Grad-CAM (grey-shaded) and Guided Grad-CAM approaches (reproduced from [87]).

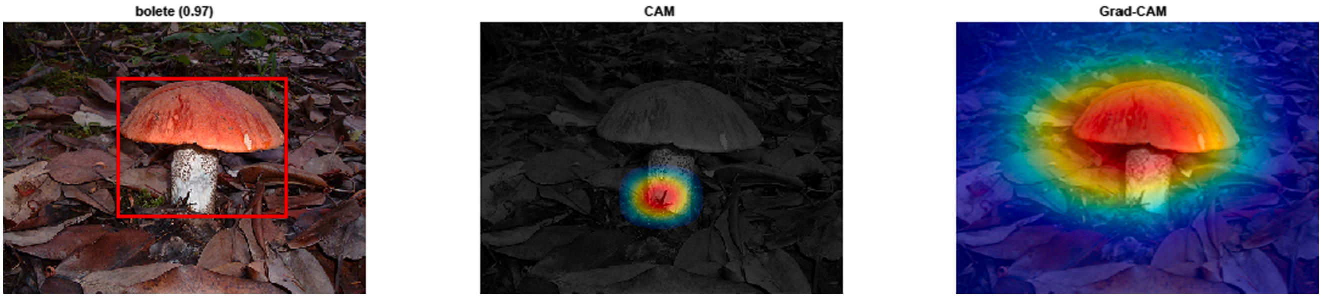


Fig. 8. Examples of CAM and Grad-CAM heatmaps.

extend its focus not only to the most discriminative regions of the input image but also to other contributing regions. The proposed approach boosted the system performance and achieved the best performance on the PASCAL VOC 2012 segmentation task [98].

2.3.9. *Grad-CAM related approaches* [87]

Grad-CAM++ [100] is introduced to alleviate two problems in Grad-CAM: the inability to localise and visualise multiple occurrences of the same class in an image and the failure to localise the entire area of the object. This improvement has been achieved by rearranging the convolutional neurons' importance weights equation of Grad-CAM and introducing a weighted average for the gradients.

$$w_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \tag{5}$$

$$M_{Grad-CAM}^c = ReLU(\sum_k w_k^c A^k) \tag{6}$$

$$w_k^c = \sum_i\sum_j \alpha_{ij}^{kc}.ReLU(\frac{\partial Y^c}{\partial A_{ij}^k}) \tag{7}$$

$$M_{Grad-CAM++}^c = \sum_k w_k^c A^k \tag{8}$$

$$w_k^c = \sum_i\sum_j \alpha_{ij}^{kc}.ReLU(\frac{1}{n}\sum_1^n D_1^k) \tag{9}$$

$$M_{SmoothGrad-CAM++}^c = \sum_k w_k^c A^k \tag{10}$$

$\frac{\partial Y^c}{\partial A^k}$: Gradient

Y^c : Class c score

A^k : Activation feature of channel k

w_k^c : Neuron importance weight

Z : Number of pixels in the activation map

α_{ij}^{kc} : Weighted average coefficients of the pixel-wise gradients

$M_{Grad-CAM}^c$: Grad-CAM heatmap

$M_{Grad-CAM++}^c$: $Grad - CAM + +$ heatmap

D_1^k : 1st derivative of the k$^{\text{th}}$ feature map

Grad-CAM [87] computes the gradients $\frac{\partial Y^c}{\partial A^k}$ of the score Y^c of class c with respect to feature activation maps A^k. To obtain the neuron importance weights w_k^c of feature map k for a target class c as in (eq. (5)), these gradients are global average pooled over the dimensions of the image indexed by i,j (Z represent the number of pixels in the activation map). Finally, feature maps are weighted and combined, followed by a ReLU operation to produce the coarse heatmap visualisation $M_{Grad-CAM}^c$ as in (eq. (6)).

Grad-CAM++ [100] proposed a generalisation to Grad-CAM to overcome its limitations. First, it reformulates (eq. (5)) by moving the ReLU operation to the neuron importance weights equation (eq. (7)). Second and most importantly, it has introduced weighted average coefficients of the pixel-wise gradients α_{ij}^{kc}. The introduced term helps to highlight the existences of objects in all feature maps with equal importance. ReLU is used to highlight positive gradients (importance in feature maps) and to ignore negative ones similar to Grad-CAM [87]. Unlike Grad-CAM, Grad-CAM++ has a constraint for the class score. It assumes that a particular class's score must be a smooth function, such as exponential or softmax functions. Discriminative localisation can be obtained using (eq. (8)). Grad-CAM++ produces heatmaps of all regions. This is beneficial with scattered and occluded objects, besides multiple instances of the same objects.

Smooth Grad-CAM++ [101] is amid to enhance visual sharpness and object localisation of heatmaps by combining SmoothGrad [72] and Grad-CAM++ [100] approaches. The first three order partial derivatives of the score w.r.t the feature map are averaged to compute α_{ij}^{kc}. The coefficient term α_{ij}^{kc} reflects the importance of a specific pixel in the feature map. Thus, it is used to compute neuron importance weights w_k^c (eq. (9)). Lastly, Smooth Grad-CAM++ [101] can be obtained using eq. (10).

2.3.10. *Other visualisation approaches*

Randomised Input Sampling for Explanation (RISE) [102] can be considered a general approach that does not require any prior knowledge of the network's weights or any network adaptation. RISE can be considered as a genuine black box explanation approach. It only requires access to the input and output of the base model. The process to generate heatmaps using RISE is as follows: the input image is element-wise multiplied by randomly generated masks. Masked inputs are then fed to the black-box model to calculate the confidence scores. Finally, the weighted sum of the random masks, using each mask's output score as its weight, is used to compute the final heatmap that explains the network decision (the contribution of each pixel to the network prediction). Small binary masks are sampled and then up-sampled to a larger resolution using bilinear interpolation to avoid adversarial effects while generating the input masks. This results in masks with values in the range of zero to one. In addition, it produces smooth heatmaps. RISE is a heavy computational approach that requires several forward propagations through the base model (it uses 4000 masks and 8000 masks for VGG-16 and ResNet50, respectively). On the other hand, it can detect many objects of the same class in an image.

Score-CAM is introduced to tackle gradient issues such as gradient saturation and false confidence [103]. Unlike Grad-CAM [87] and Grad-CAM++ [100], Score CAM [103] is independent of gradients during the calculation of the channel's importance. However, the increase in score confidence is used by Score-CAM to quantify the channel's importance. Also, unlike RISE [102], which generates random masks to be multiplied by the input image, Score-CAM uses the extracted activation maps from the last convolutional layer (theoretically, any convolutional layer can be used) as masks on the input image. The heatmap generation for Score-CAM can be summarized as follows [103]: first, activation maps are extracted. Then, the target class's score is obtained by element-wise multiplication of the input image with the extracted feature map (feature maps act as a mask on the original image), then forward passing the product through the CNN. The process is repeated for all of the extracted activation maps. Lastly, score-based weights and activation maps are combined linearly to generate the heatmap. It is found that the concept of increased confidence scores is a better way to quantify activation map importance as it avoids gradients drawbacks for weights' importance calculations.

Saliency can be defined as the smallest region of the image that alone produces a confident score or the so-called Smallest Sufficient Region (SSR) [104]. Another definition is the smallest region of the image that degrades the confident score when removed or the so-called Smallest Destroying Region (SDR) [104]. Dabkowski et al. [104] and Fong et al. [105] proposed mask-based model techniques to achieve both SSR and SDR. A resemble technique for semantic segmentation approaches (the model of [104] adapts the U-Net architecture [106], which is mainly used for semantic segmentation tasks) is used. This may explain the reason for producing better localisation (Table 1) compared to other saliency methods that attempt to detect only parts of interest to the model in an object (not the whole object) that is responsible for a specific prediction or so-called relevance heatmaps [59].

Adversarial evidence can negatively impact optimised-based visual explanation methods because the computations involved in both adversarial and optimised-based visual explanation methods are similar. Consequently, regularisation and constraint techniques are needed to counteract the faulty evidence in explanation methods [104,105]. Such techniques result in smooth and low-resolution heatmaps for which the fine-grained details are lost [107]. Moreover, they introduce hyperparameters that need to be tuned. Fine-Grained Visual explanation (FGVis) [107] method extended the mask model technique [105] and proposed a defence approach that does not introduce hyperparameters. Additionally, neither smoothing nor regularisation is needed. The proposed technique filters gradients that may introduce adversarial evidence due to the adversarial noise during the optimisation process. The main concept is to allow the activation of only the CNN neurons (feature indicators) that are triggered by both the explanation and the original image. This enforces the explanation to contain subset features of the original image features (prevent the generation of new unwanted evidence) and exist at the same location as the original image [107]. As pixels are optimised individually, high-resolution explanations that preserve image characteristics can be obtained [107].

Feedback CNN [108] is a unified system that can classify and localise objects. The proposed network uses both forward and backward paths to visualise neuron activation. It introduces feedback layers that are stacked on top of the ReLU layers and only activate the gates responsible for target neurons depending on the sign of each neuron's gradient. Feedback CNN achieved competitive performance for object localisation using weakly-supervised information compared to fully-supervised state-of-the-art systems.

Table 1
Localisation error for different saliency methods on the ImageNet validation dataset.

Approach	Localisation error (%)
Gradients [31]	41.7
Guided Backpropagation [32]	42.0
LRP [33]	57.8
CAM [34]	48.1
Grad-CAM [87]	47.5
Feedback [108]	38.8
Mask [105]	43.1
Mask [104]	36.9
Excitation backprop [122]	39.0
Occlusion [29]	48.6

Local Interpretable Model-agnostic Explanations (LIME) [109] is a non-backpropagation based approach for CNNs interpretations. To explain an input, it uses a local linear model around that input to approximate the CNN's behaviour. The process can be summarised as follows: images are segmented into features. These features are used to generate synthetic data. The CNN is then used to classify the generated synthetic data. For each synthetic image, a regression model is fit to indicate the presence or absence of such features. This means the new simpler model approximates the behaviour of the complex CNN in the region of observation. Finally, the importance of each input feature can be estimated from the coefficients of the linear model. The important features can be visualised as a map to indicate regions of the image that directly influence the model prediction. As LIME method requires many passes through the CNN, it is considered a computationally expensive approach compared to other explanation methods. Another disadvantage of LIME method lies in the approximation technique used, as it is challenging to approximate complex non-linear models such as DCNN. An example of the main contributing features generated by LIME technique is shown in Fig. 9.

Inspired by ensemble models, Rieger et al. [110] applied the same idea to the explanation methods to reduce variance and bias in machine learning tasks. Aggregating the explanation methods approach has proven its efficiency compared to single explanation methods to identify important features more accurately, reduce variance and bias, and resist adversarial attacks. Two aggregate explanation methods are introduced: AGG-Mean and AGG-Var. AGG-Mean can be calculated by taking the average over all the explanation methods. This will result in fewer errors than the typical error of an individual explanation method, consequently, low variance. Error in this context refers to the mean square error (MSE) between the aggregated explanation and the hypothesised true explanation. For AGG-Var, the AGG-Mean is divided by its standard deviation. This will include the difference between methods uncertainty in the calculation, which can be utilised to assign less relevance to regions with a high conflict between methods. Both AGG-Mean and AGG-Var have shown significant improvement quantitively and qualitatively over individual methods.

Moreover, using various explanation methods for aggregating explanation has a smoothing effect like Smooth-Grad [111] but with fewer computations, explaining the resilience of aggregating methods to adversarial attacks. Replacing ReLU units with Softplus [112] can produce more robust explanations. However, architecture modification is an undesirable approach to visualise network predictions [111].

It is clearly shown that none of the introduced approaches is perfect. Some need special design adaptions; others need a combination of two or more techniques to achieve significant visualisation results. Fig. 10 attempts to gather the post-hoc visualisation techniques, spinoffs, and enhancements in one figure. In the end, the best approach is limited by the application and network architecture.

3. Applications and use-cases of explanation methods

Visual inspection using explanation techniques can add a further dimension to the evaluation process of the robustness of neural network models. It can be used to debug models and identify biases. Standard evaluation procedures of CNNs by testing the system performance on the validation part of a dataset can be less informative because the validation dataset can be limited or biased. Consequently, applying the explanation techniques to ensure the reliability and trustworthiness of systems is vital and beneficial in many applications, particularly critical applications that cannot tolerate errors. For instance, Zhang et al. [113]

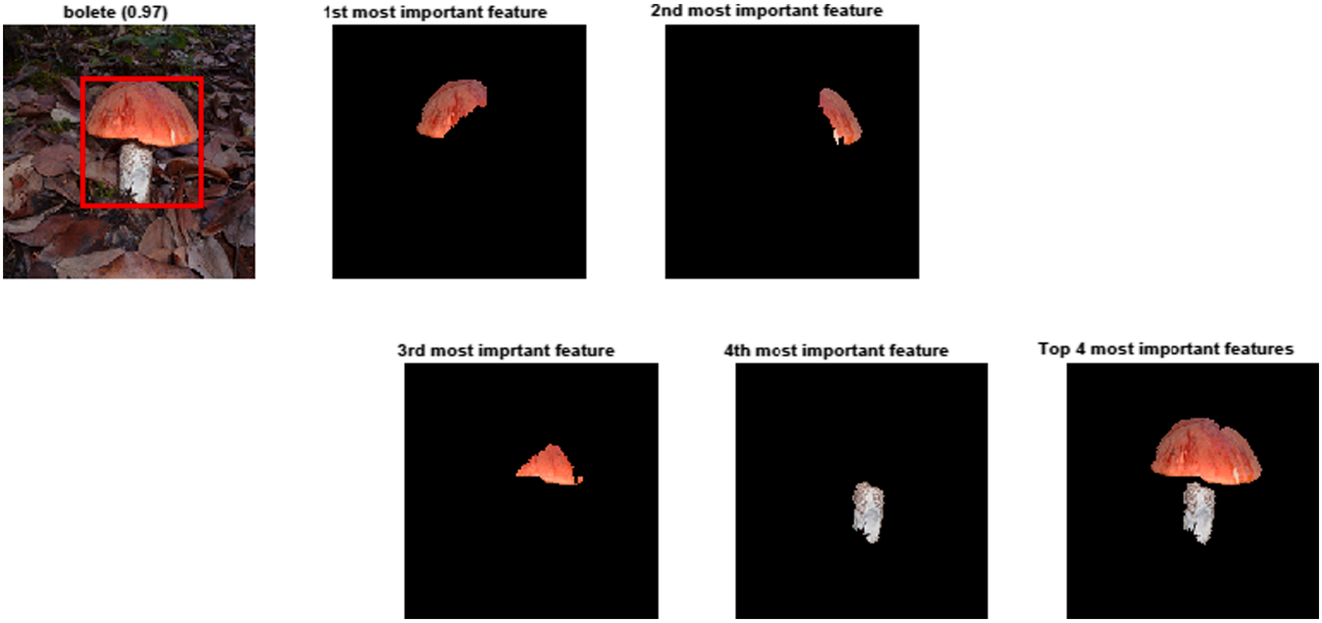

bolete (0.97)
1st most important feature
2nd most important feature
3rd most imprtant feature
4th most important feature
Top 4 most important features

Fig. 9. Examples of LIME maps.

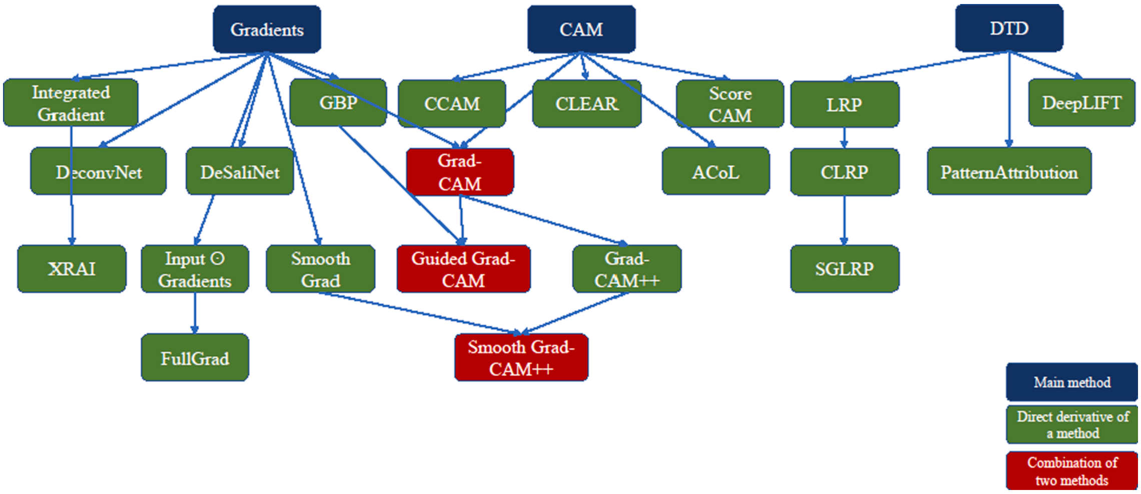


Fig. 10. A chart of post-hoc visualisation techniques.

presented the application of XAI in diagnosis and surgery, a promising research area for medical applications. Ahmed et al. [114] explored XAI usage in the fourth industrial revolution (Industry 4.0). Stakeholders in XAI are discussed from an engineering perspective in [115] with a case study on autonomous cars.

One of the most important uses of visualisation techniques is the selection and modification of CNN architectures. Visualising the learned features by the first and second layers of the AlexNet architecture reveals that the first layer filters retain a mix of high and low-frequency information. On the other hand, the second layer filters retain aliasing artefacts caused by the large stride of the first layer. Zeiler et al. [29] proposed a new architecture that can enhance the model performance by reducing the stride of the first layer. Besides, the filter size of the first convolution layer is reduced to 7 × 7 instead of 11 × 11. The modified version of AlexNet achieved high accuracy. Most importantly, the first and second layers can now preserve better representations.

Zhu et al. [116] visualise the first four convolutional layers trained to classify galaxy morphology. Filter visualisation can give insights into what the model has learned during the training process. The filters of the first convolutional layer can detect galaxy edges and corners, which are used by the second layer filters to detect simple shapes. The filters learn to detect more complex shapes and patterns as the CNN goes deep. Visualising CNN filters helps to debug the model and enhance the architecture by adding or removing layers, controlling filters sizes, and modifying the filter's stride.

Score-CAM [103] is used to debug different systems, identify dataset bias, and explain wrong predications. Even with poor classification confidence, Score-CAM can achieve adequate localisation. Nevertheless, the quality of saliency maps increases as the model performance increases. Thus, better quality maps indicate model convergence.

In the document classification domain [117], LRP [33] is used to interpret the predictions of two different models. Although the two models have achieved the same test accuracy, their approaches to classifying the documents differ. LRP explanation techniques show that the Support Vector Machine (SVM) model based its decision on the word count. In comparison, the CNN model assigns more relevance to the keywords. The case study shows that explanation techniques can be used to understand models' behaviours.

In the image classification domain [118], LRP is used to compare the predictions of a Fisher vector classifier [68] trained on the PASCAL dataset and a CNN trained on the ImageNet dataset. Both systems produce different relevance maps, though they have achieved similar classification accuracy on the horse category. The heatmaps of the Fisher vector model assign high relevance to the copyright tags that are usually presented in horse images. However, the relevance maps of the CNN model show that the model bases its decision on the horse features using the edges and contours.

Using visualisation techniques can help to mitigate the system's weaknesses by identifying biases, such as the case of the Fisher vector model in the classification of horse images. Retraining the Fisher vector model on untagged horse images can help to mitigate the bias issue.

In the biometric domain, LRP has been used to identify the pixels responsible for age and gender characteristics in face images [119]. Also, GBP [32] is used to highlight features corresponding to shadow pixels in 2D ultrasound images [120]. This technique can help to generate shadow-focused confidence maps that can be used in biometric measurements. In the medical domain, LRP explanation techniques have been used to visualise EEG heatmaps to understand which part of the brain is responsible for a particular decision [121].

The choice of the appropriate explanation method is mainly dependent on the application. Also, the method's reliability can significantly influence that choice because some methods are merely edge detectors (evaluating explanation methods is presented in section 4). Generally, applications that require high-resolution heatmaps may find gradient-based methods helpful. These methods can be used for masking foreground objects from the background with sharp edges and fine-grained details. For example, it can be used for medical applications to detect cancer cells. However, discriminative region methods can be used for weakly-supervised tasks where the annotated data for object localisation (bounding boxes) are unavailable. These methods are suitable for semi-supervised tasks. In conclusion, there is no perfect method, but researchers have to trade-off between different methods to achieve the required objective.

4. Evaluating different explanation methods

Localisation error is argued to be a descriptive metric for assessing saliency methods. Table 1 shows the performance of different saliency methods on the ImageNet validation dataset for localisation.

Results in Table 1 are reported in [104,105,108,122], following the same evaluation protocol as [108] and using the same CNN (GoogleNet). The evaluation process on the localisation task is as follows: given an image, the class of interest, and the corresponding saliency map, the object segmentation mask is computed by thresholding the foreground area to cover 95% energy out of the produced saliency map. Then, the tightest bounding box containing the whole object in the saliency map is calculated as the localisation bounding box. This localisation box is only considered valid if the Intersection over Union (IoU) with the ground truth bounding box is greater than 0.5. Different thresholds for localisation error may explain the differences in the originally reported results. However, the results reported in Table 1 use the same threshold value for consistency.

It is not only important to understand how different visualisation techniques work but also if they are valid or not. What are the intuitions behind the performance of these techniques? Are these techniques reliable enough to put our trust in their visualisations? How can we evaluate these methods? Are these methods dependable on model parameters, architecture, and training data? Visualisation methods are key tools to get intuitions into models' predictions. Consequently, understanding their failure is necessary for their usage in critical applications like medicine and security, where mistakes may cause tragic consequences.

Some studies tried to answer these questions [73,123]. Besides, an XAI toolkit called Quantus [124] is introduced to quantitatively evaluate the explanations of neural networks comprehensively and speedily. Quantus is built to ensure the transparency and reproducibility of the evaluation process. However, this field is starving for more research and investigation. Table 2 groups different evaluation studies according to the manipulated parameter and the used technique. These techniques are investigated in the following subsections.

4.1. Weights manipulation

The learned weights during the training process should influence the visualisation techniques used to get insights into the model's prediction, whether these predictions are right or wrong. Manipulating models' weights should affect the resulting heatmap, which means the heatmap is dependable on the weights learned by the model [73]. In contrast, a visualisation technique is undependable on the weights if randomising the models' weights does not affect the resulting activation map. Hence the first introduced test is model parameter randomisation [73], at which all the weights of the model are randomised at once, then the resulting heatmap is tested. Another version of the test is to randomise one layer's parameters at a time from top to bottom successively and monitor the influence on the output heatmap. Moreover, randomise a single layer while keeping other learned layers fixed. The resulting heatmaps from the randomly initialised untrained network are compared with the trained model ones (original weights). A visualisation technique that is dependable on the learned models' parameters should produce two different heatmaps for each network. In contrast, insensitive visualisation approaches to the learned models' parameters will have similar maps. Shortly, randomising models' weights should break 'disturb' the output saliency map for a visualisation technique to pass this test. Failing this test means a particular visualisation technique cannot be used to debug a model.

To provide a quantitative comparison besides the qualitative one, similarities between both maps are calculated using several metrics such as Spearman rank correlation [130], Structural Similarity Index (SSIM) [131], and histogram of gradients (HOGs) [132]. A low correlation between the produced saliency maps is observed for Gradients and Grad-CAM methods. In contrast, high correlation maps for Guided Backpropagation and Guided Grad-CAM are obtained (Table 3).

Nie et al. [125] showed through theoretical and practical analysis that visualisation methods such as Guided Backpropagation [32] and DeconvNet [29] are class insensitive. Gradients [31], Guided Backpropagation, and DeconvNet are assessed using a simple three-layer CNN with random Gaussian initialised weights. As these explanation methods should visualise weights, perturbed weights should result in random noise maps. However, it is found that Guided Backpropagation under some conditions, in this case, a sufficiently large number of filters, can be approximated as the input image regardless of the class label. On the other hand, DeconvNet and Gradients, under the same condition, can be approximated as Gaussian random variables. The behaviour of the Gradients method is understandable because it visualises the output class derivatives w.r.t input image. An operation that is mainly weights dependent. However, DeconvNet has conducted a similar attitude to Guided Backpropagation when used in a CNN with max-pooling layers. The Max-pooling layer is believed to be responsible for image-specific information in DeconvNet [59,133].

Table 2
Evaluation methods to assess different explanation techniques.

Evaluation method	Study
Weights manipulation	[73,125,126]
Data randomisation	[73]
Architecture manipulation	[126,88]
Input perturbation	[123,127]
Evaluation metrics	[61,104,128,102]
Behavioural assessment	[129,78]

Table 3
Test results of different saliency methods.

	Test		
Approach	Model Parameter Randomisation Test	Data Randomisation Test	Input invariance
Gradients	Pass	Pass	Pass
Gradients ⊙ Input	Fail	Fail	Fail
Smooth-Grad	Pass	Pass	Saliency method dependant
Integrated Gradients	Fail	Fail	Reference point dependant
Guided Backpropagation	Fail	Fail	Pass
Grad-CAM	Pass	Pass	*
Grad-CAM ++	Pass	Pass	*
Score-CAM	Pass	Pass	*
Guided Grad-CAM	Fail	Fail	*
PatternNet	*	*	Pass
Deep Taylor Decomposition	*	*	Reference point dependant

* Not reported.

Adversarial attacks by manipulating class labels and ReLU states are conducted on state-of-the-art CNNs such as VGG [2] to test the class sensitivity of the visualisation models. Unlike the Gradients approach, it is shown that Guided backpropagation and DeconvNet are input invariant. Their performance is proven to be an analogy for recovering input images, which is asserted by their insensitivity to class labels. Thus, providing high-quality heatmaps is attributed to the usage of the backward ReLU and the local connections in CNNs but is not related to CNN's weights or inputs [125].

Viering et al. [126] considered an adversary technique that manipulates the model's weights and architecture to generate any desired explanation with a minimum impact on the model's accuracy. Four techniques are introduced to manipulate Grad-CAM's explanations: constant flat and constant image explanations manipulate the model's weights to produce a constant explanation regardless of the input. On the other hand, an input pattern triggers Semi-random and malicious explanations to modify the model's architecture to produce random explanations dependent on the input. The first two techniques are easily detected by inspection as they are independent of the input, while the second two techniques are hard to detect as they are randomly dependent on the input. In all cases, the prediction accuracy does not change significantly. The manipulations are produced using almost the same process: an extra filter is added to the last convolutional layer containing the desired target explanation. Besides, the architecture or weights of the fully connected layers are changed. Results show that the Grad-CAM explanation is not robust to the adversary and follows the desired target explanations.

We want to expand on the malicious explanation triggered by the input pattern technique because hackers might exploit it as a backdoor to abuse systems. The idea is to inject some patterns into the input image. A CNN, which is highly activated by these patterns, will force the explanation to refer to the malicious pattarns. If the malicious patterns do not exist, the output of this CNN will be zero. Consequently, the explanation of the original network can be returned. The introduced techniques [126] can be extended to other gradient-based methods. However, architectural changes are necessary, which might not be a real case scenario, because when a model is deployed, its weights and architecture are kept constant. Thus, we can conclude that Grad-CAM is efficient for a normally trained model but vulnerable to adversarial manipulated models. Generally, heatmaps quality depends on the visualisation techniques, which are intuitively dependent on the model and the training data. A poor performance model will not provide high-quality maps.

4.2. Data randomisation

Data randomisation is the second introduced test by Adebayo et al. [73], in which the training labels for a classification task are permuted. The relation between the data examples and their labels is broken to test the sensitivity of different explanation methods. A CNN is then trained to fit the randomised training data with 95% accuracy. State-of-the-art CNN can be taught to memorise random labels by overfitting the model [134]. As the data is inconsistent, the test accuracy is significantly low. Visualisation techniques are used to produce heatmaps for the test set examples. The produced heatmaps for a model trained on consistent data should look different from heatmaps that have been trained on shuffled data, which means the explanation approach is sensitive to data. On the other hand, a visualisation technique that produces the same heatmaps for both networks will fail this test. Consequently, this technique is insensitive to labels randomisation, and it cannot explain the connection between an example and its label.

To provide a quantitative assessment besides the qualitative one, the correlation between the heatmaps of different visualisation techniques trained on both labels (models trained on the true label and random labels) is calculated. A low correlation means no relation, which is considered a reliable technique and vice versa for high correlation [73].

The presented evaluation approaches have shown alarming results regarding some of the widely used explanation methods. Table 3 shows the testing result of different saliency methods. Some of the tested techniques have no relation to the model or the training data. These visualisation methods are merely a simple edge detection algorithm that does not depend on the model or the training data [73]. This claim is investigated using a simple case study of a one-layer CNN model. As edges' regions in an image have different activations from surrounding pixels, they may visually emerge, which is one of the basic functions of CNN filters. These models, which failed the proposed tests, are unsuitable for investigating data or modelling dependable tasks. They cannot be used to find the relation between inputs and outputs, model debugging or data outliers because these kinds of tasks are model- and data-dependent [73].

4.3. Architecture manipulation

The DeconvNet approach is meant to visualise neurons' responses activated by objects or visual patterns. Mahendran et al. [88] argued that the response at different image depths relies on the network architecture, not the learned weights or data [88]. Through investigation of different neurons with different parameters, it is concluded that the visualisation of DeconvNet is mainly dependent on the information gained during a forward pass or so-called bottleneck information (pooling switches and ReLU masks) [88]. Consequently, DeconvNet visualisation is independent of the selected neuron activation, which means it is not neuron discriminative.

Network architecture may have a significant impact on the models' predictions. Many studies illustrate the influence of randomly initialised weights, which greatly impact the network classification capabilities [135,136]. Besides, it makes the network more immune to noise and produces high-resolution output without more training data [137]. These kinds of networks may produce saliency maps that are independable of the model parameters or the input data but rather on the model architecture. In this case, using explanation methods is possible when the network architecture is believed to be sufficient for reasonable predictions.

4.4. Input perturbation

Implementation invariance is another quantitative method to assess different saliency maps [71]. Models with different architectures that produce the same predictions for all inputs should always generate similar heatmaps to satisfy reliability. Kindermans et al. [123] proposed an additional invariance test called input invariance that adds a constant shift to the input to assess the model's sensitivity to input transformations. If an explanation method fulfils input invariance, it can be considered a reliable interpretation method.

Input transformation, which is used in some cases as a pre-processing technique, can be used to manipulate attributions. However, it does not affect models' predictions or weights. Disturbingly, some widely used methods fail this test (Table 3). However, data normalisation techniques may minimise this failure. Two networks are used to test the sensitivity of saliency methods to input transformation. Both have the same weights and produce the same output for all input instances. The only difference is the addition of the mean shift to the bias of the first layer activation, which cancels out the shift transformation. An explanation method to pass the input invariance test should produce identical heatmaps for both networks where network 1 accepts the input and network 2 accepts a shifted version of that input.

Table 3 shows that Gradients, Guided Backpropagation and PatternNet [69] pass the test because both networks have identical weights. Kindermans et al. [123] assumed that these methods would fail if models with different weights/architecture but the same output predictions for inputs were used. Gradient times input [70] fails to pass the input invariance test as the input shift is propagated to the saliency heatmap. Moreover, multiplying by the input limits visual explanations [72].

The sensitivity to the input invariance test of Integrated Gradients [71] and Deep Taylor Decomposition [82] depends on the reference point's choice, which is a hyper-parameter that can be tuned. Using a black image as a reference point for image classification tasks is a normal choice reference point [71]. At the same time, zero vector would be a suitable reference point for text-based networks [71].

SmoothGrad [72] technique uses duplicated versions of the input with added noise to produce heatmaps using any visualisation method. The resulted maps are then averaged to produce the final saliency map. Consequently, SmoothGrad depends on the underlying method used to produce these maps. If the Gradients method is used, then SmoothGrad is input invariant. On the other hand, if GI is used to produce the heatmaps, it will fail the input invariance test.

Ghorbani et al. [127] introduced three input perturbation techniques to manipulate inputs to produce different interpretations without changing the output predictions. The first perturbation is a random sign perturbation at which each pixel value is randomly changed with some constraint norm. The second perturbation technique is iterative attacks against the explanation methods at which three alterations are introduced to maximise the difference between the original and perturbed interpretation. The third perturbation technique is a gradient sign attack against influence functions at which the equation for the influence functions is linearised around the values of the current inputs and parameters.

Three explanation methods are tested: Gradients [31], Integrated Gradients [71] and DeepLIFT [81]. The tested saliency methods give different heatmaps from the original ones when subjected to input perturbations. Although input manipulation does not change the network prediction or significantly change the confidence score, it is imaginable that changing an input can produce different saliency maps as visualisation methods are sensitive to input changes. However, having the same output prediction should at least produce analogous saliency maps. The cause of this fragility is attributed not only to the explanation methods but also to the network itself that is being vulnerable to such perturbations. Ghorbani et al. [127] blamed the high-dimensionality and non-linearity of CNNs for producing fragile explanations vulnerable to adversarial attacks. It is suggested to Constrain the non-linearity of CNNs while training [138] to overcome this weakness. Though, Goodfellow et al. [40] attributed the vulnerability of models to adversarial perturbation to the models' linear nature. One can argue that they used easily optimised CNNs, which are intrinsically flawed, as easily optimised models can be easily perturbed.

4.5. Evaluation metrics

Deletion and Insertion metrics [102] are introduced to measure the changes in classification output score as important pixels are gradually removed or added from/to an image. A good explanation should show a sharp drop in confidence score for the deletion metric as important pixels, determined by saliency methods, are removed from an image, consequently a low Area Under Curve (AUC). In contrast, a high AUC indicates better explanations as important pixels are being added to an image in the case of the insertion metric. Using these metrics, the RISE [102] approach outperforms Grad-CAM [87] and LIME [109].

Ancona et al. [61] noticed that the Occlusion approach [29] could highlight the impact of individual features distinctively. However, Integrated Gradient [71] can better explain jointly features. An attribution metric called 'sensitivity-n' is introduced to understand the impact of each feature compared to the impact of a group or several ones upon deletion. For an explanation method to satisfy the sensitivity-n metric, the sum of attributions of any subset of features of a cardinality 'n' should correlate to the variation of the output caused by removing these features. Pearson Correlation Coefficient (PCC) is used to quantify the correlation between the decrease in output for a subset of the removed features and the sum of their relevance for each n. Authors [61] argued that explanation methods such as Occlusion, Gradients times input, Integrated Gradient, LRP and DeepLIFT can satisfy sensitivity-n metric if they are applied to linear or linearly behaved models.

Based on SSR, Dabkowski et al. [104] proposed a saliency metric to assess different saliency methods. First, it finds the smallest rectangular crop that contains the entire salient region. This rectangular region is then fed to the classifier to verify whether it can predict the correct class. Cropping is used to avoid adversarial artefacts that might be introduced by masking. Manipulating the image by masking SSR or SDR regions using pixels blurring or added noise may introduce adversarial artefacts. Although it is usually tiny changes, it can lead to evidence of wrong classes; that is why it needs to be avoided. A low value of saliency metric, which quantifies the amount of relevant information captured by the rectangular region, means the explanation approach can reduce the rectangular cropped size while maintaining the classification score, which is a good attribution for that explanation method [104]. The proposed masking model [104] achieved a lower saliency metric compared to Gradients [31] and excitation backprop [122].

More recently, Bansal et al. [128] proposed an evaluation metric that evaluates the sensitivity of attribution methods to the change in their hyperparameters. Explanation methods hyperparameters such as random seeds for LIME [109] or sample size and Gaussian standard deviation for Smooth-Grad [72], which are randomly set, can be used to assess the robustness of an explanation method. A robust explanation method should be independent of the arbitrary hyperparameters choices, i.e., it should reproduce the same heatmaps for different hyperparameters. For gradient-based methods, robust classifiers (classifiers trained to limit adversarial pixel noise) produce heatmaps that demonstrate smooth object structure, unlike regular classifiers, which produce uninterpretable maps. This is also valid for hyperparameters changes as gradients-based explanations are insensitive to the added random noise to the input image. For example, Smooth-Grad explanation maps for robust classifiers under different sample sizes produce almost the same results for a different number of samples. In contrast, regular models produce enhanced quality maps as the number of samples increases. On the other hand, non-gradient based methods, such as Occlusion [29], are sensitive to their hyperparameter (the patch size in the case of the Occlusion method), whether a robust or non-robust classifier is used [128].

4.6. Behavioural assessment

Yeh et al. [129] introduced infidelity and sensitivity checks to assess explanation methods quantitatively and qualitatively. Sensitivity measures the impact of insignificant perturbations on an explanation method. In comparison, infidelity measures the difference between the output perturbation and the dot product of input perturbation with the explanation. Infidelity is used to test the relevance of the important features in an explanation to a subset of predefined features. A combined map of Smooth-Grad and Integration Gradients has shown an optimal ability to minimise infidelity. Using Smooth-Grad with base explanation methods has shown a high ability to reduce sensitivity and infidelity. Methods that can optimise fidelity can pass sanity checks [73].

Montavon et al. [78] presented systematic and objective assessments for the quality of explanation methods using a simple task where the same inputs, predictions, and network architectures are used. Appropriate parameters are transferred from a known domain-related task, such as the classification of digits [139], to a target domain, in this case, handwritten characters [140]. Simple tasks do not require an expert to evaluate different explanations compared to field-specific tasks such as tumour identification. Two behavioural properties of explanation methods are investigated: explanation continuity and explanation selectivity. The produced explanation function should be continuous for a continuous output function. This means that the produced explanation for any two input equivalent data points should be equivalent [78] which can be quantified by the strong variation in the activation maps.

Three explanation methods are investigated: simple Taylor decomposition [141], sensitivity analysis (backpropagation method) [31], and LRP [33]. To investigate the explanation continuity property of a method, an MNIST digit [139] is translated from left to right while tracking the output and relevance scores. The relevance method scores the importance of image pixels according to their impact on the output prediction [33]. Only LRP produced a smooth continuous transition. Gradient-based methods such as sensitivity analysis and simple Taylor decomposition tend to be discontinuous because they are more liable to gradient noise [142] and shattered gradients [143,144] that occur due to ReLU units in CNNs.

Explanation selectivity is quantified by measuring the response of the output score's degradation when patches corresponding to important features are removed. A process known as 'region perturbation' [54], a generalisation of 'pixel-flipping' [28] in image domain tasks, illustrates how an explanation method redistributes relevance to pixels that influence the networks' predictions. The same technique is used in the text data domain by setting word embedding to zero for selected words [117]. In explanation selectivity, features are sorted in descending order according to their relevance scores. Sequentially, features are removed (by setting corresponding pixels to black or by randomly sampling values from a uniform distribution). Then, output scores are calculated. Finally, the area under the curve is calculated for the plot of output scores against the number of removed patches [78]. The low AUC represents a low output score, meaning the most influence features are correctly detected. LRP and Guided Backpropagation have achieved high explanation selectivity compared to simple Taylor decomposition, sensitivity analysis, and DeconvNet. LRP outperforms other methods due to its ability to detect negative evidence [59]. In addition, it produces less noisy heatmaps because of the normalisation property [59].

Other experiments are conducted to understand the reason for the failure of explanation methods that visualise the element-wise product of the gradients and inputs [70] or explanation methods that can be approximated to gradients times inputs. The element-wise product of a fixed input with two random vectors is calculated. The two results are then compared. It is observed that the input dominates the product, especially for sparse inputs, even with a significant change in the random vectors [73]. Explanation methods such as Integrated Gradients [71], LRP [33] and DeepLIFT [81] can be reduced to gradients times input under certain conditions [23,61,145,70]. For these methods, the experiments have concluded that as the gradients tend to be visually noisy, they will return mostly the input [73,69].

5. Conclusion

This paper presents an extensive review of different visualisation techniques as explanation methods for the operation of convolutional neural networks. State-of-the-art techniques are discussed and compared. Besides, the merits and drawbacks of the visualisation methods are highlighted.

Although explainable AI techniques are essential in understanding the behaviour of CNNs, they can be easily misinterpreted, especially in the case of visualisation methods. For example, a saliency method can highlight the edges of an object as its activation pixels. Although this may be a simple edge detection in the image processing sense, it could be misinterpreted by unrobust visualisation methods as revealing a relation between the trained model's weights and the output labels. Thus, visual inspection in these cases is insufficient because explanation methods of unrobust systems may suffer from user subjectivity and unreliable explanations.

The presented analysis and discussion offer guidance on how and in which cases these methods can be fruitfully utilised. Understanding the strengths and weaknesses of visualisation approaches is the key to explaining the systems' behaviour. Consequently, explanation methods can facilitate the deployment of deep learning-based systems in real-life applications. Important application domains such as natural language processing or multi-object detection systems for autonomous navigation and driving (e.g., robotic assistants for the elderly or autonomous cars) suffer delays in wider adaption and mass deployment because explaining the systems' decisions is overlooked.

Furthermore, there is a significant debate among researchers to define the qualitative characteristics of a model's transparency and how the properties of related explanation methods can be assessed. One side addresses the issue as an object recognition task in a weakly-supervised manner, where the whole object should be identified by the explanation method in order to be reliable. The other side argues that a good explanation method should highlight only the most discriminative parts of an object which make it belongs to a particular class.

We believe this review is a significant contribution that enables the researchers in the field to better understand the structure and properties of different visualisation techniques and facilitate the choices of the appropriate method for a specific application.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the Assistive Devices for empowering disAbled People through robotic Technologies (ADAPT) project. ADAPT is selected for funding by the INTERREG VA France (Channel) England Programme which is co-financed by the European Regional Development Fund (ERDF). The European Regional Development Fund (ERDF) is one of the main financial instruments of the European Union (EU) cohesion policy.

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.

[2] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *3rd International Conference on Learning Representations, {ICLR} 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[3] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition. in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 779–788, 2015.

[6] W. Liu, et al., SSD: Single shot multibox detector. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[7] E. Shelhamer, J. Long, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 640–651.

[8] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495.

[9] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11211 LNCS, pp. 833–851.

[10] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Muller, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, vol. 11700. 2019.

[11] Z.C. Lipton, The mythos of model interpretability, Commun. ACM 61 (10) (2018) 36–43.

[12] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead", Nature Machine Intelligence. 1 (5) (2019) 206–215.

[13] D. Alvarez-Melis, T.S. Jaakkola, Towards robust interpretability with self-explaining neural networks. in *Advances in Neural Information Processing Systems*, 2018.

[14] C. Seifert *et al.*, "Visualizations of Deep Neural Networks in Computer Vision: A Survey," 2017.

[15] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (5) (2019) 1–42.

[16] Q.-S. Zhang, S.-C. Zhu, "Visual interpretability for deep learning: a survey", Frontiers of Information Technology and Electronic Engineering. 19 (1) (2018) 27–39.

[17] Q. Zhang, R. Cao, F. Shi, Y.N. Wu, S.C. Zhu, Interpreting CNN knowledge via an explanatory graph. in *32nd AAAI Conference on Artificial Intelligence*, 2018, 2018..

[18] Q. Zhang, Y. Yang, H. Ma, Y.N. Wu, Interpreting cnns via decision trees. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019.

[19] A. Adadi, M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)", IEEE Access 6 (2018) 52138–52160.

[20] A. Abdul, J. Vermeulen, D. Wang, B.Y. Lim, M. Kankanhalli, Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. in *Conference on Human Factors in Computing Systems -*, 2018.

[21] E. Tjoa, C. Guan, "A Survey on Explainable Artificial Intelligence (XAI), Towards Medical XAI" 14 (8) (2019) 1–21.

[22] C. Chen, O. Li, C. Tao, A.J. Barnett, J. Su, C. Rudin, This looks like that: Deep learning for interpretable image recognition. in *Advances in Neural Information Processing Systems*, 2019.

[23] S.M. Lundberg, S.I. Lee, A unified approach to interpreting model predictions. in *Advances in Neural Information Processing Systems*, 2017.

[24] M. Sundararajan, A. Najmi, The many shapley values for model explanation. in *37th International Conference on Machine Learning*, 2020, 2020..

[25] M.T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations. in *32nd AAAI Conference on Artificial Intelligence*, 2018, 2018..

[26] C. Molnar, Interpretable Machine Learning. A Guide for Making Black Box Models Explainable, Book, 2019.

[27] W. Samek, G. Montavon, S. Lapuschkin, C.J. Anders, K.-R. Muller, Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications, Proc. IEEE 109 (3) (2021) 247–278.

[28] F. Grün, C. Rupprecht, N. Navab, and F. Tombari, "A Taxonomy and Library for Visualizing Learned Features in Convolutional Neural Networks," vol. 48, 2016.

[29] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks arXiv:1311.2901v3 [cs.CV] 28 Nov 2013," *Comput. Vision–ECCV 2014*, 2014.

[30] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Object detectors emerge in deep scene CNNs. in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[31] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*, 2014.

[32] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*, 2015.

[33] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, O. D. Suarez, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS ONE 10 (7) (2015).

[34] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning Deep Features for Discriminative Localization. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[35] J. Long, N. Zhang, T. Darrell, Do convnets learn correspondence?. in *Advances in Neural Information Processing Systems*, 2014.

[36] A. Mahendran, A. Vedaldi, Understanding deep image representations by inverting them. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[37] D. Erhan, Y. Bengio, A. Courville, P. Vincent, Visualizing higher-layer features of a deep network, Bernoulli (2009).

[38] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[39] C. Szegedy, et al., Intriguing properties of neural networks. in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

[40] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples. in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[41] A. Mahendran, A. Vedaldi, Visualizing Deep Convolutional Neural Networks Using Natural Pre-images, Int. J. Comput. Vis. 120 (3) (2016) 233–255.

[42] A. Dosovitskiy, T. Brox, Inverting visual representations with convolutional networks. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[43] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. (2002).

[44] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding Neural Networks Through Deep Visualization," 2015.

[45] D. Wei, B. Zhou, A. Torrabla, W. Freeman, "Understanding Intra-Class Knowledge Inside, CNN" 6 (2) (2015) 6–12.

[46] A. Nguyen, J. Yosinski, and J. Clune, "Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks," 2016.

[47] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, J. Clune, Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. in *Advances in Neural Information Processing Systems*, 2016.

[48] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L.i. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.

[49] Q. Zhang, Y.N. Wu, S.C. Zhu, Interpretable Convolutional Neural Networks. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[50] B. Zhou, D. Bau, A. Oliva, A. Torralba, Interpreting Deep Visual Representations via Network Dissection, IEEE Trans. Pattern Anal. Mach. Intell. 41 (9) (2019) 2131–2145.

[51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. (2014).

[52] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift. in *32nd International Conference on Machine Learning*, 2015, 2015..

[53] L.M. Zintgraf, T.S. Cohen, T. Adel, M. Welling, Visualizing deep neural network decisions: Prediction difference analysis. in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2019.

[54] W. Yu, K. Yang, Y. Bai, H. Yao, Y. Rui, "Visualizing and Comparing Convolutional Neural Networks" (2014).

[55] L. Van Der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. (2008).

[56] R. Girshick, J. Donahue, T. Darrell, U.C. Berkeley, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (2014) 580–587.

[57] W. Yu, K. Yang, Y. Bai, H. Yao, Y. Rui, DNN Flow: DNN feature pyramid based image matching. in *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, 2014.

[58] M.D. Zeiler, G.W. Taylor, R. Fergus, Adaptive deconvolutional networks for mid and high level feature learning. in *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[59] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, K.-R. Muller, Evaluating the visualization of what a deep neural network has learned, IEEE Trans. Neural Networks Learn. Syst. 28 (11) (2017) 2660–2673.

[60] M. Robnik-Sikonja, I. Kononenko, Explaining classifications for individual instances, IEEE Trans. Knowl. Data Eng. 20 (5) (2008) 589–600.

[61] M. Ancona, E. Ceolini, C. Öztireli, M. Gross, Towards better understanding of gradient-based attribution methods for deep neural networks. in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[62] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533–536.

[63] K. Simonyan, A. Vedaldi, A. Zisserman, "Deep Fisher Networks and Class Saliency Maps for Object Classification and Localisation", ILSVRC Work. (2014).

[64] Y.Y. Boykov, M.P. Jolly, Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. in *Proceedings of the IEEE International Conference on Computer Vision*, 2001.

[65] D. Reynolds, "Gaussian Mixture Models", in *Encyclopedia of*, in: S.Z. Li, A. Jain (Eds.), Encyclopedia of Biometrics, Springer US, Boston, MA, 2009, pp. 659–663.

[66] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, K.R. Müller, How to explain individual classification decisions, J. Mach. Learn. Res. (2010).

[67] P. Felzenszwalb, D. McAllester, D. Ramanan, A discriminatively trained, multiscale, deformable part model. in *26th IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[68] F. Perronnin, J. Sánchez, T. Mensink, Improving the Fisher kernel for large-scale image classification. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010.

[69] P.J. Kindermans, et al., Learning how to explain neural networks: Patternnet and Patternattribution. in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[70] A. Shrikumar, P. Greenside, A.Y. Shcherbina, A. Kundaje, Not Just a Black Box : Learning Important Features Through Propagating Activation Differences. in *Proceedings of the 33rd International Conference on MachineLearning*, 2016.

[71] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks. in *34th International Conference on Machine Learning*, 2017, 2017..

[72] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "SmoothGrad: removing noise by adding noise," 2017.

[73] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, B. Kim, Sanity checks for saliency maps. in *Advances in Neural Information Processing Systems*, 2018.

[74] S. Srinivas, F. Fleuret, Full-gradient representation for neural network visualization. in *Advances in Neural Information Processing Systems*, 2019.

[75] M.A.A.K. Jalwana, N. Akhtar, M. Bennamoun, A. Mian, CAMERAS: Enhanced Resolution And Sanity preserving Class Activation Mapping for image saliency, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. (2021) 16322–16331.

[76] A. Kapishnikov, T. Bolukbasi, F. Viegas, M. Terry, XRAI: Better attributions through regions. in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[77] K. Bykov, A. Hedström, S. Nakajima, and M. M.-C. Höhne, "NoiseGrad: enhancing explanations by introducing stochasticity to model weights," 2021.

[78] G. Montavon, W. Samek, K.-R. Müller, Methods for interpreting and understanding deep neural networks, Digital Signal Processing: A Review Journal. 73 (2018) 1–15.

[79] J. Gu, Y. Yang, V. Tresp, Understanding individual decisions of CNNs via contrastive backpropagation, arXiv. (2018).

[80] B. K. Iwana, R. Kuroki, and S. Uchida, "Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation," in *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, 2019.

[81] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," *34th Int. Conf. Mach. Learn. ICML 2017*, vol. 7, pp. 4844–4866, 2017.

[82] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, K.-R. Müller, Explaining nonlinear classification decisions with deep Taylor decomposition, Pattern Recognit. 65 (2017) 211–222.

[83] M. Lin, Q. Chen, S. Yan, Network in network. in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

[84] A. Krizhevsky, V. Nair, and G. Hinton, "CIFAR-10 and CIFAR-100 datasets," *https://www.cs.toronto.edu/~kriz/cifar.html*, 2009.

[85] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Sci. Dep. Univ. Toronto, Tech. (2009).

[86] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," pp. 1–18, 2012.

[87] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization, Int. J. Comput. Vis. 128 (2) (2020) 336–359.

[88] A. Mahendran, A. Vedaldi, Salient deconvolutional networks. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[89] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Is object localization for free? - Weakly-supervised learning with convolutional neural networks. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[90] C. Szegedy, et al., Going deeper with convolutions. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[91] F. N. Iandola, S. Han, and W. J. Dally, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and textless1MB model size," no. April 2019, 2016.

[92] H. A. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," in *Computer Vision and Pattern Recognition*, 2009.

[93] X. Zhang, Y. Wei, J. Feng, Y. Yang, T. Huang, Adversarial Complementary Learning for Weakly Supervised Object Localization. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[94] Y. Wei, J. Feng, X. Liang, M. M. Cheng, Y. Zhao, and S. Yan, "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[95] X. Zhang, Y. Wei, G. Kang, Y. Yang, T. Huang, Self-produced guidance for weakly-supervised object localization. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.

[96] W. Li, H. Jafari, and C. Rother, "Localizing Common Objects Using Common Component Activation Map," pp. 28–31.

[97] D. Kumar, A. Wong, G.W. Taylor, Explaining the Unexplained: A CLass-Enhanced Attentive Response (CLEAR) Approach to Understanding Deep Neural Networks. in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

[98] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.

[99] K. Li, Z. Wu, K.C. Peng, J. Ernst, Y. Fu, Tell Me Where to Look: Guided Attention Inference Network. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[100] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, 2018.

[101] D. Omeiza, S. Speakman, C. Cintas, and K. Weldemariam, "Smooth Grad-CAM++: An enhanced inference level visualization technique for deep convolutional neural network models," *arXiv*. 2019.

[102] V. Petsiuk, A. Das, and K. Saenko, "RisE: Randomized input sampling for explanation of black-box models," in *British Machine Vision Conference 2018, BMVC 2018*, 2019.

[103] H. Wang, et al., Score-CAM: Score-weighted visual explanations for convolutional neural networks. in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

[104] P. Dabkowski, Y. Gal, Real time image saliency for black box classifiers. in *Advances in Neural Information Processing Systems*, 2017.

[105] R.C. Fong, A. Vedaldi, Interpretable Explanations of Black Boxes by Meaningful Perturbation. in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[106] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015.

[107] J. Wagner, J.M. Kohler, T. Gindele, L. Hetzel, J.T. Wiedemer, S. Behnke, Interpretable and fine-grained visual explanations for convolutional neural networks. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019.

[108] C. Cao, et al., Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[109] M.T. Ribeiro, S. Singh, C. Guestrin, 'Why should i trust you?' Explaining the predictions of any classifier. in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[110] L. Rieger and L. K. Hansen, "Aggregating explanation methods for stable and robust explainability," no. 2014, 2019.

[111] A.-K. Dombrowski, M. Alber, C. J. Anders, M. Ackermann, K.-R. Müller, and P. Kessel, "Explanations can be manipulated and geometry is to blame," pp. 1–34, 2019.

[112] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, R. Garcia, Incorporating second-order functional knowledge for better option pricing. in *Advances in Neural Information Processing Systems*, 2001.

[113] Y. Zhang, Y. Weng, and J. Lund, "Applications of Explainable Artificial Intelligence in Diagnosis and Surgery," *Diagnostics*, vol. 12, no. 2. 2022.

[114] I. Ahmed, G. Jeon, and F. Piccialli, "From Artificial Intelligence to eXplainable Artificial Intelligence in Industry 4.0: A survey on What, How, and Where," *IEEE Trans. Ind. Informatics*, 2022.

[115] F. Hussain, R. Hussain, and E. Hossain, "Explainable Artificial Intelligence (XAI): An Engineering Perspective," Jan. 2021.

[116] X.P. Zhu, J.M. Dai, C.J. Bian, Y. Chen, S. Chen, C. Hu, Galaxy morphology classification with deep convolutional neural networks, Astrophys. Space Sci. (2019).

[117] L. Arras, F. Horn, G. Montavon, K.R. Müller, W. Samek, 'What is relevant in a text document?': An interpretable machine learning approach, PLoS ONE (2017).

[118] S. Lapuschkin, A. Binder, G. Montavon, K.R. Muller, W. Samek, Analyzing Classifiers: Fisher Vectors and Deep Neural Networks. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[119] F. Arbabzadah, G. Montavon, K. R. Müller, and W. Samek, "Identifying individual facial expressions by deconstructing a neural network," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9796 LNCS, no. Gcpr, pp. 344–354, 2016.

[120] Q. Meng, et al., Automatic shadow detection in 2D ultrasound images. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.

[121] I. Sturm, S. Lapuschkin, W. Samek, K.R. Müller, Interpretable deep neural networks for single-trial EEG classification, J. Neurosci. Methods 274 (2016) 141–145.

[122] J. Zhang, S.A. Bargal, Z. Lin, J. Brandt, X. Shen, S. Sclaroff, Top-Down Neural Attention by Excitation Backprop, Int. J. Comput. Vis. (2018).

[123] P.J. Kindermans, et al., The (Un)reliability of Saliency Methods. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019.

[124] A. Hedström *et al.*, "Quantus: An Explainable AI Toolkit for Responsible Evaluation of Neural Network Explanations," Feb. 2022.

[125] W. Nie, Y. Zhang, A.B. Patel, A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. in *35th International Conference on Machine Learning*, 2018, 2018..

[126] T. Viering, Z. Wang, M. Loog, E. Eisemann, "How to Manipulate CNNs to Make Them Lie, the GradCAM Case" 1 (2019) 1–13.

[127] A. Ghorbani, A. Abid, J. Zou, Interpretation of Neural Networks Is Fragile, Proc. AAAI Conf. Artif. Intell. (2019).

[128] N. Bansal, C. Agarwal, and A. Nguyen, "SAM: The Sensitivity of Attribution Methods to Hyperparameters," 2020.

[129] C.-K. Yeh, C.-Y. Hsieh, A. S. Suggala, D. I. Inouye, and P. Ravikumar, “On the (In) fidelity and Sensitivity for Explanations,” no. NeurIPS, 2019.

[130] J.H. Zar, “Spearman Rank Correlation”, in Encyclopedia of Biostatistics (2005).

[131] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: From error visibility to structural similarity, IEEE Trans. Image Process. (2004).

[132] T. Surasak, I. Takahiro, C. H. Cheng, C. E. Wang, and P. Y. Sheng, “Histogram of oriented gradients for human detection in video,” in *Proceedings of 2018 5th International Conference on Business and Industrial Research: Smart Technology for Next Generation of Information, Engineering, Business and Social Science, ICBIR 2018*, 2018.

[133] A. Odena, V. Dumoulin, C. Olah, Deconvolution and Checkerboard Artifacts, Distill (2017).

[134] C. Zhang, B. Recht, S. Bengio, M. Hardt, O. Vinyals, Understanding deep learning requires rethinking generalization. in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2019.

[135] A. M. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng, “On random weights and unsupervised feature learning,” in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 2011.

[136] G. Alain and Y. Bengio, “Understanding intermediate layers using linear classifier probes,” 2016.

[137] V. Lempitsky, A. Vedaldi, D. Ulyanov, Deep Image Prior. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[138] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, “Parseval networks: Improving robustness to adversarial examples,” in *34th International Conference on Machine Learning, ICML 2017*, 2017.

[139] Y. LeCun, C. Cortes, and C. J. C. Burges, “MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges,” 2011. [Online]. Available: http://yann.lecun.com/exdb/mnist/. [Accessed: 15-Aug-2018].

[140] L. van der Maaten, A New Benchmark Dataset for Handwritten Character Recognition, Tech. Report. Tilbg. Univ. Tilburg, Netherlands, 2009, pp. 1–9.

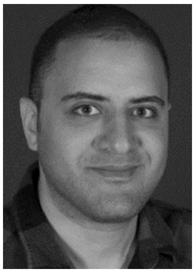
[141] S. Bazen, X. Joutard, The Taylor Decomposition: A Unified Generalization of the Oaxaca Method to Nonlinear Models, AMSE Work. Pap. (2013).

[142] J.C. Snyder, M. Rupp, K. Hansen, K.R. Müller, K. Burke, Finding density functionals with machine learning, Phys. Rev. Lett. (2012).

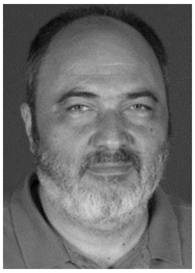
[143] D. Balduzzi, M. Frean, L. Leary, J. P. Lewis, K. W. D. Ma, and B. McWilliams, “The shattered gradients problem: If resnets are the answer, then what is the question?,” in *34th International Conference on Machine Learning, ICML 2017*, 2017.

[144] G. Montúfar, R. Pascanu, K. Cho, Y. Bengio, On the number of linear regions of deep neural networks. in *Advances in Neural Information Processing Systems*, 2014.

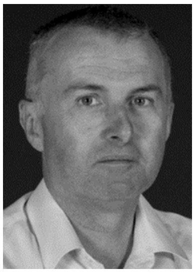
[145] P.-J. Kindermans, K. Schütt, K.-R. Müller, and S. Dähne, “Investigating the influence of noise and distractors on the interpretation of neural networks,” no. Nips, 2016.



ELHASSAN MOHAMED is a Ph.D. candidate with the School of Engineering, University of Kent, Canterbury, UK. He received his M.Sc. degree with distinction in Embedded Systems and Instrumentations form the same university in 2016. He received his B.Sc. degree from Mansoura University, Mansoura, Egypt in Electronics and Communications in 2011. Currently, he is a part of the ADAPT team that is working on developing smart assistive devices for disabled people. His research interests focus on Computer Vision, Embedded Systems, Artificial Intelligence, and Robotics.



KONSTANTINOS SIRLANTZIS is Associate Professor of Intelligent Systems, Head of the Intelligent Interaction Research Group and Academic Lead of the Kent Assistive Robotics Laboratory (KAROL) at the School of Engineering, University of Kent. His main research interests focus on Pattern Recognition, Artificial Intelligence, Robotics, Computer Vision, and their application to Assistive Technology (AT) systems and their security. He successfully gained over £3M in research awards from public and private funders in the UK and Internationally. He has published more than 120 peer-reviewed papers and organized International Conferences (EST 2019) and Thematic Sessions (AAATE 2019) on topics of robotic assistive systems.



GARETH HOWELLS is currently a Professor of Secure Electronic Systems at the University of Kent in the UK and Founder, Director and Chief Technology Officer of Metrarc Ltd, a university spin-out company. He has been involved in research relating to pattern recognition and image processing for over 30 years and has published over 200 papers in the technical literature, co-editing two books and contributing to several other edited publications. His core research interests are in applying soft computing and pattern recognition techniques to the domains of device authentication, biometrics, secure communications, and identity management.