UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Multilevel Explainable Artificial Intelligence: Methods and Applications

by

Halil Ibrahim Aysel

MSc

ORCiD: 0000-0002-4981-0827

A thesis for the degree of
Doctor of Philosophy

July 2025

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Doctor of Philosophy

Multilevel Explainable Artificial Intelligence: Methods and Applications

by Halil Ibrahim Aysel

Thanks to their astonishing prediction ability, deep neural networks (DNNs) have been deployed in various disciplines, from computer vision to natural language processing. However, their opaque decision-making mechanism makes it challenging to employ them in sensitive areas such as healthcare, legal settings, and autonomous driving. Many works have been proposed in the explainable artificial intelligence (XAI) field to overcome this issue and make DNNs more transparent, trustworthy, and deployable. However, most of these methodologies suffer from several drawbacks.

This thesis explores the current landscape of XAI and identifies critical shortcomings in the field that require urgent attention. Through a thorough examination of these limitations, we reveal key gaps that motivate our contributions. To address these challenges, we propose a novel methodology, *multilevel XAI*, which generates human-like explanations in the form of linguistic and visual concepts for machine learning and computer vision tasks. Our approach demonstrates that producing multilevel concept-based explanations can be both cost-effective and achieved without significantly compromising model performance.

Building on this, we introduce a novel weakly supervised semantic segmentation framework, *semantic proportions-based semantic segmentation (SPSS)*. This approach facilitates effective semantic segmentation without the need for costly and impractical pixel-wise ground-truth segmentation maps, which are often challenging to obtain in real-world scenarios. By leveraging class proportions as the sole supervision during training, *SPSS* enables an intuitive and efficient generation of segmentation maps. Furthermore, this framework opens opportunities to integrate the explainability components of *multilevel XAI*, paving the way for future research to achieve semantic segmentation with significantly reduced annotation costs.

We further identified that one of the most significant gaps in the concept-based XAI field—on which this thesis also specifically focuses—is the absence of standardised measures and benchmarks for fair evaluation and selection of the most effective methodologies. To address this, we propose three novel measures and benchmarks to advance

the field. We encourage the research community to employ these measures and benchmarks for a fair comparison among concept-based XAI techniques.

Finally, we discuss the limitations of our work and possible future directions that, once realised, could significantly impact the XAI, machine learning, and computer vision communities.

Contents

List of Figures

List of Tables

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:
    - **Halil Ibrahim Aysel**, Xiaohao Cai, and Adam Prugel-Bennett. "Multilevel Explainable Artificial Intelligence: Visual and Linguistic Bonded Explanations." *IEEE Transactions on Artificial Intelligence,* 5(5), pp. 2055–2066, https://doi.org/10.1109/TAI.2023.3308555, 2023.
    - **Halil Ibrahim Aysel**, Xiaohao Cai, and Adam Prugel-Bennett. "Explainable Artificial Intelligence: Advancements and Limitations." *Applied Sciences,* 15, 7261, https://doi.org/10.3390/app15137261, 2025.

Signed:..... Date:.....

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Dr Xiaohao Cai and Prof. Adam Prügel-Bennett, for their invaluable guidance, support, and encouragement throughout the course of this research. Their insightful advice and constructive feedback have been crucial in shaping this thesis and my academic growth. I am deeply thankful to my friends and colleagues in the Vision, Learning and Control (VLC) group, for fostering an environment of collaboration and support. The collective knowledge and discussions within the group have been invaluable throughout this journey and have significantly contributed to the successful completion of this research. I extend my heartfelt thanks to Republic of Türkiye Ministry of National Education for their generous sponsorship and support. Their contribution has been vital in providing me with the resources and environment necessary to pursue this research. Finally, I would like to express my profound appreciation to my family. Their unconditional love, patience, and belief in me have been my anchor throughout this journey. Their unwavering support has given me the strength to persevere through challenges and accomplish my goals.

To everyone who has contributed to my academic and personal growth, directly or indirectly, I am sincerely thankful.

Definitions and Abbreviations

AI	Artificial Intelligence
CAM	Class Activation Mapping
CAVs	Concept Activation Vectors
CBMs	Concept Bottleneck Models
CEM	Concept Existence Measure
CGIM	Concept Global Importance Measure
CLM	Concept Location Measure
CNNs	Convolutional Neural Networks
CoAM	Concept Activation Mapping
GAP	Global Average Pooling
DNNs	Deep Neural Networks
GDPR	General Data Protection Regulation
GPUs	Graphics Processing Units
LIME	Local Interpretable Model-Agnostic Explanations
LRP	Layer-wise Relevance Propagation
ML	Machine Learning
MLP	Multilayer Perceptron
SHAP	Shapley Additive Explanations
SP	Semantic (class) Proportions
SPSS	SP-based Semantic Segmentation
SVMs	Support Vector Machines
WSSS	Weakly Supervised Semantic Segmentation
XAI	Explainable Artificial Intelligence

Chapter 1

Introduction

Deep neural networks (DNNs) have demonstrated impressive predictive performance across various domains, including medicine [1, 2], robotics [3, 4] and economics [5]. They have been successfully applied to a wide range of problems, such as object detection [6, 7], stock prediction [8], image generation [9, 10] and machine translation [11, 12], among many others. This success can be attributed to advancements in deep learning [13], the availability of massive datasets [14] and the increasing computational power provided by graphics processing units (GPUs) [15].

Despite the significant performance improvements in DNNs over recent years, gaining trust in their predictions remains a challenge due to the complex and opaque nature of their decision-making processes [16–21]. To address this, model interpretability has become essential in uncovering the "black box" of deep networks. Interpretability, often defined as *the ability to provide explanations in terms understandable to humans* [22], plays a crucial role in bridging the gap between DNNs' performance and user trust[1].

The techniques developed to interpret and explain machine learning (ML) models are broadly categorised as explainable artificial intelligence (XAI) methodologies [26, 27]. Consequently, this thesis adopts XAI as its core term. In recent years, the field of XAI has experienced exponential growth, with numerous approaches proposed to address the transparency challenge of black-box models. These approaches aim to make such models safer for deployment, especially in sensitive domains, including healthcare [28, 29] and law [30]. By improving transparency, XAI seeks to achieve several objectives, such as ensuring fairness by detecting and mitigating discrimination or unexpected behaviours in ML systems. XAI techniques also assist practitioners in debugging their systems by identifying biases in data or in the models themselves [20, 31]. Furthermore, explainability is not merely a desirable feature; since 2018, the General Data Protection

[1] Ongoing discussions have explored the distinctions between terms like interpretability, explainability, trustworthiness and transparency, debating their appropriate use in the field [18, 23–25]. In this thesis, we focus on the shared goal of XAI methodologies—to make AI more understandable to humans—and leave a detailed discussion of the differences among these terms for future work.

Regulation (GDPR) in Europe has mandated that artificial intelligence (AI) systems provide justifications for their decisions, further underscoring the critical role of XAI [32, 33].

XAI methodologies can be categorised along several dimensions, such as the type of explanations they produce, the ML models they are applicable to, and the specific tasks they aim to address. Proposed techniques frequently belong to multiple subcategories within these dimensions. For instance, a single method might generate both visual and conceptual explanations or be applicable to both traditional ML models and deep learning architectures. This overlapping nature reflects the versatility and multifaceted nature of XAI techniques, which are not limited to a single category but instead span across several, depending on their design and application. The next section presents the dimensions along which XAI methodologies are commonly categorised.

1.1 Overview of XAI Categorisation: Key Dimensions

This section explores six critical dimensions for categorising XAI methodologies: explanation phase, explanation level, modality of explanations, model specificity, target audience and granularity. These dimensions provide a structured framework for understanding how XAI techniques differ in their approaches and applications (refer to Figure 1.1 for an overview). This discussion also sets the stage for the modality-focused literature review presented in Chapter 2.

Explanation Phase. XAI techniques are typically categorised into two main types: post-hoc and ad-hoc methods. Post-hoc methods are applied after the model has been trained and offer explanations without altering the model structure or any train/test time intervention [34, 35]. These methods aim to interpret the decision-making process retrospectively, often using visualisations or perturbation techniques. In contrast, ad-hoc methods refer to models that are either inherently interpretable, such as linear models and decision trees; or that incorporate explainability directly into their architecture during training [36, 37]. While inherently interpretable methods often have limited predictive capacity, techniques that integrate explainability into black-box models typically experience a performance trade-off due to the structural train/test time interventions. Unlike post-hoc methods, which provide explanations after the fact, ad-hoc techniques offer immediate, by-product interpretability.

Explanation Level. XAI methods can also be distinguished by whether they provide global or local explanations. Global explanations aim to give a comprehensive view of how a model behaves across a wide range of inputs, offering insight into the overall functioning of the model. Local explanations, in contrast, focus on specific predictions, providing details on why the model made a certain decision for a particular input. In

Dimension 1 - Explanation Phase	
Post hoc (e.g., Saliency maps, LRP)	Applied after model training to explain decisions
Ad hoc (e.g., CBMs)	Built-in interpretability in model architecture
Dimension 2 - Explanation Level	
Global (e.g., CAVs, Post-hoc CBMs)	Provides a general understanding of model behaviour
Local (e.g., SHAP, LIME)	Explains individual decisions or predictions
Dimension 3 - Model Specificity	
Model specific (e.g., Grad-CAM)	Methods tailored to specific models like CNNs
Model agnostic (e.g., LIME, SHAP)	Can be applied to any type of model
Dimension 4 - Target Audience	
Researchers (e.g., interpret. tools)	Designed for model developers seeking insights
End users (e.g., interactive visuals)	Tailored for users needing output understanding
Dimension 5 - Granularity	
Fine-grained (i.e., concept level)	Provides detailed insights into specific parts of inputs
Coarse-grained (i.e., class level)	Offers a general understanding of model predictions
Dimension 6 - Modality of Explanation	
Visual (e.g., CAM, Occlusion)	Explanations provided visually, often as heatmaps
Conceptual (e.g., CAVs, CBMs)	Explanations provided as text e.g., concept names

FIGURE 1.1: XAI dimensions for various applications.

short, this category highlights the scale at which a model's behaviour is interpreted, whether explaining its behaviour in general or for individual instances.

Model Specificity. For this aspect, XAI methods are labelled as either model-specific or model-agnostic. Model-specific methods [38–41] are designed to work with particular types of architectures, such as convolutional neural networks (CNNs), utilising their unique architecture for explanations. Model-agnostic methods [36, 42–45], however, can be applied to any ML model, regardless of its structure, as they only require test inputs and the prediction function. This categorisation reflects the adaptability of XAI techniques, indicating whether they are specialised for certain architectures or can be broadly applied across different models.

Target Audience. XAI methods are designed to cater to different groups of users with varying needs. Some tools are tailored for researchers and developers who require deep insights into model behaviour to refine and improve model architectures. These tools often provide detailed, technical explanations and are useful during model development. On the other hand, there are methods aimed at end users who need to understand model outputs without delving into the underlying technical details. For

these users, interactive visualisations or simplified explanations are more appropriate, helping them trust and comprehend a system's decisions. Google's what-if tool [46], AI Explainability 360 toolkit by IBM [47] and H2O AutoML platform [48] are some user-friendly examples.

Granularity. Granularity refers to the level of detail provided by an XAI technique. In this direction, methodologies can be categorised as offering fine-grained or coarse-grained explanations. Fine-grained explanations, such as concept-level insights, delve into more detail of a model's decisions, offering feature-specific information [49, 50]. On the other hand, coarse-grained explanations, like class-level or object-level insights, offer a more general understanding of why a model made a particular prediction, typically explaining what contributed to the decision overall [34, 35, 39]. This category helps determine the level of precision or detail in the explanations, depending on the needs of the user. To give an example, for an animal classification task, a coarse-grained explanation would be a saliency map highlighting the class of interest for a test image, whereas a fine-grained explanation would generate more granular, concept-wise heatmaps highlighting different parts of the object in an input image alongside their textual descriptions.

Modality of Explanations. In this aspect, the distinction lies between the type of generated explanations. For visual tasks, there are two common modalities: visual and conceptual. Visual explanations often involve heatmaps that highlight important regions of an image, giving an intuitive, visual representation of what influenced the model's decision [42, 44]. Conceptual explanations, on the other hand, ideally offer explanations as human-understandable concepts, e.g., *stripe* for a *zebra* image [36, 41]. In Chapter 2, we delve deeper into these two modalities, offering more detailed insights and a review of the relevant literature surrounding both modalities. For each modality, we will explore key methodologies, drawing on existing research to highlight the strengths and limitations of these approaches. This examination also aims to provide a comprehensive understanding of the role each modality plays in enhancing AI interpretability and trustworthiness.

1.2 Research Motivation and Contributions

One of the most critical areas where explainability has become essential is computer vision. DNNs, widely deployed in tasks such as image classification, object detection, and semantic segmentation, have provided superior performance to the traditional methodologies while bringing together the mentioned transparency challenge. In high-stake domains where computer vision models are widely used such as autonomous driving [51] and medical diagnostics [52], it is not enough for these models to be accurate—they must also provide insights into how they reach their conclusions to ensure

trust and accountability. This need has driven the development of explainability techniques specifically designed for computer vision models, enabling users to interpret and visualise the inner workings of these models.

In this thesis, we explore XAI methodologies that are specifically designed for, or adaptable to, computer vision tasks. We identify their current limitations and propose novel approaches to address and overcome these limitations as detailed below.

Traditional visual explanation techniques, such as class activation mapping (CAM) [38], Grad-CAM [39] and layer-wise relevance propagation (LRP) [35], generate heatmaps that highlight the regions in an image most relevant to a model's prediction. Similarly, saliency maps [34] visualise the contribution of individual pixels to a model's decision, providing a detailed spatial understanding of what part of an image influenced the outcome. These methods are particularly useful for giving intuitive, visually interpretable feedback on a model's focus during inference. However, while they excel in visual attribution, they often fall short in offering semantic, high-level explanations detailing why certain regions or pixels were important in terms of human-understandable concepts like *object parts*, *texture* or *shape*.

In contrast, concept-based explanations such as concept activation vectors (CAVs) [36, 53] and concept bottleneck models (CBMs) [37, 41, 54] provide an alternative form of interpretability by linking model predictions to human-understandable concepts. For instance, CAVs allow a model to explain its predictions based on predefined, interpretable concepts, for instance identifying that an image contains an *antelope* because of the *horn* concept, rather than merely highlighting the body of the animal. Similarly, CBMs are designed to make predictions through an intermediate layer of predefined concepts. This ensures that the model uses human-recognisable features like *tumour size* or *inflammation* in medical diagnosis to make a decision. These approaches help users understand what high-level, human-like features are driving a model's decisions. However, while concept-based methods offer deeper semantic interpretability, they lack the spatial, pixel-level precision that visual attribution techniques provide. This makes them less suitable for tasks where it is important to know exactly which parts of an image were crucial for a model's decision, such as identifying a specific region in an X-ray image responsible for a medical diagnosis.

In summary, concept-based methodologies are better suited for applications where understanding the conceptual reasoning behind a prediction is critical, whereas visual attribution techniques are preferable when spatial localisation and visual interpretability are key. Despite this trade-off, the combination of both types of explanations could offer a more comprehensive understanding of a model's behaviour, particularly in fields like healthcare and autonomous driving, where both *what* and *where* are crucial for trust and safety. In this respect, we aim to bring together the two desired properties of XAI

and propose *human-understandable concepts alongside their pixel attribution maps*. The details of our contributions are as below.

- We propose a novel concept-based XAI methodology, *multilevel XAI*, which provides intuitive, multilevel explanations while maintaining reasonable annotation costs. This methodology generates human-like explanations for vision tasks in both linguistic and visual forms, without significantly compromising the predictive performance of DNNs. The explanations produced by multilevel XAI take the form of concept-wise saliency maps, highlighting image regions that strongly influence class predictions. Compared to conventional class-wise saliency maps, these multilevel explanations are more intuitive and easier to interpret. Additionally, they are more reliable than other concept-based explanations due to the inclusion of visual attribution maps as a by-product.

- Although our multilevel XAI approach qualitatively aligns many concepts with human interpretations, quantitatively validating this alignment remains challenging without ground-truth concept-wise segmentation maps, which are prohibitively expensive to collect. If such maps were available, quantitative evaluation could leverage the intersection-over-union (IoU)—an evaluation metric widely used for segmentation tasks—to compare ground truth with predicted concept maps. However, the challenge of collecting these maps led us to propose a novel weakly-supervised semantic segmentation methodology called *semantic proportions-based semantic segmentation (SPSS)*. This technique predicts segmentation maps without requiring per-pixel ground-truth segmentation maps, instead relying on significantly less information—namely, the proportions of semantic classes. Given that multilevel XAI produces explanations in the form of concept-wise saliency maps, we envision using these maps to estimate class proportions in future work. Due to time constraints, we have not pursued this further in this thesis. However, SPSS demonstrates that this direction holds promise for advancing weakly-supervised segmentation and improving explainability.

- Building on our multilevel XAI approach and SPSS framework, we identified a significant gap in the evaluation criteria and benchmark datasets within the concept-based XAI field. To address this, we propose three novel measures: the *concept global importance measure (CGIM)* the *concept existence measure (CEM)* and the *concept location measure (CLM)*, designed to evaluate the concept prediction and localisation capabilities of methodologies in this domain. Furthermore, we advocate for the use of the well-known `Caltech-UCSB Bird (CUB)` [55] dataset as a benchmark, leveraging its diverse range of labels. Through qualitative and quantitative analysis, we demonstrate that while concept-based methodologies

are widely used and recognised, their concept prediction capabilities are surprisingly limited. Moreover, their highly ranked concept outputs often fail to correspond to the correct regions in a given image, raising concerns about their reliability and safe application. We encourage the research community to adopt our proposed evaluation measures and the CUB as the benchmark dataset to establish a unified and fair standard for comparison.

1.3 Thesis Structure

The remainder of this thesis is structured as follows.

Chapter 2 presents an extensive review of existing literature related to XAI methodologies, with a particular emphasis on categorising these techniques into distinct subgroups based on the *modality of explanations*. Additionally, this chapter includes a concise overview of weakly supervised semantic segmentation (WSSS) methods, highlighting recent advancements and challenges within this domain.

Chapter 3 proposes a novel multilevel XAI methodology. This approach aims to enhance interpretability by generating concept-wise saliency maps, offering more granular insights into how individual features contribute to model decisions. The proposed methodology is thoroughly evaluated across various benchmark datasets, demonstrating its superiority over existing XAI techniques.

In Chapter 4, we present a new WSSS technique. This chapter explores how the proposed SPSS model can achieve effective semantic segmentation across multiple tasks where only coarse annotations in the form of class proportions are available. We detail the model architecture, training strategy, and extensive experiments showcasing its performance compared to other state-of-the-art methods.

In Chapter 5, we propose three novel evaluation criteria—CGIM, CEM and CLM—along with a benchmark dataset to establish a standardised framework for evaluating concept-based XAI methodologies. Additionally, we introduce the concept activation mapping (CoAM) framework, which addresses a critical gap in current concept-based XAI techniques: the lack of visual concept attribution. This chapter underscores the need for a more careful and rigorous design of concept-based XAI methodologies.

Finally, Chapter 6 provides a comprehensive conclusion of the thesis. We summarise the key contributions and findings, discuss the limitations and challenges encountered throughout the research, and propose potential future directions aimed at overcoming these challenges. This chapter also reflects on broader implications for the fields of XAI and weakly supervised learning, suggesting avenues for further exploration and innovation.

1.4 Publications

- Published
  - **Halil Ibrahim Aysel**, Xiaohao Cai, and Adam Prugel-Bennett. "Multilevel Explainable Artificial Intelligence: Visual and Linguistic Bonded Explanations." *IEEE Transactions on Artificial Intelligence*, 5(5), pp. 2055–2066, https://doi.org/10.1109/TAI.2023.3308555, 2023. [**Chapter 3**]
  - **Halil Ibrahim Aysel**, Xiaohao Cai, and Adam Prugel-Bennett. "Explainable Artificial Intelligence: Advancements and Limitations." *Applied Sciences,* 15, 7261, https://doi.org/10.3390/app15137261, 2025. [**Chapter 2**]
- Under Review/Revision
  - **Halil Ibrahim Aysel**, Xiaohao Cai, and Adam Prugel-Bennett. "Semantic Segmentation by Semantic Proportions." Under review in *Pattern Recognition.* [**Chapter 4**]
  - **Halil Ibrahim Aysel**, Xiaohao Cai, and Adam Prugel-Bennett. "Concept-Based Explainable Artificial Intelligence: Measures and Benchmarks." Under review in *Image and Vision Computing*. [**Chapter 5**]

Chapter 2

Literature Review

This chapter briefly explores the history of CNNs and examines their frequently cited "black-box" nature. It then reviews key XAI methodologies for computer vision tasks to offer a concise yet informative overview. This overview is based on the modality of explanations, distinguishing between visual and conceptual approaches. Finally, we delve into the WSSS (weakly supervised semantic segmentation) techniques, laying the groundwork for the SPSS methodology introduced in Chapter 4. This chapter complements the more specific literature reviews presented in this thesis' main chapters.

2.1 Black-box Nature of CNNs

First proposed in 1998 for handwritten character recognition [56], CNNs have shown a remarkable performance, especially in vision tasks. Thanks to the ImageNet dataset [14] and efficient GPUs that enabled thousands of computations in parallel, AlexNet by Krizhevsky et al., achieved state-of-the-art performance in ImageNet LSVRC-2012 competition [57]. Following AlexNet, several even more performant architectures have been proposed including but not limited to VGG [58], GoogleNet [59] ResNet [60], DenseNet [61], MobileNet [62], EfficientNet [63] and ConvNeXt [64, 65]. With these CNN-based architectures, human-level performance has already been reached and arguably outperformed in various tasks. However, CNNs' black box nature has been seen as the main obstacle to their further deployment, especially in fields where human life is at stake.

The so-called black-box nature of CNNs arises from their end-to-end learning process, which contrasts with traditional ML methods such as decision trees [66] and support vector machines (SVMs) [67] where features are manually designed by experts—for example, edge and texture detectors or colour histograms—tailored to specific tasks [68, 69]. As the decision process of traditional ML techniques is built around these

predefined, human-understood features, it is relatively easy to follow the reasoning behind their predictions. In contrast, CNNs automatically learn features directly from data through multiple layers of abstraction without any human intervention. This automatic feature extraction helps CNNs achieve state-of-the-art performance, but it also complicates our understanding of what the model is learning at each stage. The learned representations are highly complex and the decision process is non-linear which leads to an opaque decision-making that obscures the internal workings of the model [70].

Several methodologies were proposed to tackle these opaque predictions and make deep networks more transparent. In this direction, efforts to enhance the interpretability of CNNs have focused on two primary strategies: visualising model behaviour and aligning learned high-level features with human-understandable concepts.

Visualisation techniques aim to illuminate how CNNs build complex representations hierarchically from simpler ones, as early studies revealed that initial layers learn basic features like edges and lines, while later layers capture textures and patterns, culminating in the penultimate layer focusing on object parts—insights contributing to what [71] describes as *Algorithmic Transparency* [72–74]. Additionally, pixel attribution methods highlight specific regions of input images that most strongly influence model predictions, providing a clearer understanding of what drives specific decisions [34, 75].

On the other hand, feature-concept alignment methodologies seek to map high-level features learned by CNNs to human-interpretable concepts [36, 41, 76]. These approaches reconnect abstract representations of deep networks with predefined, semantically meaningful concepts, akin to the handcrafted features used in traditional models. By doing so, they aim to bridge the gap between the opaque, high-dimensional workings of CNNs and human intuition, making the decision-making process more transparent and understandable.

An ideal XAI methodology would seamlessly address both directions detailed above: it would not only match high-level features—automatically extracted during training—to human-understandable concepts but also localise these concepts spatially within an examined input sample. Such a methodology would provide a more holistic explanation, enabling users to understand not only what the model has learned but also where these concepts are represented in the input. This dual capability would allow for more intuitive and interpretable model explanations, improving trust and transparency in AI systems across various domains.

2.2 Visual Explanations

Visual explanations are arguably the most widely used XAI techniques for computer vision tasks as they offer a direct way to interpret model decisions by providing visual

representations of what a model makes its decisions based on. This category encompasses a wide range of methods, often focusing on how particular parts of an image contribute to a model's output. The goal of visual explanations is to make the inner workings of deep networks, especially CNNs, more transparent and interpretable.

2.2.1 Activation Maximisation

Visual explanations for CNN decisions can take the form of patterns that strongly activate a neuron, a feature map, an entire layer, or a predicted class—a process commonly referred to as activation maximisation. A notable methodology in this direction is random noise image optimisation. This approach starts with a random noise image, and the gradients of a specific unit are computed with respect to this image. The process identifies the patterns that most strongly activate the targeted unit, resulting in visionary pattern images that offer insights into the network's learned representations [73, 77–79]. Activation maximisation with random image optimisation, while useful for understanding which input patterns maximise the response of specific neurons or layers in black box models, comes with several limitations. One significant drawback is that the generated visualisations often lack interpretability and clarity, especially for higher-level neurons. The images produced tend to be highly abstract and sometimes visually unrealistic, making it difficult for humans to intuitively grasp what the model is focusing on. Additionally, they require numerous optimisation steps to generate meaningful activations, which makes them computationally expensive and hence their real-time or large-scale applications challenging.

An alternative approach, which eliminates these drawbacks, outputs real images instead. This is achieved by feeding the entire training set to the trained model and selecting a group of images that highly activate a specific unit [73, 80]. However, this approach places a significant burden on end-users, who must manually sift through highly activating images to identify common patterns—a process that is highly susceptible to human bias.

In addition to the mentioned drawbacks, activation maximisation techniques are also prone to adversarial artefacts—small changes in the input can drastically change the output activation without corresponding to any meaningful difference in the input image, undermining the reliability of the explanations. Overall, while activation maximisation offers insights into model behaviour, its limitations in clarity, computational cost, and susceptibility to noise reduce its practical utility in explainability.

2.2.2 Pixel Attribution

Pixel attribution techniques aim to find out image parts that contribute the most to their class predictions by creating heatmaps. These maps are used to mask input images to highlight specific parts crucial for a specific prediction. A heatmap is obtained by assigning an importance score to each pixel or a group of pixels. The intuition behind pixel attribution draws from the principles of linear models. In the case of a simple linear model where an input $\boldsymbol{x}$ with P features is classified by, say, a single neuron without any non-linear activation, the class score for a particular class c is computed as $S_c = \sum_i w_c^{(i)} x^{(i)}$, where $\boldsymbol{w}_c$ is a weight vector. This intuitively means that each feature in $\boldsymbol{x}$ contributes to the overall score based on its corresponding weight in $\boldsymbol{w}_c$. A feature with a higher score, i.e., $w_c^{(i)} * x^{(i)}$, has a greater impact on the class score, as S_c is a weighted sum of the feature values. Thus, features with higher values after being multiplied by their corresponding weights are considered more important in determining the class prediction. This simple idea is the foundation of pixel attribution methodologies, which assign a score to each pixel in an input image based on its contribution to the class prediction.

We can adapt the linear example above to a more complex image classification scenario achieved by CNNs. Let $\boldsymbol{x} \in \mathbb{R}^P$ denote an input image with P pixels, to be classified as one of the C classes. A trained model defines a mapping function $f : \mathbb{R}^P \rightarrow \mathbb{R}^C$, which generates a probability vector expressed as $f(\boldsymbol{x}) = [S_1, ..., S_C]$, where S_c represents the probability score for class c. Attribution methods assign a relevance score to each pixel in $\boldsymbol{x}$, denoted as $\boldsymbol{r}_c = [r_c^{(1)}, ..., r_c^{(P)}]$, where $r_c^{(i)}$ represents the relevance score of pixel i for class c. Obtaining these relevance scores helps us mask the input image accordingly and derive a saliency map [71]. As CNNs are complex architectures and include multiple layers with nonlinearities, the linear approach cannot be directly applied, i.e., obtaining $\boldsymbol{r}_c$ is not as straightforward as $\boldsymbol{w}_c$. However, various approximations have been proposed as detailed below.

2.2.2.1 Backpropagation-based saliency mapping

Backpropagation-based methods are widely used for saliency mapping [34, 72, 81]. In this direction, Simonyan et al., presented the first saliency mapping technique where they take gradients of a class score S_c for class c with respect to the pixels of the input image $\boldsymbol{x}$ [34]. These gradients result in a relevance score vector, $\boldsymbol{r}_c = \frac{\partial S_c}{\partial \boldsymbol{x}}$, with the same size as the input image. Higher positive relevance scores indicate important pixels for class c, while lower ones show insignificant ones. In addition, high negative values can be seen as an indicator of other classes or backgrounds. These relevance scores then can be used to weigh the pixels to create a saliency map [34, 71]. As Springenberg et al. present, the other two well-known gradients-based methods, deconvolution [72, 82]

and guided backpropagation, indeed follow the same process as Simonyan's approach apart from the way they backpropagate through the activation functions [81].

Integrated Gradients [83] provides a principled way to quantify the contribution of each input feature to a model's prediction. It addresses some of the limitations of the early gradient-based methods, which can be noisy or misleading due to local irregularities in the model's gradients. They compute attributions by integrating the gradients of the model's output with respect to the input along a straight path from a baseline (e.g., a black image or zero vector) to the actual input. Mathematically, it averages these gradients over multiple steps along the path, ensuring that the attributions are both robust, which refers to explanations remaining consistent when the input data or the model itself undergoes insignificant changes. For instance, explanations would be expected to stay the same when the pixel values of an input image change in a way that the predicted object protects its characteristics. The author claims that their approach satisfies desirable theoretical properties, such as completeness, which ensures that the sum of all attributions matches the difference between the model output for the input and the baseline. By providing a clear and interpretable mapping of input features to their contributions, Integrated Gradients has been recognised as an important technique in XAI research.

Deep learning important features (DeepLIFT) [84] is another backpropagation-based pixel attribution method that provides a robust framework for explaining the predictions of deep networks by comparing the activations of neurons to their reference activations. Unlike gradient-based methods, which can suffer from issues such as vanishing gradients or noise, DeepLIFT assigns contribution scores by tracking the changes in outputs relative to a baseline input. It propagates these contributions backwards through the network using a set of predefined rules to ensure consistency and efficiency. The key innovation of DeepLIFT lies in its ability to handle nonlinearities more effectively by considering both the input and the reference, allowing it to capture meaningful attributions even in complex networks. DeepLIFT is also claimed to satisfy the completeness property (the sum of attributions matches the difference between the model output for the input and the baseline) similar to the Integrated Gradients approach. In addition, it also holds the symmetry (equal changes in symmetric inputs receive equal attributions) property, which contributes to their robustness and safe use.

LRP (layer-wise relevance propagation) [35, 85–87] is a powerful explainability technique designed to interpret the decisions of DNNs by tracing back the contributions of individual input features to the model's output. LRP works by decomposing the prediction score and redistributing it layer by layer, from the output back to the input, using a set of conservation rules. This redistribution ensures that the total relevance is preserved at each layer, ultimately assigning relevance scores to input features in a way that reflects their contribution to the prediction. By doing so, LRP is claimed to

satisfy important properties such as relevance conservation and provides insights into how specific features, such as pixels in an image, influence the model's output.

Backpropagation-based XAI methodologies have been pivotal in unravelling the inner workings of complex neural networks. They offer a relatively efficient way to link input features to model predictions, enabling practitioners to gain insights into model behaviour and hence increase trustworthiness. These methods are particularly appealing due to their simplicity and adaptability across various neural architectures. However, their limitations are equally noteworthy. One key issue is their lack of precision, as they often produce coarse, blurry visualisations that highlight large, indistinct regions of the image, making it difficult to pinpoint exactly what features the model is focusing on. This can be particularly problematic when interpreting decisions in tasks like medical imaging or autonomous driving, where fine-grained details are critical. These methodologies also often suffer from instability and lack of robustness, with explanations being sensitive to minor input perturbations. They may also struggle to capture global model behaviour, instead focusing on local feature importance, which can lead to misleading or incomplete interpretations. Furthermore, the ones that employ gradients may be vulnerable to vanishing gradients and susceptible to noise. Additionally, saliency maps by backpropagation-based methodologies tend to be post-hoc explanations, meaning they provide insights only after a decision is made, which might not always reflect the true reasoning process of the model.

In summary, these shortcomings reduce the effectiveness of backpropagation-based methodologies in providing clear, consistent, and trustworthy interpretations, especially in high-stakes applications, which underline the need for complementary XAI approaches that provide more consistent and comprehensive explanations.

2.2.2.2 Class activation mapping (CAM)

Another group of methods, CAM, leverage the final convolutional layer of DNNs, which is shown to capture the most meaningful and complete object signals [74]. Unlike backpropagation-based approaches, CAM focuses on high-level features within these layers rather than tracing gradients or activations back to the input pixels. The first CAM method, introduced by Zhou et al., was designed for architectures incorporating a global average pooling (GAP) layer between the final convolutional and classification layers [38].

In this method, the GAP layer computes a scalar value F^k for each feature map $\boldsymbol{f}^k$, which is then passed to the classification layer along with weights $\boldsymbol{w}$. The class score for a particular class c is calculated as $S_c = \sum w_c^k F^k$, where w_c^k denotes the contribution of F^k—and by extension, the feature map $\boldsymbol{f}^k$—to the score for class c. Each feature map $\boldsymbol{f}^k$ in the final convolutional layer, prior to the GAP operation, is expected to highlight

the region that corresponds to the concept it represents when scaled by its calculated weight w_c^k.

For instance, in a face recognition task, a feature map activated by the nose might receive a high weight, while one responding to unrelated features, such as a car wheel, would be assigned a low or even negative weight. The weighted sum of these feature maps is then upsampled to the input image's resolution, producing a saliency map that highlights the most relevant regions for the examined class.

A key limitation of the CAM approach is that it can only be applied to CNNs with a GAP layer between the last convolutional and classification layers, such as GoogleNet [59]. For architectures lacking this structure, CAM requires modifying the model by adding a GAP layer after the final convolutional layer. However, this alteration necessitates model retraining, which can lead to performance degradation compared to the original model.

GradCAM [39] was introduced to overcome the limitations of CAM, particularly its dependency on GAP. As a generalisation of CAM, GradCAM eliminates the need for specific model types or architectural modifications, making it more versatile and broadly applicable. This is achieved thanks to the way GradCAM weighs each feature map; differently from CAM, it takes the gradients of the output score S_c for a given class c with respect to each feature map f^k at the last convolutional layer, i.e., $\frac{\partial S_c}{\partial f^k}$. After this process, k different gradient maps are obtained. Finally, by applying GAP, a single weight,

$$w_c^k = \frac{1}{Z} \sum_i \frac{\partial S_c}{\partial f_i^k}, \tag{2.1}$$

for each feature map is obtained where i indicates the location of each pixel and Z is the total number of pixels in the feature map f^k. Following that, similar to CAM, a weighted sum of the feature maps generates a heatmap with the same dimensions as f^k (also see Figure 2.1 for an illustration). ReLU is also applied to keep features that positively affect a given class and ignore the negative signals that probably are for the other classes. Lastly, acquired heatmaps are upsampled to the size of the input image to highlight important parts of it. In addition, the authors show how to get more class-discriminative and fine-grained results using simple element-wise multiplication between saliency maps generated with GradCAM and guided backpropagation [81].

More approaches were proposed to further refine the resulting activation maps of GradCAM. For instance, Chattopadhay et al. proposed the GradCAM++ method to generate better localisation and handle multiple instances in a single image [88]. The main difference of this approach from GradCAM is taking into account only positive partial derivatives of feature maps at the last convolutional layer. ScoreCAM [75], presented by Wang et al., gets rid of the dependency on gradients of the GradCAM method by masking input images according to feature maps to obtain perturbed images. Scores

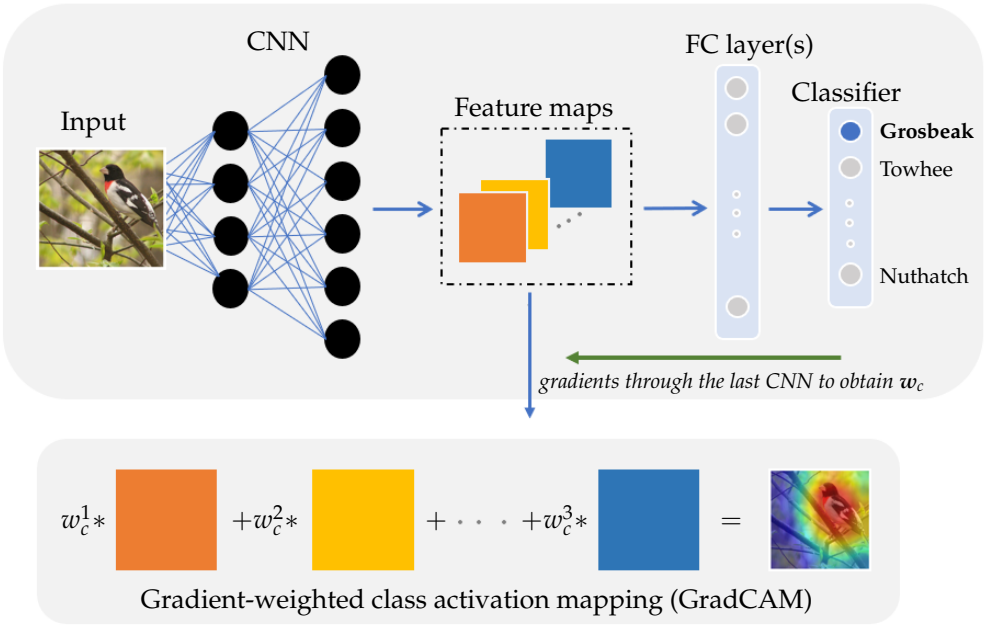


FIGURE 2.1: An overview of gradient-weighted class activation mapping (GradCAM) [39]. Gradients of the predicted class score with respect to the feature maps of the last convolutional layer are computed to obtain weights $\boldsymbol{w}_c$. These weights are used to linearly combine the feature maps, and the resulting weighted sum is used to mask the input image to obtain a saliency map, highlighting the regions most relevant to the prediction.

obtained by forward passes of these images are used as weights, for instance, w_c^k for feature map k. This method claims to reduce noise in the saliency maps caused by gradients.

Advanced CAM methods, including Grad-CAM and Grad-CAM++, have been widely adopted for visualising the important regions of an image that contribute to a model's decision, but they also come with notable limitations. One key issue is that while they can be applied to a broader range of architectures than traditional CAM (which requires GAP), the visualisations they produce are often still relatively coarse and low-resolution. This can obscure fine details that might be critical in tasks such as medical image analysis or detailed object recognition, where understanding subtle features is essential. Moreover, CAM-based methods generally highlight large regions of an image, making it difficult to differentiate between important and unimportant features within these highlighted areas. Additionally, CAM methods typically focus on explaining a model's final decision for a specific class, offering little insight into the intermediate layers or the broader decision-making process across the network. This narrow focus limits their ability to provide a holistic understanding of how the model processes and interprets input data. Overall, while CAM methods offer valuable insights, their lack of spatial precision and limited scope reduce their effectiveness in generating deep, fine-grained explanations.

2.2.2.3 Perturbation-based methodologies

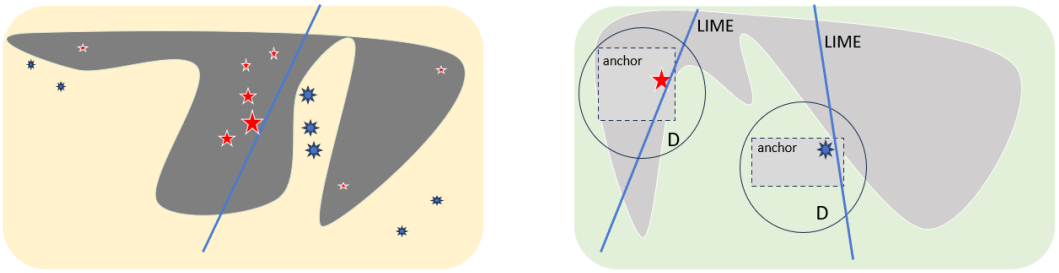
Another approach to pixel attribution involves perturbation-based methodologies. These methods use altered versions of an image to gain insights into a model's inner workings. For example, in the occlusion sensitivity technique by Zeiler et al., a portion of an image is replaced with a patch that contains either the average pixel value of the entire image or a uniform colour. Newly created perturbations are then passed through the trained model to see the effect of the occluded area on the prediction. Despite being as simple as blocking a part of an image, occlusion is shown to be an effective approach. To give an example, the authors showed that when they occlude a dog's head in a given image, the probability for the dog class drops significantly, which indicates that the head of the dog is crucial for the trained model when making that specific prediction [72].

In a similar methodology, RISE [45] employs an automated approach to generate perturbations for attribution maps. Given a trained CNN model $f : \mathbb{R}^{H \times W} \rightarrow \mathbb{R}^{C}$ that maps an input image $\boldsymbol{X}$ of size $H \times W$ to an output class c, RISE introduces a binary mask $\boldsymbol{M}$ sampled from a random distribution $\mathcal{D}$ and of the same size as $\boldsymbol{X}$. By performing element-wise multiplication of $\boldsymbol{M}$ and $\boldsymbol{X}$, a perturbed image is created with random occlusions at specific pixel locations. Repeating this process with multiple masks produces a set of perturbed images. The model's prediction probabilities for class c are then computed for each perturbed image and used as weights. Finally, a weighted sum of all perturbed images is used to generate an attribution map. The idea of occluding parts of an image or starting from a blank image and incrementally adding random patches (or pixels) to observe changes in the classification score for a given output class has inspired many model-agnostic methods, which we discuss next.

We now review some well-known model-agnostic methods, which fall under the category of perturbation-based pixel attribution techniques. These methodologies are not designed specifically for CNNs and can be applied to any trained ML model, as they only require the investigated input sample and the prediction function. Often referred to as "black-box" XAI techniques, these methods do not rely on access to the trained model's internal components, such as weights or feature maps, unlike most of the approaches reviewed so far.

One of the well-known model-agnostic techniques, local interpretable model-agnostic explanations (LIME) [42], is presented by Ribeiro et al., and followed by different variants such as G-LIME [89] and S-LIME [90]. The idea behind LIME is to approximate a trained complex ML model with a simpler linear model in the local vicinity of an input sample. In this context, LIME is based on the assumption that we introduced earlier in this section: linear models can be interpreted by evaluating the weights they optimise per feature. LIME works by first splitting an input image into superpixels which

are contiguous regions of similar pixels. It then perturbs the image by randomly turning some superpixels on and off to create variations of the original image. For each of these perturbed images, the distance to the original image is calculated to capture how much they differ. The original black-box model is then used to classify these perturbed images, and the predictions are recorded. Based on the distances and the corresponding predictions, LIME constructs a new, simpler linear model that approximates the behaviour of the original complex model in the local region around the investigated sample. Finally, the most important superpixels—those that have the greatest impact on the prediction—are highlighted as pixel attribution maps for the model's decision. By doing so, LIME helps identify which parts of the image were most influential for the model's decision.



(A) LIME. Adapted from [42]. (B) Anchors. Adapted from [43].

FIGURE 2.2: The difference between LIME and Anchors approaches: LIME fits the best possible linear line while anchors "guarantee" that almost all the examples satisfying the rules are from the same class. $\mathcal{D}$ is the perturbation space for both approaches, the straight blue line is the linear model that LIME proposes, and the dashed box is the local area that the Anchor method "anchors" the explanations.

To illustrate how LIME works, we present Figure 2.2a for a binary classification task. As shown, a complex pattern, highlighted in brown colour, is captured by the original trained model. The big red star is the instance to be explained, and other red stars represent the perturbed samples from the same class while the blue octagrams are the samples from the opposite class. The larger an instance, the closer it is to the data point being queried. Using the predicted classes from the original complex model for the perturbed instances, along with the calculated distances, the linear model (represented by the blue line) is optimised to approximate and explain the class prediction of the red star in its local vicinity. Although this new model does not reflect the global behaviour of the complex model, it is locally faithful; meaning that it reflects the behaviour of the complex model in the local area of the given instance. The weights of this linear line show the importance of each image patch in the input image [42].

One major issue that LIME and other linear approximation approaches face is to determine the neighbourhood in which the given explanations are valid. There is no specified way to decide which data points these explanations are applicable to. This is a vital issue as the coverage of explanations is not clear. In other words, explanations would be tricky and unstable when trying to predict unseen examples by relying on the

weights of the newly fitted model. An attempt to mitigate this limitation was made by the same authors in [43], where a novel method called Anchors: high-precision model-agnostic explanations is presented. The difference between Anchors and LIME can be seen in Figure 2.2b. Anchors aim to fix a local area where all the instances are from the same class to achieve consistent "anchored" explanations. In this way, explanations help users predict unseen data points with high precision and less effort. This approach is presented in the form of a rule-based system, and an example explanation may be,

- **if** you are *[younger than 50]* and *[female]*, **then** you are very unlikely to have cancer,

where *[age: under 50]* and *[gender: female]* are referred to as anchors. As long as these anchors are present in new input, it is highly likely to be predicted as belonging to the same class as samples sharing these feature values. These features are assumed to cause the classification and provided that they are held; changes in the rest of the features do not affect the prediction. Similarly, for an image classification task, the anchors may be,

- **if** the given image includes *[stripes]*, **then** it is almost guaranteed that the model will classify it correctly as a zebra,

and hence the super-pixels involving *[stripes]* are called anchors. Anchors aim to meet two main requirements: precision and coverage. The former is about how precise the explanations are, while the latter defines the local area where these explanations are applicable. For a zebra classification task, precision is the percentage of perturbed examples from the same class as the input image that also contains *[stripes]*, whereas coverage is defined as the fraction of the number of samples that hold the given anchor *[stripes]* to all perturbed instances [43].

SHapley Additive exPlanations (SHAP) by Lundberg et al. is another powerful and widely used model-agnostic XAI approach [44]. It is rooted in the concept of Shapley values from cooperative game theory, where the goal is to fairly attribute the contribution to each player in a game based on their impact on the overall outcome. Applied to ML, SHAP assigns a value to each feature, representing its contribution to the final prediction for an individual instance. What makes SHAP particularly appealing is its theoretical rigour and its ability to provide consistent, model-agnostic explanations, making it highly versatile across different types of ML models, from linear to deep learning architectures. This versatility is achieved through several specialised variants designed to enhance both efficiency and accuracy depending on the model type. For instance, Kernel SHAP is the most general version, allowing SHAP values to be computed for any ML model. However, it relies on a kernel-based approximation technique that samples many combinations of feature subsets, which can be computationally intensive, particularly for high-dimensional data. Despite its flexibility, Kernel

SHAP's high computational cost often limits its scalability, especially for large datasets or complex models.

To address these performance limitations, model-specific variants of SHAP have been developed, with Tree SHAP [91] standing out as a prominent example. Tailored for tree-based models like decision trees, random forests, and gradient boosting machines, Tree SHAP leverages the structure of these models to compute exact SHAP values much more efficiently than Kernel SHAP. This makes it an ideal choice for structured data, where tree-based models are often preferred due to their predictive power and interpretability. In a similar vein, Deep SHAP combines SHAP values with DeepLIFT [84], to provide efficient explanations for deep learning models. By utilising backpropagation, Deep SHAP makes it feasible to explain complex neural networks in a computationally efficient manner, which is critical for tasks involving unstructured data like images or text.

These various SHAP variants highlight the method's remarkable adaptability across different ML models, which helped SHAP to become one of the cornerstone techniques in XAI. Its versatility in offering both local explanations—detailing the contributions of individual features to specific predictions—and broader global insights—captured through aggregations across the entire model—makes it an invaluable tool for diagnosing model behaviour, ensuring fairness, and promoting accountability in ML systems.

Perturbation-based XAI methodologies, such as SHAP and LIME, have gained considerable traction for their flexibility and ability to provide model-agnostic explanations. These methods excel at generating intuitive local explanations by estimating the contribution of individual features to a model's predictions, making them valuable tools for interpreting complex ML models. Their general applicability across different model types is a significant strength, allowing practitioners to apply them to a wide range of tasks without requiring specific model architecture knowledge.

However, these methodologies are not without limitations. One significant challenge is their computational complexity. Estimating feature attributions often requires multiple model evaluations for various perturbed inputs, which can become infeasible for high-dimensional datasets or computationally expensive models. Another drawback is their reliance on assumptions about the data, such as feature independence, which is rarely true in real-world datasets. This can lead to misleading or less accurate interpretations in the presence of feature correlations. Additionally, while perturbation-based methods are effective at providing local explanations, deriving consistent global insights can be difficult, especially when local attributions vary widely across instances. Finally, being model-agnostic, these approaches might not fully capture the internal mechanics or nuances of a model, potentially limiting their utility in domains requiring a deep understanding of a model's workings.

2.3 Conceptual Explanations

While visual explanations by activation maximisation and pixel attribution techniques are widely employed for computer vision tasks, there is also a growing interest in conceptual forms of explanations. Methodologies proposed in this direction aim to provide a high-level understanding of how models make decisions by linking their internal representations to human-interpretable concepts. This is particularly useful when the goal is to explain complex models in terms of concepts or features that align with human reasoning.

Network dissection [49] is a systematic methodology for interpreting the internal representations of neural networks by quantifying how individual neurons encode specific human-interpretable concepts. It provides a way to analyse the emergent structure within deep networks, particularly CNNs. The technique involves mapping the activation patterns of neurons to a predefined set of semantic concepts derived from labelled datasets, such as object categories, textures, or scenes. By evaluating the alignment between neuron activations and these concepts, network dissection enables researchers to understand the role and specialisation of neurons in the decision-making process of a model. This method is significant in demystifying the "black-box" nature of neural networks, providing insights into their interpretability, which are critical for debugging, enhancing trustworthiness, and identifying biases in AI systems.

Despite these advantages, network dissection has significant limitations. A key drawback is its reliance on segmentation maps for concepts, which are costly and time-consuming to create, especially for large and diverse datasets. This dependence can introduce biases and restrict the scope of analysis to predefined, segmented concepts, potentially overlooking emergent or complex features not present in the available segmentation maps. Additionally, network dissection focuses on individual units, which may fail to account for the collective behaviour of multiple units crucial for encoding higher-level abstractions. Lastly, it is better suited for static, pre-trained models and may struggle to offer insights for fine-tuned or dynamically evolving architectures.

Concept whitening [76] is another well-known technique. It works by modifying the latent space of neural networks to improve interpretability and disentanglement of learned representations. It introduces a specialised transformation layer into the network, which ensures that specific latent dimensions are decorrelated and aligned with human-understandable concepts. This is achieved through a whitening process that removes redundancy among features, making each dimension orthogonal to others, while simultaneously associating them with predefined semantic concepts. By explicitly controlling the latent dimensions to represent interpretable concepts, concept whitening facilitates understanding and debugging of model behaviour, and can potentially lead to better generalisation by reducing overfitting to irrelevant correlations in the data.

CAVs (concept activation vectors), presented by Kim et al., represent one of the key methodologies in the concept-based explanations realm [36]. By defining a set of high-level concepts, such as "striped" or "spotted" for, say, animal classification, CAVs measure the alignment of these concepts with the model's internal representations and allow researchers to examine how a deep learning model responds to these human-interpretable concepts. This is achieved by calculating directional derivatives of model predictions with respect to the predefined concepts at an intermediate layer of the examined model. This enables the generation of more meaningful explanations in the form of human-understandable concepts compared to the CNNs' learnt high-level features or highlighted coarse-grained object locations. CAVs have been particularly valuable in medical imaging where model domain-specific explanations are crucial [92, 93].

Crabbe et al. [94] proposed the concept activation regions (CARs) which extends the CAVs approach to address its limitations in modelling scattered concept examples in a DNN's latent space. Unlike CAVs, which assume that concept examples align with a single direction in the latent space, CARs represent concepts as regions encompassing multiple clusters. This representation uses the kernel trick and support vector classifiers to define concept activation regions. CARs enable describing how concepts relate to DNN predictions and also show how specific features correspond to concepts. Additionally, CARs demonstrate the potential for DNNs to autonomously identify established scientific concepts, such as grading systems in prostate cancer analysis.

Presented as an extension to the earlier idea [95, 96] of first predicting the concepts and then using the predicted concepts to predict a final target, CBMs (concept bottleneck models) [41] offer a compelling approach within the realm of conceptual explanations in XAI. These models are designed to provide interpretable decisions by incorporating human-understandable concepts into their decision-making process. In a CBM, the model first predicts a set of predefined concepts—such as "texture", "shape", or "tumour size"—which serve as high-level, interpretable features. These predicted concepts are then used as inputs to make the final decision. By doing so, CBMs allow users to directly inspect the model's reasoning at an intermediate level, offering transparency into how each concept contributes to the outcome. This structured approach not only makes the decision-making process more understandable but also allows for interventions at the concept level, enabling users to correct mispredicted concepts before they affect the final output. In this direction, there have been several works exploring efficient ways to achieve systematic interventions and model corrections [54, 97–99].

CBMs differ from CAVs in how they treat concepts. While CBMs are trained with the explicit goal of predicting and utilising predefined concepts as an integral part of their decision pipeline, CAVs operate in a post-hoc manner. CAVs are used to probe and analyse a trained model's internal representations to measure how well these representations align with specific concepts, without requiring the model to explicitly predict

those concepts during training. In other words, CAVs extract and quantify the influence of concepts after the model has been trained, while CBMs actively incorporate and rely on concepts throughout the learning process. This distinction gives CBMs a unique advantage when it comes to interpretability and control, as they offer built-in transparency and allow for direct correction at the concept level, enhancing both the interpretability and robustness of the model. Several extensions were introduced to improve CBMs [100–104].

One key drawback of CBMs is their high cost as they require manual concept annotation for every training image. This process can be time-consuming and resource-intensive, especially when working with large datasets. One solution is proposed in Chapter 3 where class-wise attributes were used to train CBMs which significantly reduced the annotation cost. Another solution is to integrate CAVs [36] in CBMs, which require only a set of positive and negative examples per concept, making them less annotation-intensive compared to the full concept annotations required by CBMs. Moreover, CAVs can even be derived from a completely different dataset to form *concept banks*, which then can be employed to explain models trained with related datasets, making CAVs more flexible and scalable.

In this context, post-hoc CBMs [37] achieve this integration to create a more efficient and controlled approach. They work by leveraging CAVs to obtain concept values, eliminating the need for intermediate concept predictions. Following that, a single layer is introduced as a bottleneck inspired by CBMs to map the concept values to the final classes. By doing so, post-hoc CBMs maintain the model intervention property of traditional CBMs while avoiding the high costs of concept annotation for every image, thanks to CAVs. This approach efficiently brings together the flexibility of CAVs and the structured decision-making process of CBMs, allowing for more scalable and transparent model analysis. Even though post-hoc CBMs were shown to be effective as a global explicator via model editing experiments, their concept prediction and localisation abilities are under-explored. This is due to post-hoc CBMs not being trained to explicitly predict the concepts unlike traditional CBMs, which hinders the possibility to do any direct evaluation on individual concept predictions.

Another methodology that aims to reduce the concept annotation cost, the CounTEX framework, connects image classifiers with textual concepts, leveraging a multi-modal embedding space, such as that provided by CLIP [105], to generate counterfactual explanations. It aims to explain classifier decisions by identifying and quantifying the contribution of specific, human-interpretable concepts derived from text. By mapping between the latent spaces of the target classifier and the CLIP model, CounTEX creates a projection mechanism to explain both correct and incorrect classifications in terms of these textual concepts. This approach aims to address the challenge of concept-annotated datasets requirement by utilising text-driven concepts [106].

An essential aspect of concept-based interpretability is the accurate prediction and localisation of highly important concepts. For example, if an animal classifier identifies an image as an *antelope* and indicates the *horn* as a critical concept for this prediction, the *horn* should not only be present in the input image but also activate the network in a way that aligns with the image region containing the *horn*.

While several methodologies discussed earlier provide relatively efficient solutions for concept-based explanations, their outputs are typically limited to a single level of abstraction. This means they lack the visual components necessary to highlight the precise regions corresponding to concepts in a given image. Furthermore, there is a noticeable absence of standardised evaluation metrics and benchmark datasets, making it challenging to compare these methodologies and comprehensively assess their strengths and limitations.

This thesis addresses these gaps in concept-based explanation methodologies. In Chapter 3, we introduce multilevel XAI, a method that generates both concepts and their corresponding heatmaps within the input image. To address the lack of evaluation tools, Chapter 5 proposes novel metrics for concept prediction and localisation while advocating for the use of an existing dataset, `Caltech-UCSD Birds (CUB)` [55], as a benchmark to evaluate concept-based explanation methodologies. These contributions enable more robust comparisons and comprehensive assessments of their performance.

2.4 Weakly Supervised Semantic Segmentation

Semantic segmentation is a core task in computer vision, focused on assigning semantically meaningful labels to every pixel in an image to identify specific objects. This task has diverse applications, including autonomous driving [107, 108], scene understanding [109], and medical image analysis [110, 111]. By enabling machines to extract detailed semantic information from images, it brings them closer to human-like visual perception. However, the inherent complexity and variability of real-world scenes, combined with the substantial need for labelled data to train deep learning models, make semantic segmentation a challenging problem. To tackle these difficulties, researchers have proposed various methods, such as fully convolutional networks (FCNs) [112], encoder-decoder architectures [113, 114], and attention mechanisms [115]. These approaches have driven significant progress in semantic segmentation, establishing it as a vibrant and rapidly evolving research field.

Despite these architectural advancements, the requirement of pixel-wise segmentation maps remains a significant challenge for successful semantic segmentation applications. WSSS approaches have been proposed to address this expensive per-pixel annotation challenge by leveraging less detailed coarse annotations, such as image-level labels, bounding boxes, scribbles, or points, to significantly reduce annotation efforts.

Depending on the granularity and type of available annotations, weak supervision can be categorised into these types, each offering unique advantages and challenges that influence the methods and performance of WSSS approaches.

XAI has also emerged as a promising avenue for providing weak supervision in this context, offering significant potential for reducing reliance on fully annotated data. Techniques like Grad-CAM [39], LRP [35], and Integrated Gradients [83] generate saliency maps that highlight key regions contributing to a model's predictions, making them valuable for WSSS. These outputs can serve as pseudo-annotations, guiding segmentation models by localising salient object parts. Hybrid approaches that combine XAI-generated pseudo-labels with weak annotations, such as image-level labels or bounding boxes, further enhance segmentation accuracy while minimising annotation costs. By aligning model interpretability with segmentation tasks, XAI fosters more transparent and explainable WSSS pipelines, bridging the gap between coarse annotations and precise segmentation, and ultimately reducing the need for dense supervision.

Below, we explore methodologies that utilise various levels of annotations, ranging from image-level labels to saliency maps generated by XAI techniques, as forms of weak supervision for semantic segmentation tasks. These methodologies were shown to be effective in outputting segmentation maps while keeping the annotation cost relatively low.

Image-level labels indicate the presence of classes in an image without providing spatial information. They are the weakest form of supervision, as they lack localisation details. Methods using image-level labels often begin with CAM to localise discriminative regions associated with each class. While CAM-based approaches are computationally efficient, they tend to focus on the most salient parts of an object, leading to incomplete segmentations. Recent advancements aim to overcome these limitations through seed refinement, self-training, and affinity propagation, enabling broader and more accurate object coverage [116–118].

Bounding box annotations provide a coarse spatial indication of object locations, offering a balance between annotation cost and localisation precision. Early methods like BoxSup [119] refine segmentations iteratively within the boundaries defined by the boxes. Contemporary approaches integrate box constraints into deep learning models, achieving better spatial precision by aligning predicted masks with bounding box edges and leveraging region-based loss functions [120, 121].

Scribbles and points serve as sparse spatial supervision, offering minimal yet explicit guidance about object locations. Scribble-based methods propagate annotations to unmarked regions using edge detection, graph-based propagation, or energy minimisation techniques [122–124]. Point-based methods rely on a few annotated pixels per object, often integrating these with background priors and unsupervised techniques to complete the segmentation [125].

Several studies have demonstrated the use of XAI outputs as weak supervision for semantic segmentation, effectively bridging the gap between limited annotations and high-quality segmentation models [126, 127]. For instance, HiResCAM-generated heatmaps have been used to refine annotations, enhancing segmentation models by highlighting relevant regions, particularly in complex image scenarios [128]. These approaches have been especially impactful in fields like medical imaging, where XAI-driven heatmaps help identify regions of interest for weakly supervised segmentation, reducing reliance on dense pixel-level annotations while improving model accuracy [129]. This integration of interpretability and weak supervision demonstrates the dual benefit of enhanced transparency and reduced annotation efforts.

Core approaches in WSSS revolve around generating and refining pseudo-labels, designing robust loss functions, and leveraging auxiliary information to improve segmentation accuracy. Loss functions are tailored to handle weak supervision, employing strategies to encourage confident predictions, and consistency regularisation to ensure stability under perturbations. Additionally, auxiliary cues, like saliency maps and edge detectors, help strengthen spatial information and guide model training.

The evaluation of WSSS methods relies on benchmark datasets and performance metrics that capture segmentation quality. Popular datasets, such as PASCAL VOC [130], MS COCO [131], and Cityscapes [132], provide a range of challenges in terms of object diversity, scale, and complexity. These datasets often include subsets tailored for weak supervision, with annotations like image-level labels or bounding boxes. The primary metric for assessing WSSS methods is Intersection over Union (IoU), which quantifies the overlap between predicted and ground-truth masks. To complement IoU, metrics like boundary accuracy and object localisation precision are sometimes used to provide finer insights into model performance. As WSSS research advances, the development of more diverse datasets and evaluation criteria will play a crucial role in driving progress and ensuring the robustness of these methods in real-world scenarios.

WSSS has emerged as a promising approach to address the high annotation costs associated with fully supervised methods, providing high-quality segmentation with minimal supervision. By utilising diverse forms of weak annotations, WSSS effectively balances annotation efficiency with segmentation performance. However, it still faces significant challenges such as localisation bias, where methods like CAM focus on the most discriminative object parts while neglecting less salient regions, and noise in pseudo-labels, which can propagate errors during training. Scalability and generalisation also remain critical concerns, as models often struggle to adapt to diverse datasets and real-world scenarios. To address these challenges, recent trends include the adoption of transformer-based architectures [133] for capturing global and contextual information, contrastive learning to enhance pixel-level discrimination and robust strategies for mitigating pseudo-label noise. Additionally, hybrid supervision strategies, combining multiple weak annotations, and domain adaptation techniques are

gaining traction to improve generalisation across varied domains. As research continues to evolve, these advancements position WSSS as a key enabler for robust and scalable segmentation solutions across diverse applications.

Chapter 3

Multilevel XAI: Visual and Linguistic Bonded Explanations

Applications of DNNs are booming in more and more fields but lack transparency due to their black-box nature. XAI is therefore of paramount importance, where strategies are proposed to understand how these black-box models function. The research so far mainly focuses on producing, for example, class-wise saliency maps, highlighting parts of a given image that affect the prediction the most. However, this method does not fully represent the way humans explain their reasoning and, awkwardly, validating these maps is quite complex and generally requires subjective interpretation. In this chapter, we conduct XAI differently by proposing a new XAI methodology in a multilevel (i.e., visual and linguistic) manner. By leveraging the interplay between the learned representations, i.e., image features and linguistic attributes, the proposed approach can provide salient attributes and attribute-wise saliency maps, which are far more intuitive than the class-wise maps, without requiring per-image ground-truth human explanations. It introduces self-interpretable attributes to overcome the current limitations in XAI and bring the XAI closer to a human-like explanation. The proposed architecture is simple in use and can reach surprisingly good performance in both prediction and explainability for DNNs thanks to the low-cost per-class attributes.

Our work has the potential of gaining end users' trust in DNNs and making it possible to answer "why" by creating human-like explanations. Future applications could include sensitive fields where practitioners are desperate to understand how black-box models decide on a specific prediction before their deployment. A prominent example is medical imaging where it is sine qua non to see how DNNs make decisions. Our technique could help domain experts trust the automated system they get help from. This is achieved differently from currently available techniques that can only highlight the part of an image that DNNs seem to rely on. We argue that self-explainable DNNs are the future of ML applications. As DNNs are currently the most preferred techniques

and their most apparent limitation is the complicated decision process, we bring about a novel and cheap technique that, to the best of our knowledge, has never been proposed before.

3.1 Introduction

Recent developments in computational resources with a significant rise in data size have led DNNs, such as multilayer perceptron (MLP) and CNNs, to be widely used in various tasks, such as image classification. Despite their excellent performance in prediction, DNNs are seen as black boxes as their decision process generally includes a huge number of parameters and nonlinearities [19, 31, 72]. The lack of explanation in these black boxes hinders their direct implementation in important and sensitive domains such as medicine and autonomous driving, where human life may directly be affected [24, 134, 135].

An example would be the DNNs trained to detect coronavirus. Although many works have been conducted and claimed to have a high predictive performance in detecting COVID-19 cases, a Turing Institute's recent report [136] disappointingly found that AI used to detect coronavirus had little to no benefit and may even be harmful, mainly due to unnoticed biases in the data and its inherent black-box nature (also see e.g. [137]). Another example is a woman who was hit and killed by an autonomous car. An investigation showed that the death was caused by the incapability of the car in detecting a human unless they were near a crosswalk [138]. In addition to these life-related examples, there are plenty of others where bias in training data or the model itself causes unwanted discrimination that may immensely affect people's lives. Amazon's AI-enabled recruitment tool is an example of how discriminative these models could be by only recommending men and directly eliminating resumes including the word "woman"; the company later announced that this tool had never been used to recruit people due to the detected bias [139]. These examples clearly show that for ML models to gain acceptance, it is critical to be able to reason why a certain decision has been made to prevent any unwanted consequences.

Explanations delivered by XAI can help ML practitioners debug their models by for example investigating the misclassification cases [33] and detecting bias in data [63]. There have been several works in this context to reveal the reasoning of the black-box models [34, 38, 42, 43, 45, 81, 88, 141]. However, the most widely used technique, creating class-wise saliency maps (e.g. see left of Figure 3.1) to indicate the areas that contribute to the prediction the most, have severe innate limitations. The first is the validation process of these maps, which is mostly qualitative or requires labour-intensive object-wise annotations [142, 143]. A recent study in [125] showed that full supervision of object segmentation by humans takes around 78 seconds per instance while

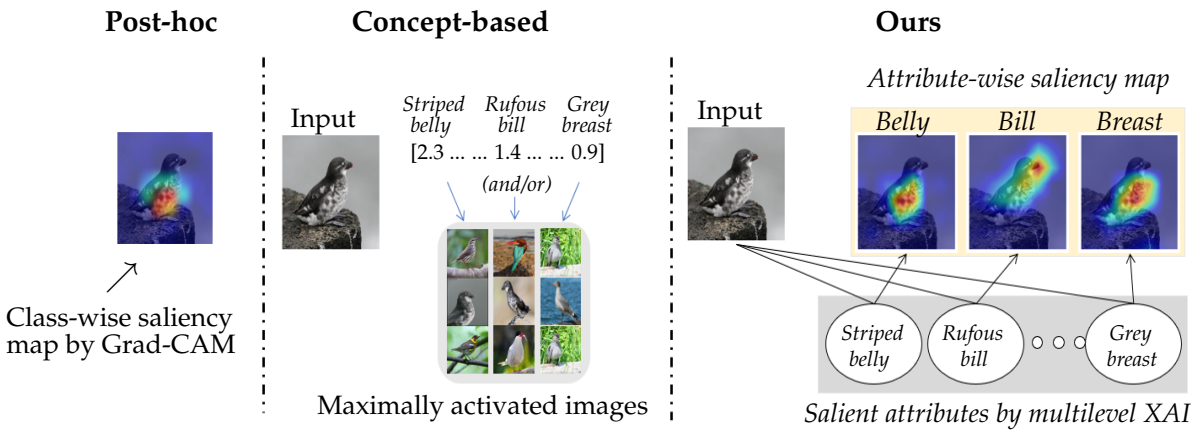


FIGURE 3.1: Explainability of the proposed multilevel XAI model. A bird image from the Least Auklet class is predicted correctly by our approach, with human-like multilevel explanations via salient attributes (e.g. "striped belly") and the corresponding attribute-wise saliency maps (*right*). Result by Grad-CAM [39] (*left*) and concept-based models such as [41, 140] (*middle*) are also given for comparison.

higher error rate bounding boxes take 10 seconds per instance to produce, which are much more expensive than 1 second per instance image-level annotations. Moreover, requiring a higher level of annotation by experts is rather impractical. Another limitation stems from the discrepancy between these maps and human-like explanations. Humans naturally explain their reasoning using discriminative words (e.g. domestic *vs* wild or weak *vs* strong to differentiate a cat from a lion) together with pointing to where those words lie in the given image if visually permitting [142, 143] (*cf.* our results on the right of Figure 3.1). To produce human-like explanations, this multilevel (i.e., visual and linguistic) manner is crucial, which also inspires the work in this chapter.

In this chapter, we propose a new methodology called *multilevel XAI* to delve into DNNs by leveraging visual and linguistic attributes. Our approach exploits per-class attributes (rather than per-image attributes, which are too expensive and generally impractical) to interpret DNNs in e.g. classifying raw images. By creating multilevel explanations, i.e., linguistic salient attributes and attribute-wise saliency maps, our method can provide explanations close to those we might expect from humans (e.g. see the right of Figure 3.1). This is a big step forward in XAI and this new methodology does not suffer from the above-mentioned limitations existing in current XAI solutions. The proposed setting adds a small extra cost to the training set, i.e., per-class attributes, which can be easily obtained if needed using for example online search engines or some autonomous tools (e.g. GPT-3 API [144]), and once acquired they can always be in use since in most cases they are time and image invariant.

Our main contributions lie in: i) proposing a multilevel XAI methodology which is easy to use and can achieve near human-like explanations; ii) implementing extensive

experiments on both coarse-grained and fine-grained datasets to validate the performance of the proposed approach; and iii) conducting insightful discussions in XAI and future paths.

The rest of the chapter is organized as follows. Section 3.2 presents the related work in the XAI field. Section 3.3 details the methodology of our proposed multilevel XAI. We provide the specifics of the datasets used in the experiments in Section 3.4. The experiment details and the results are given in Section 3.5. In Section 3.6, we raise a number of general research questions regarding XAI and discuss how much our technique, together with its limitations, addresses them. Finally, we conclude in Section 3.7. Pseudo-codes and further experiments are also provided.

3.2 Related Work

The complexity of ML models generally affects the transparency/explainability because of the difficulty of following the model prediction process [33]. One line of research is where researchers employ inherently explainable models and utilise white-box models such as Bayesian rules [145] and linear models [146] to handle complex problems. These models, generally, struggle to reach the prediction ability of DNNs.

3.2.1 Post-hoc Approaches

Methodologies in the XAI field mainly aim to propose methods to understand how high-performance black-box machine/deep learning models work. The majority of XAI methods introduced ideas to explain pre-trained models in a post-hoc manner, i.e., they are neither interested in the training setting nor in changing any of the models' components. These methods could be model-independent requiring only prediction function [42, 43] or model-dependent that need additional information of the trained model such as feature maps at a certain layer [38] or gradients [39]. DNNs for visual tasks do not output any textual justification. Modern visual-language models are effective in describing image content but lack outputting discriminative features that cause the prediction [142]. Forcing these models to output more discriminative features is one related work proposed in [143]. It aims to output multilevel explanations for vision-language tasks, e.g., visual question answering and activity recognition. Apart from a completely different focus against the work in this work, this method also requires labour-intensive per-image annotations during training that are avoided in our work.

3.2.2 Ante-hoc Approaches

More recently, there have also been attempts to train self-explainable models, also known as ante-hoc approaches. They can output explanations alongside their predictions, hence eliminating the need for any post-hoc design. Most of the state-of-the-art methods for visual recognition tasks are based on the parametric softmax function which projects latent features to the class space. One line of research presents methodologies based on non-parametric distance-based learning in the latent space and eliminates the use of softmax projection, making the decision processes of DNNs more human-understandable. These methods cluster training images in the latent space to obtain class centroids and then classify test images based on their distances to these centroids [147, 148]. As our classifier is pre-trained with attribute-class information and is frozen during the X-MLP and X-CNN training (see the detail in Section 3.3.2), we share the main aim of these methodologies towards more understandable predictions – putting effort into modelling the latent data structure. Explanations by distance-based methods are achieved by constraining the class centroids to be samples from the training set, and predictions are claimed to be inherently explainable as class centroids (i.e., real observations from the training set) can be displayed as a reason for predictions.

CBMs [41] share ideas with our work and are analogous to X-MLP, but neither of them is multilevel. In particular, unlike CBM, our X-MLP and X-CNN only require class-wise attributes, ensuring our approach is significantly cheaper to implement. Similar to CBM in terms of being analogous to our X-MLP, the work in [140] proposed a framework that can leverage concepts in different levels of supervision scenarios but with more storage and training capacity requirements. Explanations by these existing ante-hoc approaches are pre-defined concepts (i.e., meaningful words such as stripes), non-defined concepts (e.g. concepts $1, 2, 3, \ldots$), and/or the images that maximally activate these concepts (see also the middle of Figure 3.1). In contrast, our explanations are multilevel, possessing the advantage over the ante-hoc approaches mentioned above in terms of being capable of providing a spatial location in individual images associated with each linguistic attribute (see Figure 3.17 for an example that presents the significance of our multilevel explanations).

The zero-shot learning regime is where side information (e.g. attributes and class taxonomies) is exploited to classify images of classes that have no labelled samples during training [149]. The aim is to match image features with class attributes and then to classify unseen classes thanks to the prior side information. There are various techniques to find the best match that allow unseen class predictions [150–153]. Although we integrate side information into our training process similar to zero-shot learning, we are not interested in unseen classes; instead, the proposed work aims to train self-explainable models in many-shot case. Unlike the majority of XAI methods, our explanations are multilevel outputting both linguistic and visual explanations. Finally,

different from other extremely limited number of multilevel attempts that specifically work on vision-language models, our training setting is significantly cheaper and does not require per-image annotations.

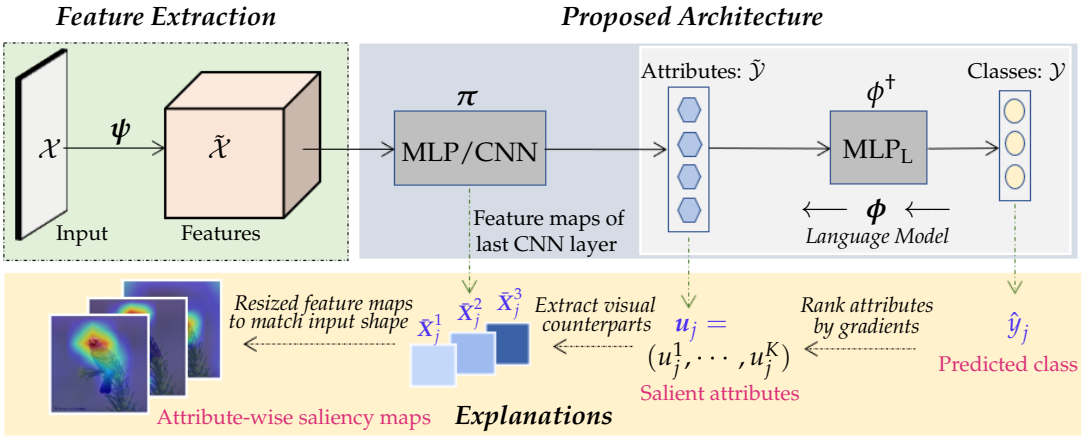


FIGURE 3.2: Multilevel XAI architecture. Image features $\tilde{\mathcal{X}}$ are extracted from images $\mathcal{X}$ using feature extraction model $\boldsymbol{\psi}$. Class labels $\mathcal{Y}$ are embedded into class attributes $\tilde{\mathcal{Y}}$ using language model $\boldsymbol{\phi}$. DNN $\boldsymbol{\pi}$ (e.g. MLP and CNN) is then trained to match $\tilde{\mathcal{X}}$ with $\tilde{\mathcal{Y}}$. The explainability of DNN $\boldsymbol{\pi}$ for image $\boldsymbol{X}_j$ is by the obtained salient attributes $\boldsymbol{u}_j$ (linguistic) and the attribute-wise saliency maps $\bar{\boldsymbol{X}}_j^i$ (visual).

3.3 Methodology of Multilevel XAI

In this section, we introduce our multilevel XAI methodology, see Figure 3.2 for its main architecture. It consists of three main components: i) a pre-trained feature extraction block generating high-level image features from input images (left of Figure 3.2); ii) a self-explainable DNN block bridging the extracted features with linguistic attributes (middle of Figure 3.2); and iii) a language model block (being frozen after training) linking the linguistic attributes to the output class labels (right of Figure 3.2). All of these blocks are important and are well studied in various fields, yet in the XAI regime, their study is rather limited. To the best of our knowledge, this is the first time they have been used to explain neural networks in a multilevel (i.e., visual and linguistic) manner particularly when the per-image attributes are unavailable. Further description is given below.

Preliminary. Let $\mathcal{X}$ be the set of images and $\mathcal{Y} = \{1, 2, \cdots, C\}$ be the set of C class labels. Let $\mathcal{S} = \{(\boldsymbol{X}_i, y_i) \mid \boldsymbol{X}_i \in \mathcal{X}, y_i \in \mathcal{Y}, i = 1, 2, \ldots, N\}$ be a training set with N image/label pairs, where y_i is the ground truth label of image $\boldsymbol{X}_i \in \mathbb{R}^{M_1 \times M_2 \times M_3}$ with M_3 set to 1 and 3 respectively for grey and colour images.

In our formalism, we used a pre-trained embedding $\boldsymbol{\psi} : \mathcal{X} \to \tilde{\mathcal{X}}$ from input images, $\boldsymbol{X}_i$, to high-level visual feature vectors, $\boldsymbol{\psi}(\boldsymbol{X}_i)$. Rather than learn a mapping to classes y_i, we instead learn a mapping to linguistic features describing the classes. This can

be viewed as an embedding, $\boldsymbol{\phi} : \mathcal{Y} \to \tilde{\mathcal{Y}}$, of the classes to a linguistic feature space, $\tilde{\mathcal{Y}}$. This is a distributed embedding in that a single class will have many features associated with it and each feature will be associated with many different classes. We are left with the relatively simple task of finding a mapping between visual feature vectors $\boldsymbol{\psi}(\boldsymbol{X}_i)$ and linguistic feature vectors $\boldsymbol{\phi}(y_i)$. To achieve this we use a neural architecture $\boldsymbol{\pi}(\boldsymbol{\psi}(\boldsymbol{X}_i), \boldsymbol{W})$, where $\boldsymbol{W}$ are trainable weights chosen by minimising the loss (energy) function

$$\sum_{(\boldsymbol{X}_i, y_i) \in \mathcal{S}} \ell(\boldsymbol{\phi}(y_i), \boldsymbol{\pi}(\boldsymbol{\psi}(\boldsymbol{X}_i), \boldsymbol{W})), \tag{3.1}$$

where $\ell : \tilde{\mathcal{Y}} \times \tilde{\mathcal{Y}} \to \mathbb{R}$ is a single-sample loss function defined in the linguist feature space. To make a class prediction, we independently train an inverse mapping $\phi^\dagger : \tilde{\mathcal{Y}} \to \mathcal{Y}$ from linguistic features to classes. Thus we can make class predictions using $\phi^\dagger(\boldsymbol{\pi}(\boldsymbol{\psi}(\boldsymbol{X}_i), \boldsymbol{W}))$.

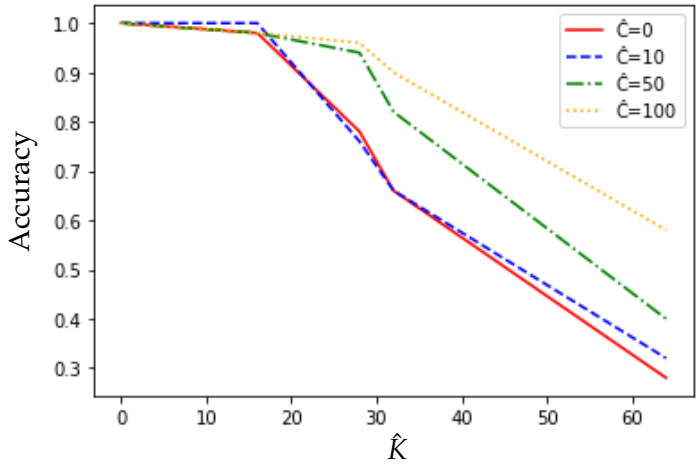


FIGURE 3.3: Classification accuracy of MLP_L using the generated training datasets $\mathcal{T}$ with different $\hat{C}$ and $\hat{K}$ values on the AwA2 dataset. Note that the set of C samples is perturbed $\hat{C} > 0$ times to obtain $C\hat{C}$ number of samples; and $\hat{K}$ represents the number of attributes whose values are manipulated per class against the ground-truth attribute-class matrix.

Feature extraction. Within our framework, there is a freedom to choose the feature extraction models $\boldsymbol{\psi}$ (see the left of Figure 3.2). To illustrate this flexibility, we have used both pre-trained ResNet101 [60] and VGG16 [58] for visual feature extraction. In both cases, we cut the network before the final dense layers.

3.3.1 Class Embedding

A central component in our approach is the introduction of a meaningful K-dimensional embedding space, $\tilde{\mathcal{Y}}$. We consider a mapping $\boldsymbol{\phi}$ from class $y_i \in \mathcal{Y}$ to an embedding vector $\tilde{\boldsymbol{y}}_i \in \tilde{\mathcal{Y}}$, where each component of $\tilde{\boldsymbol{y}}_i$ is a linguistic attribute. In practice, $\boldsymbol{\phi}$ would be a probabilistic embedding describing the conditional probability of $\mathbb{P}(\tilde{\boldsymbol{y}}_i \mid y_i)$; however, obtaining this is difficult. Instead, we start from a matrix $\mathbf{A} \in \mathbb{R}^{C \times K}$ provided

Algorithm 1 Multilevel XAI: training and test

TRAINING

Input: Image set $\mathcal{X}$, class label set $\mathcal{Y}$, pre-trained feature extraction network $\boldsymbol{\psi}$, and dataset $\mathcal{T}_0$.
Output: Models $\boldsymbol{\pi}$ and $\phi^\dagger$
$\tilde{\mathcal{X}} = \boldsymbol{\psi}(\mathcal{X})$ ▷ Image feature extraction
Generate dataset $\mathcal{T}$ by using Algorithm 2 on $\mathcal{T}_0$
Train $\phi^\dagger$ with $\mathcal{T}$ ▷ Map attributes to classes via MLP_L
Train $\boldsymbol{\pi}$ ▷ Match visual and linguistic attributes
return Models $\boldsymbol{\pi}$ (i.e., $\boldsymbol{\pi}_\text{MLP}/\boldsymbol{\pi}_\text{CNN}$) and $\phi^\dagger$

TEST

Input: Image $\boldsymbol{X}_j$, models $\boldsymbol{\psi}$, $\boldsymbol{\pi}$ and $\phi^\dagger$
Output: X-MLP/X-CNN ▷ Self-explainable predictions
$\hat{y}_j = \phi^\dagger(\boldsymbol{\pi}(\boldsymbol{\psi}(\boldsymbol{X}_j)))$ ▷ Predict the class label for $\boldsymbol{X}_j$
Calculate $\boldsymbol{u}_j$ using Eqn 3.2
X-MLP/X-CNN: Pick the top K^* largest components of $\boldsymbol{u}_j$ as the salient linguistic attributes
X-CNN: Extract the visual counterparts of the K^* attributes from the last CNN layer of $\boldsymbol{\pi}_\text{CNN}$
return X-MLP/X-CNN

by experts (in our case, this was conveniently provided by the zero-shot learning community, see Table 3.1 for an example [154]), which can be interpreted as $\mathbb{E}(\tilde{\boldsymbol{y}}_i \mid y_i)$. In our approach, we also need to learn the "inverse mapping", $\phi^\dagger(\tilde{\boldsymbol{y}}_i)$, giving $\mathbb{P}(y_i \mid \tilde{\boldsymbol{y}}_i)$. We learn this mapping using an MLP (MLP_L in our model, see the right of Figure 3.2), where our inputs are noisy vectors $\tilde{\boldsymbol{y}}_i$ (i.e., the rows of matrix $\mathbf{A}$) and our targets are the classes y_i. This mapping is learned entirely without seeing the training images and is then frozen. Although this is a rather simple approach, it is extremely fast to learn and leads to good performance. Given the Gaussian nature of the data with equal and isotropic variances, a distance-to-template classifier would also be a valid and potentially optimal choice. In this work, we opted for an MLP due to its flexibility and general applicability across tasks. In this regard, our choice reflects a modelling preference, and other options, such as distance-based classifiers, are expected to be equally valid in this setting.

3.3.2 Explainable Neural Networks

Below we introduce the strategies regarding how the DNN $\boldsymbol{\pi}$ in our proposed multilevel architecture (middle of Figure 3.2) can be explainable. Note that $\boldsymbol{\pi} : \boldsymbol{\psi}(\boldsymbol{X}_i) \rightarrow \boldsymbol{\phi}(y_i)$, where $\boldsymbol{X}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. Since $\boldsymbol{\phi}(y_i)$ is not unique, we train $\boldsymbol{\pi}$ to learn the

match between features and attributes in an unsupervised way (regarding $\tilde{\mathcal{Y}}$) using the training set $\mathcal{S}$ (i.e., image/label pairs), with the trained MLP_L and pre-trained $\boldsymbol{\psi}$.

$\forall \boldsymbol{X}_j \in \mathcal{X}$ used for testing, let $\hat{y}_j = \phi^\dagger(\boldsymbol{\pi}(\boldsymbol{\psi}(\boldsymbol{X}_j)), \boldsymbol{W})$ be the predicted class label of the test image $\boldsymbol{X}_j$. Let $\pi_k(\boldsymbol{\psi}(\boldsymbol{X}_j))$ be the k-th attribute of $\boldsymbol{\pi}(\boldsymbol{\psi}(\boldsymbol{X}_j))$, where $k \in \{1, 2, \ldots, K\}$. We define $\boldsymbol{u}_j \in \mathbb{R}^K$ as the importance of the attributes for the test image $\boldsymbol{X}_j$ and evaluate it by taking the gradient of the predicted class label $\hat{y}_j$ with respect to every attribute of $\boldsymbol{\pi}(\boldsymbol{\psi}(\boldsymbol{X}_j))$, i.e.,

$$\boldsymbol{u}_j = (u_j^1, u_j^2, \cdots, u_j^K) = \Big(\frac{\partial \hat{y}_j}{\partial \pi_1(\boldsymbol{\psi}(\boldsymbol{X}_j))}, \frac{\partial \hat{y}_j}{\partial \pi_2(\boldsymbol{\psi}(\boldsymbol{X}_j))}, \cdots, \frac{\partial \hat{y}_j}{\partial \pi_K(\boldsymbol{\psi}(\boldsymbol{X}_j))}\Big). \tag{3.2}$$

Then the top K^* largest of $\{u_j^k\}_{k=1}^K$ will be selected as the *salient linguistic attributes* for image $\boldsymbol{X}_k$. In this sense, $\boldsymbol{\pi}$ therefore can be explained by these salient linguistic terms. As examples, two of the most common neural networks – MLP and CNN – are adopted for $\boldsymbol{\pi}$ below.

Explainable MLP. Here $\boldsymbol{\pi}$ represents an MLP, say $\boldsymbol{\pi}_{\text{MLP}}$ consisting of a few dense layers. $\boldsymbol{\pi}_{\text{MLP}}$ can then be explained by the obtained salient linguistic terms in a single-level manner. We also call $\boldsymbol{\pi}_{\text{MLP}}$ explainable MLP (X-MLP).

Explainable CNN. Here $\boldsymbol{\pi}$ represents a CNN, say $\boldsymbol{\pi}_{\text{CNN}}$ consisting of convolutional layers with K channels followed by a GAP layer. $\boldsymbol{\pi}_{\text{CNN}}$ can then be explained by the obtained salient linguistic terms. Moreover, we can also find out where these salient attributes are related in the given test image $\boldsymbol{X}_j$ by exploiting the spatial information preservation property of the CNN structure. For the i-th attribute, $i \in \{1, 2, \ldots, K\}$, its attribute-wise saliency map (i.e., heat map mask) say $\bar{\boldsymbol{X}}_j^i$ can be obtained by the output of the last CNN layer after being upsampled to the same size of $\boldsymbol{X}_j$; in other words, the salient part of $\boldsymbol{X}_j$ is corresponding to the i-th salient attribute $\bar{\boldsymbol{X}}_j^i$.

In contrast to $\boldsymbol{\pi}_{\text{MLP}}$, $\boldsymbol{\pi}_{\text{CNN}}$ provides both the salient linguistic terms and the corresponding attribute-wise saliency maps in a multilevel manner with no extra cost. For ease of reference, we call $\boldsymbol{\pi}_{\text{CNN}}$ explainable CNN (X-CNN). The procedures regarding training and testing of our approach are summarised in Algorithm 1.

3.4 Data

Animals with Attributes (AwA1) is a well-known dataset used in zero-shot learning [154]. Due to the absence of raw images and copyright issues, an alternative version of it named AwA2 was introduced in [149]. It is a medium-scale coarse-grained dataset with 37 322 images from 50 classes collected from public web sources, including 85 attributes per class available. Table 3.1 (see Table 3.3 for its full version) presents a size of 5×7 excerpt of AwA2, exemplifying the nature between the attributes and different

classes. The other benchmark dataset used in this work is CUB-200-2011 [55]. CUB is a fine-grained dataset containing around 11 800 images of 200 different bird classes, including 312 attributes per class. These linguistic attributes will be exploited to create self-explainable DNNs under our proposed methodology.

TABLE 3.1: An excerpt of the attribute-class matrix **A** for the AwA2 dataset. Attribute values are in [0, 100] and standardised before use.

Classes \ Attributes	*Gray*	*Patches*	*spots*	*Lean*	*Tail*	*Strong*	*Muscle*	···
Antelope	12.34	16.11	9.19	39.99	40.59	33.56	26.14	···
Grizzly bear	3.75	1.25	0	0	9.38	78.48	48.89	···
Killer whale	1.25	68.49	32.69	22.68	41.67	63.35	10.45	···
Beaver	7.5	0	7.5	8.75	86.56	32.81	24.38	···
Dalmatian	0	37.08	100	63.68	53.75	34.93	23.75	···

3.4.1 Generation of the Training Dataset $\mathcal{T}$ for $\mathrm{MLP_L}$

Recall that we have the attribute-class matrix $\mathbf{A} \in \mathbb{R}^{C\times K}$, where C and K represent the number of classes and the number of attributes per class, respectively. Table 3.3 shows the full matrix $\mathbf{A}$ for the AwA2 dataset. The original matrix $\mathbf{A}$ can directly form a dataset, i.e.,

$$\mathcal{T}_0 = \left\{(\tilde{\boldsymbol{y}}_k, y_k) \mid \tilde{\boldsymbol{y}}_k \in \tilde{\mathcal{Y}}, \dagger_{\|} \in \mathcal{Y}, \| = \infty, \in, \ldots, \mathcal{C}\right\}, \tag{3.3}$$

but this is too small to train $\mathrm{MLP_L}$; note that $\tilde{\boldsymbol{y}}_k = (\tilde{y}_k^1, \cdots, \tilde{y}_k^K)$ is the k-th row of $\mathbf{A}$ for class k, and $\tilde{y}_k^i$ is the i-th attribute of $\tilde{\boldsymbol{y}}_k$.

We augment $\mathcal{T}_0$ by upsampling each sample $(\tilde{\boldsymbol{y}}_k, y_k) \in \mathcal{T}_0$ to $\hat{C}$ number of samples by randomly manipulating $\hat{K}$ number of attributes of $\tilde{\boldsymbol{y}}_k$ among the total K, with the aim of perturbing the original samples. The values of the selected attributes for manipulation can be conducted randomly. In our setting, we use two values $\beta_0 > 0$ and $\beta_1 < 0$, and change the positive values of the selected attributes to β_1, otherwise, to β_0. Note that this setting is an arbitrary choice (here we use $\beta_0 = 1.5$ and $\beta_1 = -0.5$), which can be replaced by other ways appropriate. We finally generate a training dataset $\mathcal{T}$ with $C\hat{C}$ ($\hat{C} > 0$) number of samples. In our experiments, $\hat{C}$ is set to 100. The data generation process is summarised in Algorithm 2.

3.5 Experiments

All the experiments were implemented on a personal laptop with the following specifications: i) i7-8750H CPU; ii) GeForce GTX 1060 GPU; and iii) 16GB RAM. The proposed methodology is trained and tested on the coarse-grained and fine-grained benchmark datasets. Training of $\mathrm{MLP_L}$ takes around 15 minutes. Training of X-MLP and X-CNN

Algorithm 2 Generation of the training dataset $\mathcal{T}$ for MLP_L

Input: Dataset $\mathcal{T}_0$, empty dataset $\mathcal{T}$, upsampling rate $\hat{C} \in \mathbb{N}$, the number of attributes manipulated $\hat{K}$, $\beta_0 > 0$ and $\beta_1 < 0$.
Output: Training dataset $\mathcal{T}$ ▷ Generated training dataset to train MLP_L
Get the values of C and K from the dataset $\mathcal{T}_0$
for $i = 1$ to $\hat{C}$ **do**
  for $k = 1$ to C **do**
    Get the current sample $(\tilde{\boldsymbol{y}}_k, y_k) \in \mathcal{T}_0$
    Set $\tilde{\boldsymbol{y}}_{k,i} = \tilde{\boldsymbol{y}}_k$
    for $j = 1$ to $\hat{K}$ **do**
      Generate a random number $t \in \{1, \cdots, K\}$ for which attribute value to change
      if $\tilde{y}^t_{k,i} \leq 0$ **then**
        $\tilde{y}^t_{k,i} = \beta_0$
      else
        $\tilde{y}^t_{k,i} = \beta_1$
      end if
    end for
    Add $(\tilde{\boldsymbol{y}}_{k,i}, y_k)$ into $\mathcal{T}$
  end for
end for
return Dataset $\mathcal{T}$

takes around 30 minutes and 80 minutes, respectively. The pre-trained feature extraction models (i.e., $\boldsymbol{\psi}$) ResNet101 and VGG16 are downloaded from Keras' website[1]. The implementation setup and results in the XAI regime are given below.

3.5.1 Implementation Setup

I) For the feature extraction model $\boldsymbol{\psi}$, pre-trained ResNet101 and VGG16 are respectively used for datasets AwA2 and CUB. The sizes of the extracted features for each image in datasets AwA2 and CUB are respectively $8 \times 8 \times 2048$ and $8 \times 8 \times 512$. **II)** The language model MLP_L is a few layers wide MLP (here 3 layers are used). To train it, two training sets, $\mathcal{T}$, with size of 5 000 and 20 000 respectively for datasets AwA2 and CUB are formed.

MLP_L, including the order of the attributes, is frozen after the training completes. **III)** $\boldsymbol{\pi}_{\text{MLP}}$ is a few layers wide MLP (here 4 layers are used) taking 2 048 and 512 features extracted by $\boldsymbol{\psi}$ and outputting 85 and 312 attributes for datasets AwA2 and CUB, respectively. $\boldsymbol{\pi}_{\text{CNN}}$ for simplicity is set to one single convolutional layer with the size of $8 \times 8 \times 85$ and $8 \times 8 \times 312$ for datasets AwA2 and CUB, respectively. A 30/70 split of the data was formed for training/test. **IV)** For comparison, *fine-tuned* ResNet101 and

[1] Pre-trained ResNet101 and VGG16: https://keras.io/api/applications/

VGG16 are obtained by directly using the training set of image/label pairs of AwA2 and CUB, respectively.

The Adam optimizer with a learning rate of 0.001 and batch size of 32 is used in all experiments. The number of epochs is set to 100 and early stopping is applied (with patience set to 10 based on the validation loss). We stress that our main goal is to make DNNs self-explainable rather than accuracy-driven. It is expected that the prediction performance reported could be improved for example with hyper-parameter fine-tuning and/or wiser selection of the feature extraction model $\boldsymbol{\psi}$.

3.5.2 Classification Performance of $\mathrm{MLP_L}$

Although our main focus in this work is XAI rather than the classification accuracy of $\mathrm{MLP_L}$, it is worth evaluating the classification accuracy performance of $\mathrm{MLP_L}$ under the training dataset $\mathcal{T}$. To do so and also investigate the impact of the parameters $\hat{C}$ and $\hat{K}$, we first generate different training dataset $\mathcal{T}$ using different upsampling rate $\hat{C}$ and different value of $\hat{K}$ (i.e., the number of attributes whose values are manipulated) ranging from 0 to 65 (i.e., up to two-thirds of the total 85 attributes per class in the AwA2 dataset). Afterwards, to evaluate the accuracy performance of $\mathrm{MLP_L}$ using the generated datasets $\mathcal{T}$ regarding different $\hat{C}$ and $\hat{K}$, we further split each dataset $\mathcal{T}$ into two parts with the ratio of 70/30 for training and test, respectively.

Figure 3.3 shows the classification accuracy of $\mathrm{MLP_L}$ corresponding to different $\hat{K}$ and $\hat{C}$ values. It is seen that the accuracy decreases when $\hat{K}$ becomes larger for each $\hat{C}$, which is reasonable, since the larger the $\hat{K}$, the higher the perturbation of the ground truth attribute-class samples. The results also show that larger $\hat{C}$ (which leads to a bigger dataset $\mathcal{T}$) results in more robust models against more noisy samples, e.g. see the yellow line in Figure 3.3 for the case of $\hat{C} = 100$. In contrast, when $\hat{C}$ is small, there is a significant classification performance drop as shown in Figure 3.3; e.g., the classification accuracy drops from over 90% (for $\hat{C} = 100$) to 65% (for $\hat{C} = 10$) when $\hat{K} = 32$.

Finally, for completeness, we investigate the case of no perturbation, i.e., the case of $\hat{C} = 0$. In this case, Table 3.3, $\mathcal{T}_0$, is directly used for the $\mathrm{MLP_L}$ training. We generate test sets following the same way of generating $\mathcal{T}$ by manipulating the samples in $\mathcal{T}_0$ once with different $\hat{K}$ values. As shown in Figure 3.3 for $\hat{C} = 0$, the model's performance drops significantly when $\hat{K}$ becomes larger, indicating the limited performance of the model trained just by using the original set $\mathcal{T}_0$. In our XAI experiments, we pick $\hat{C} = 100$ and $\hat{K} = 8$ and remark that there is a freedom of choice for these hyperparameters to obtain better results.

3.5.3 Classification Performance of the Proposed Architecture

Table 3.2 shows that the proposed multilevel XAI architecture in Figure 3.2 can achieve surprisingly good performance (i.e., over 90% and $\sim$ 50% accuracy for the 50-class and 200-class datasets AwA2 and CUB, respectively) in classification accuracy against the fine-tuned neural networks (i.e., ResNet101 and VGG16 trained directly on the labelled data, which lack explainability) on hold-out test set even though this is not the main aim of this work. The neural networks' performance in accuracy highly depends on the quality of the data acquired. However, most of the data researchers work on, if not all, could be biased, insufficient and/or sensitive. Creating architectures that can explain themselves and simultaneously reach high prediction performance – just like the one introduced in this work – is arguably the long-term pursuit in ML.

TABLE 3.2: Classification accuracy. X-MLP/X-CNN can achieve comparable performance against the fine-tuned ResNet101 and VGG16, which, however, lack the linguistic and visual explainability that X-MLP/X-CNN delivers.

Data	Model	Test Accuracy	Explainability
AwA2	ResNet101	95.8 ± 1.3	N/A
	X-MLP	90.5 ± 0.8	Unilevel
	X-CNN	90.1 ± 1.1	Multilevel
CUB	VGG16	57.2 ± 1.4	N/A
	X-MLP	54.9 ± 1.5	Unilevel
	X-CNN	44.6 ± 1.1	Multilevel

3.5.4 Explainability Performance of Our Multilevel XAI Method

3.5.4.1 Explainability for correct predictions

For a given image Least Auklet from the CUB dataset, see Figure 3.1, both the fine-tuned DNN (VGG16 in our case) and our proposed method can easily classify it correctly. However, the fine-tuned DNN gives no explanation for why it reaches a decision by itself. Post-hoc XAI methods (such as [39, 42, 43]) could be employed to see whether the classified object as a whole in the given image is the main part that the fine-tuned DNN focuses on (i.e., left of Figure 3.1), but this level of explanation is rather limited and is an incomplete reflection of human-like explanations as discussed throughout the chapter. In contrast, the attribute-wise level of explanation that the proposed multilevel XAI model delivers (i.e., right of Figure 3.1) is much richer, wider, deeper and self-explainable thanks to the linguistic attributes. In detail, some of the most salient attributes that affect the prediction are presented as striped belly, rufous bill and grey breast. Their corresponding saliency maps convincingly highlight the correct part of the image for the mentioned individual attributes. This type of explanation is desirable and is an important indicator of the match between image features and class-wise

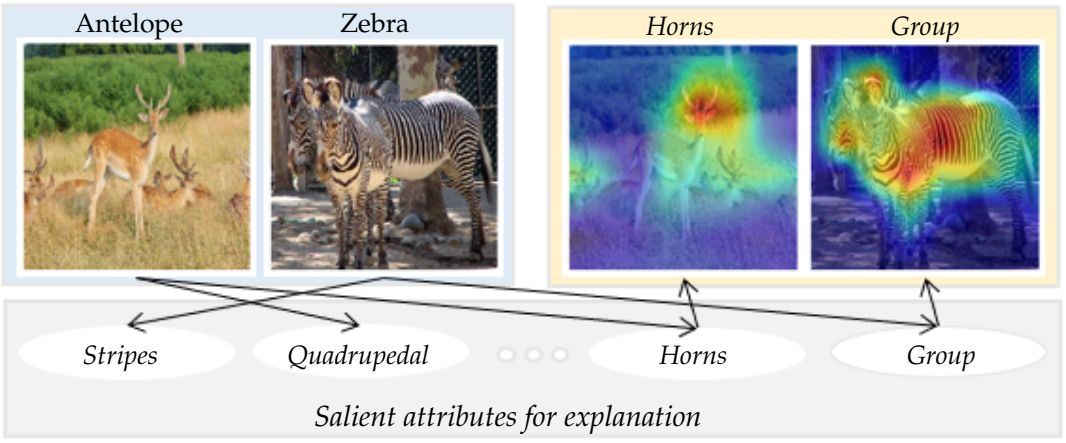


FIGURE 3.4: Explainability of the proposed approach for correct prediction. Human-like multilevel explanations are delivered by the salient attributes and their saliency maps (by X-CNN), which are matched well.

attributes that are learnt in an unsupervised way by the proposed architecture (i.e., unsupervised in the sense that the training images have not been labelled by linguistic attributes and/or salient regions have not been given).

Figure 3.4 demonstrates the power of the proposed model in explainability with more challenging images. Linguistic self-explainable attributes of stripes and group are outputted as salient for zebra, while quadrupedal and horns are outputted for antelope by X-MLP and X-CNN. Attribute-wise saliency maps for horn and group outputted by X-CNN show the human-like explanation power of our approach. Some attributes can be well captured by DNNs with examples shown in Figures 3.1, 3.4 and 3.18. To further demonstrate this property, we present below more images from a variety of classes, showing that the presented attributes are indeed learnt rather than special to the given images or classes.

The attribute-wise saliency maps for example in Figure 3.5 show that the attribute *horns* is clearly learnt for the Ox, Antelope and Buffalo classes; for more results see Figures 3.6 and 3.7. Further examples of the correct class prediction with the multilevel explainability by our approach are shown in Figures 3.8 and 3.9. For abstract attributes, their saliency maps could give us an idea of what part of the image activates that specific attribute, e.g. active or weak. Moreover, the attribute values given by experts in Table 3.1 for the predicted classes indicate whether experts think these attributes are helpful in discriminating one class from the others. After checking, we can see these salient attributes obtained by our approach for the predicted classes are indeed meaningful.

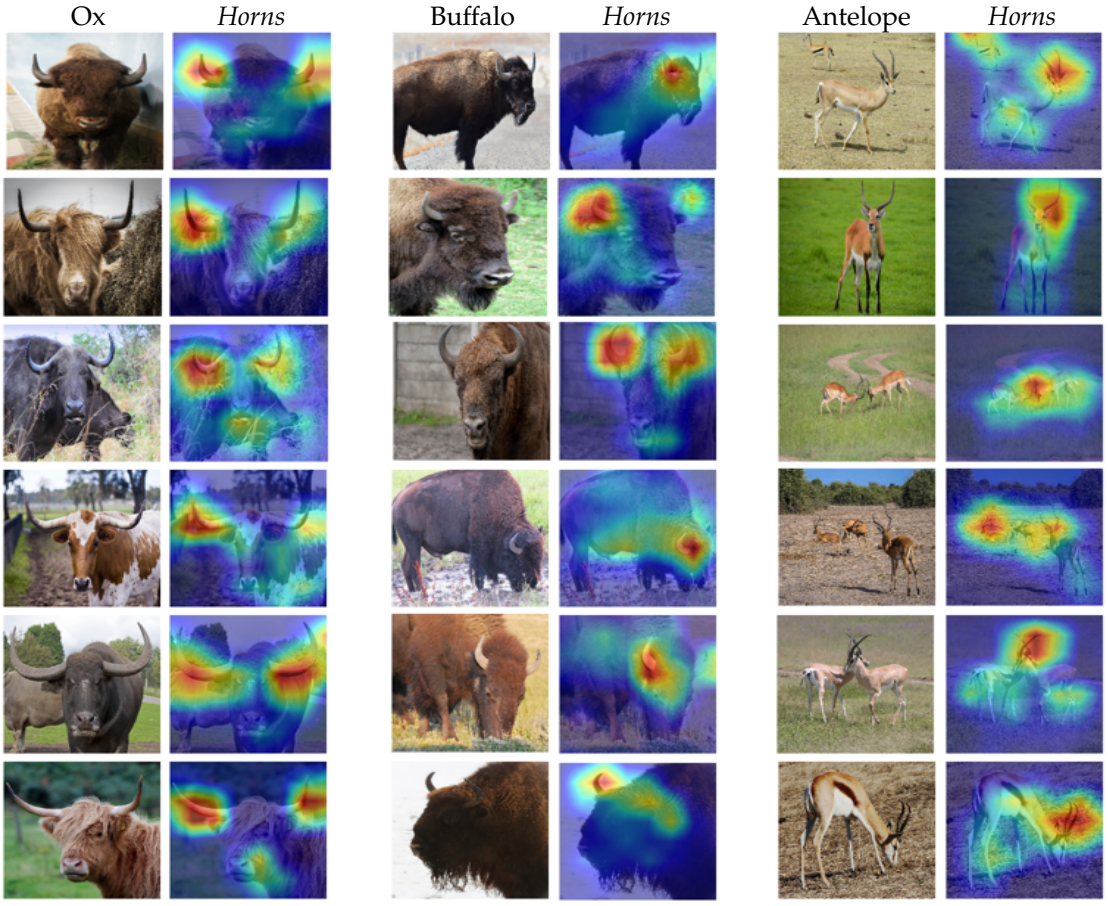


FIGURE 3.5: Images from different classes in the AwA2 dataset with their obtained attribute-wise saliency maps by our approach. Examples from Ox, Buffalo and Antelope classes show that the attribute *horns* is well captured by X-CNN.

3.5.4.2 Explainabilility regarding attribute-class prediction

Although the class-wise attributes are learnt in an unsupervised way by our approach, the previous experiments have shown the power of our model in activating meaningful attributes. To further demonstrate its ability in attribute prediction, we present the attribute-class prediction averaged over the test samples for each class, see Table 3.4. The predicted attribute values in this table are expected to be close to Table 3.3 (i.e., the ground truth). Moreover, Figure 3.10 shows the top five classes that maximally activate the given individual attributes (i.e., white, pads, etc.), together with one representative image from each class. It shows that the maximally activated attributes are indeed meaningful and highly relevant to the classes that activate them. More results are given in Figures 3.11, 3.12, 3.13 and 3.14 for colour, skin-type, movement and body-part related attributes, respectively.

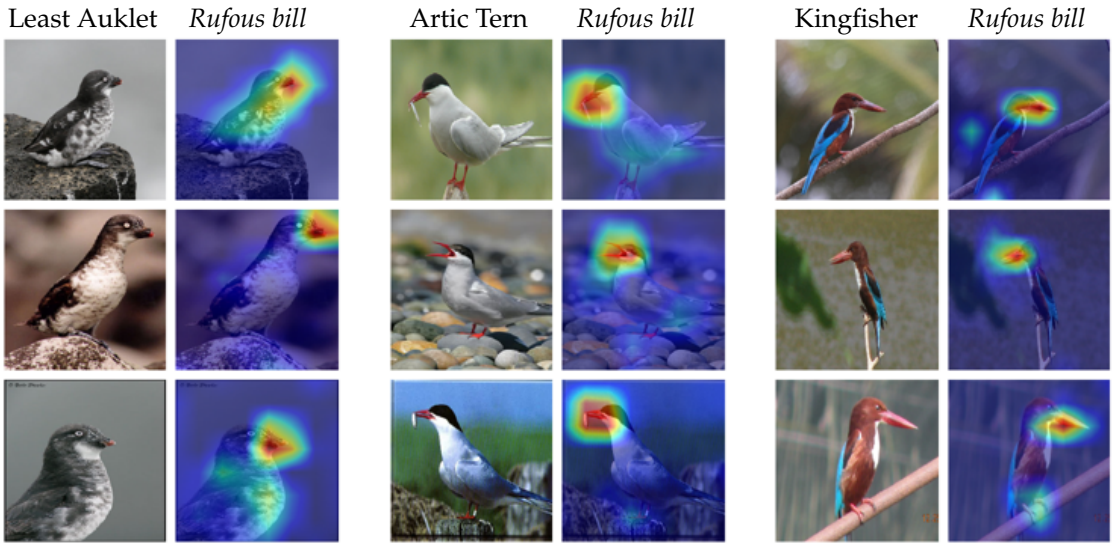


FIGURE 3.6: Images from different classes in the CUB dataset with their obtained attribute-wise saliency maps by our approach. Examples from Least Auklet, Artic Tern and Kingfisher classes show that the attribute *rufous bill* is well captured by X-CNN.

FIGURE 3.7: Images from different classes in the CUB dataset with their obtained attribute-wise saliency maps by our approach. Examples from Kittiwake and Artic Tern classes show that the attribute *rufous leg* is well captured by X-CNN.

3.5.4.3 Explainabilility for incorrect prediction

Reaching 100% prediction accuracy is not the case for any method given a nontrivial task. Therefore, investigating the reason behind wrong predictions is equally important. The left of Figure 3.17 shows a grizzly bear which is misclassified as the polar bear and, to understand the reason behind this, the top five salient attributes obtained by the proposed approach for both grizzly bear and polar bear classes are presented. These attributes are indeed the ones that differentiate these two classes (*cf.* the full

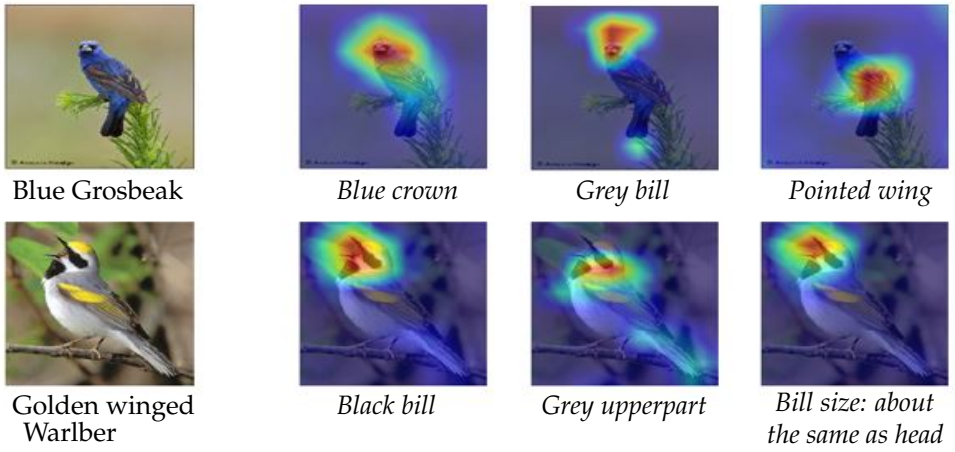


FIGURE 3.8: Explainability of the proposed approach for correct class prediction. Left: randomly selected test images in the CUB dataset. Right: the top three most salient attributes helping the neural network make the correct classification and the corresponding attribute-wise saliency maps.

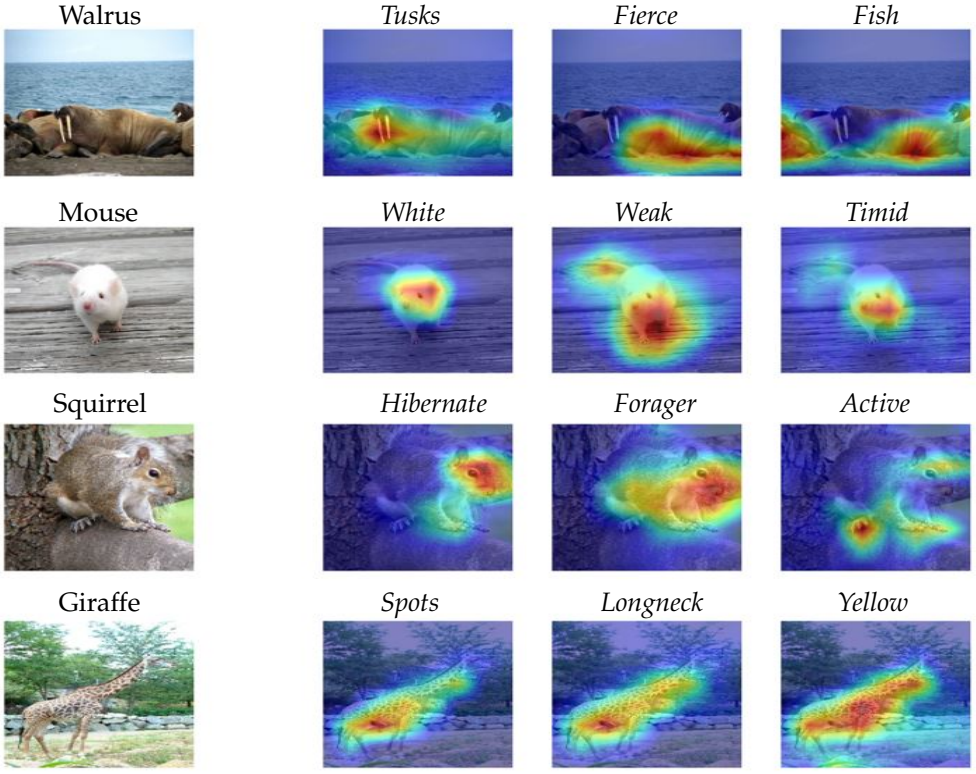


FIGURE 3.9: Explainability of the proposed approach for correct class prediction. Left: randomly selected test images in the AwA2 dataset. Right: the top three most salient attributes helping the neural network make the correct classification and the corresponding attribute-wise saliency maps.

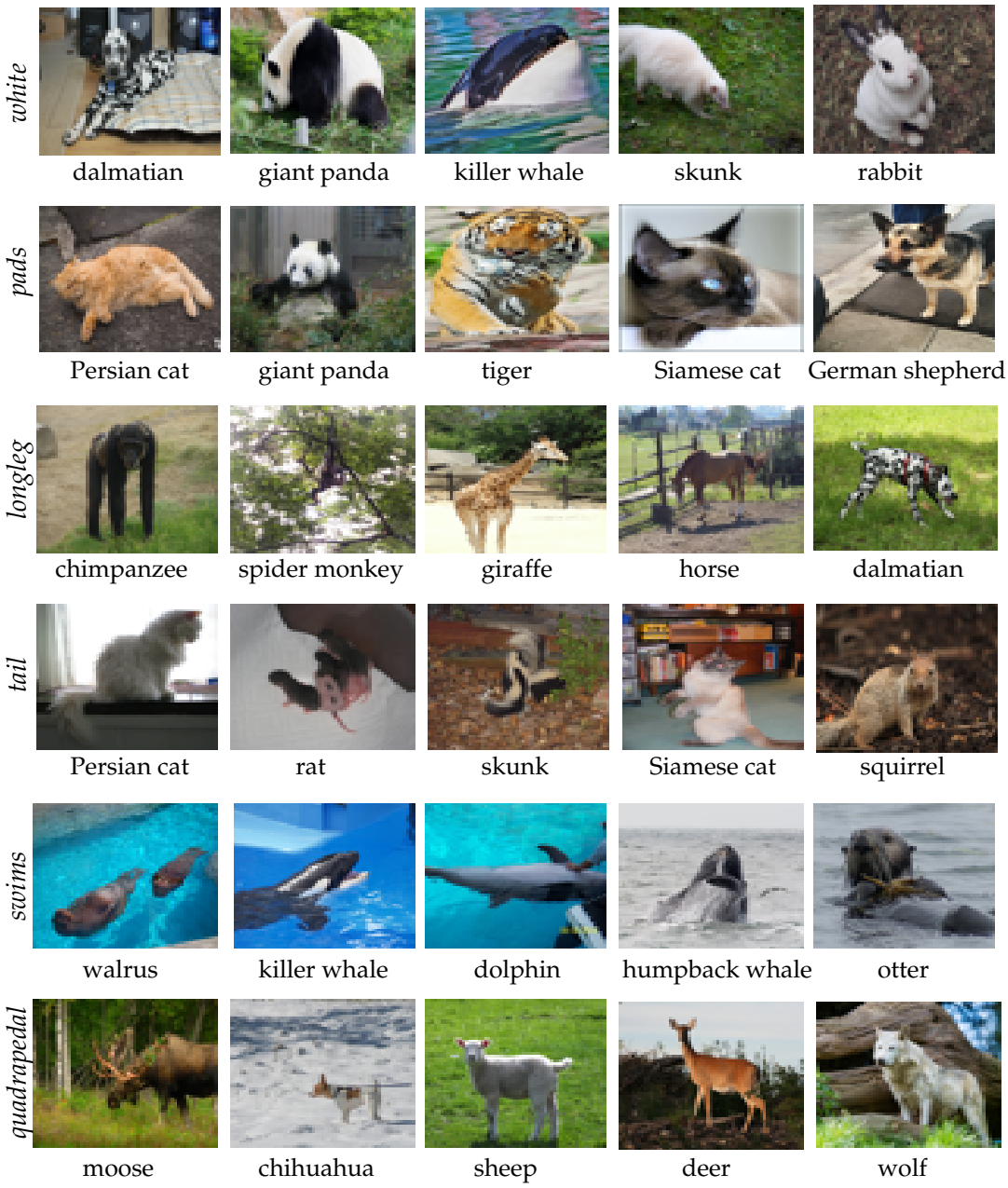


FIGURE 3.10: The top five classes that maximally activate the given individual attributes on the left. For better illustration, one representative image from each class is also shown together with the class name.

attribute-class matrix in Table 3.3). The attribute-wise saliency maps of the most salient attributes (i.e., white and brown) for both classes provide further insights regarding why this misclassification occurred. After occluding the part considered as "white" (by our approach) with a brown patch, the manipulated image is then correctly classified by our approach as a grizzly bear with high confidence, see the right of Figure 3.17; furthermore, the saliency map for the most salient attribute, "brown", now indeed highlights both the head of the bear and the brown patch, showing that our approach clearly learns what brown is and considers it as a strong indicator of grizzly bear class.

FIGURE 3.11: The top five classes that maximally activate the given individual colour-related attributes on the left. For better illustration, one representative image from each class is also shown together with the class name.

Further examples of incorrect class prediction with the multilevel explainability by our approach are shown in Figures 3.15 and 3.16. The grizzly bear classified as polar bear and the whale classified as dolphin are some of the most frequent misclassification cases detected. After checking the attribute values in Table 3.1 given by the experts for the predicted classes and the ground-truth classes, we can see these salient attributes obtained by our approach are indeed consistent with the ones given by experts for the predicted classes; see also more discussion in Section 3.6 for the challenges e.g. the linguistic alignment and the nature of explainability.

FIGURE 3.12: The top five classes that maximally activate the given individual skin type attributes on the left. For better illustration, one representative image from each class is also shown together with the class name.

3.5.4.4 Sensitivity between attributes and features

To further investigate the effectiveness of the linguistic attributes in our method in explainability, we test a zebra image and its artificial conversion to a horse using CycleGan [155], i.e., the attribute *stripes* is removed from the zebra, see Figure 3.18. Again, all three models (i.e., fine-tuned ResNet101, X-MLP and X-CNN) classified the zebra image as zebra and the artificially generated horse as a horse. At this point the fine-tuned ResNet101 has no explanation ability to show what changed in the original image that forces it to output "horse". In contrast, our model clearly shows that "stripes" is one

FIGURE 3.13: The top five classes that maximally activate the given individual movement-related attributes on the left. For better illustration, one representative image from each class is also shown together with the class name. In particular, for the attribute 'flys' (see the last row in the figure), only one class that maximally activates it is shown since there is only one aminal (i.e., bat) that can fly in the dataset.

of the salient attributes for the original zebra image with the attribute value of 2.15 and it drops to 0.41 for the artificially generated horse image. To validate the reason behind this visually, the attribute-wise saliency maps, generated by our approach with *no extra cost*, indicate that the X-CNN model focuses on the body of the zebra where the "stripes" lie, whereas arbitrary parts of the artificially generated horse image are highlighted when asked to show where the stripes are, see the bottom of Figure 3.18.

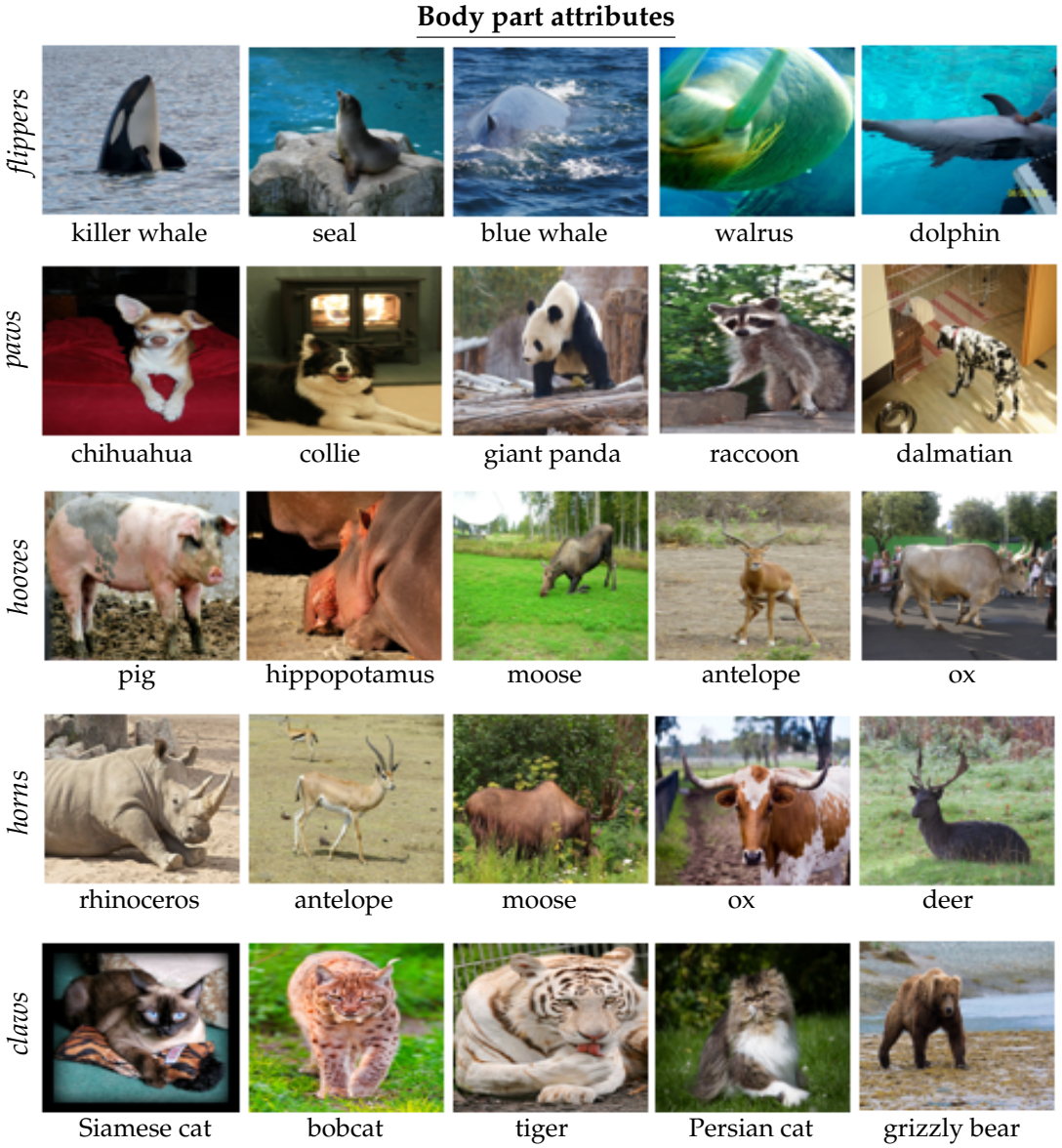


FIGURE 3.14: The top five classes that maximally activate the given individual body part attributes on the left. For better illustration, one representative image from each class is also shown together with the class name.

3.5.4.5 Class embeddings with shuffled attribute values

Previous experiments are conducted using the prior information (i.e., the attribute-class matrix shown in Table 3.1) provided by experts. An interesting and natural question is: what will the results be if the attribute-class matrix takes different values? In other words, to what extent, will the prior information in the attribute-class matrix be helpful in interpretability? To investigate this, we first shuffled all the columns of the attribute-class matrices for both datasets. In an extreme case, say the values of "ground" and "water" for the tiger class may be switched, which apparently would cause a dramatic information loss against the one provided by experts. The shuffled datasets are then used to train the model MLP_L. Surprisingly, we found that it converged as fast as using

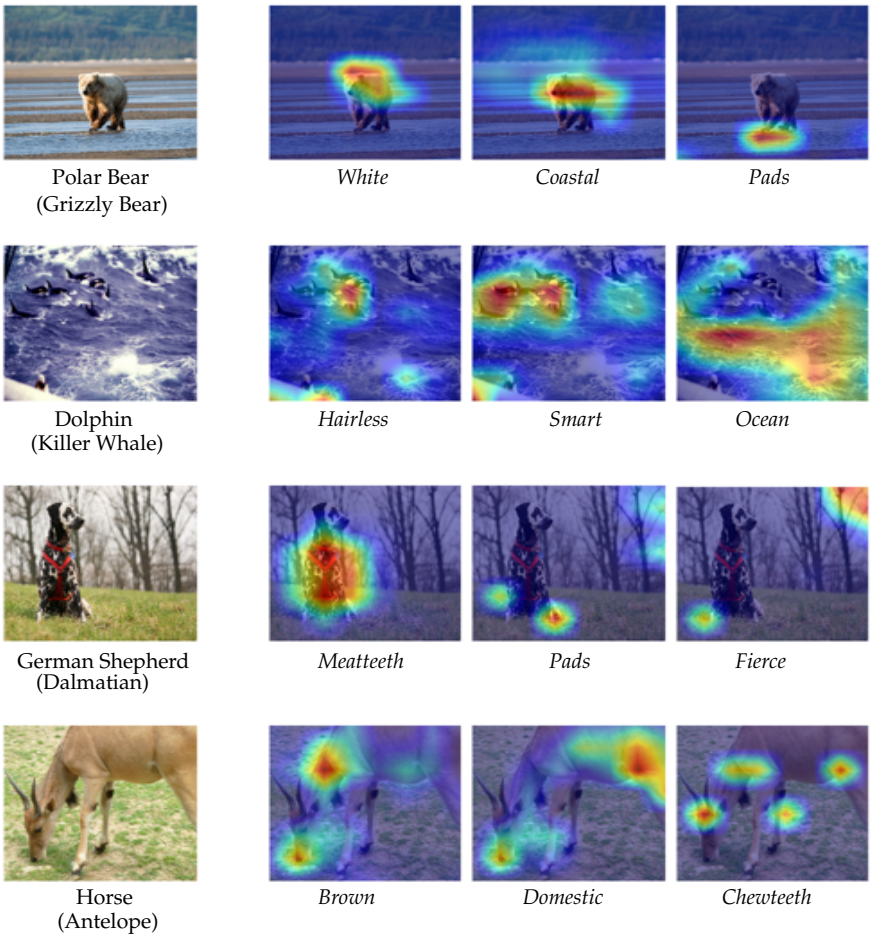


FIGURE 3.15: Explainability of the proposed approach for incorrect class prediction. Left: randomly selected test images in dataset AwA2; e.g., Polar Bear (Grizzly Bear) means the Grizzly Bear class is incorrectly predicted to be Polar Bear. Right: the top three most salient attributes helping the neural network make the incorrect classification and the corresponding attribute-wise saliency maps.

the original data; moreover, the newly trained models X-MLP and X-CNN also reached an accuracy close to the ones obtained by using the original data.

Figure 3.19 shows the interpretability results provided by our approach in this attributes shuffling scenario. Obviously, the linguistic attributes become meaningless; moreover, we also observe that the salient regions no longer appear to be associated with the object being recognised and defy an easy human explanation in contrast to what was observed in Figure 3.1.

This finding by our approach shows that those pre-determined attribute lists are crucial to explainability. It also suggests that purely relying on the accuracy of DNNs (which might be trained on data with unknown flaws) could be perilous and the corresponding interpretability is essential. Further discussion is in Section 3.6.

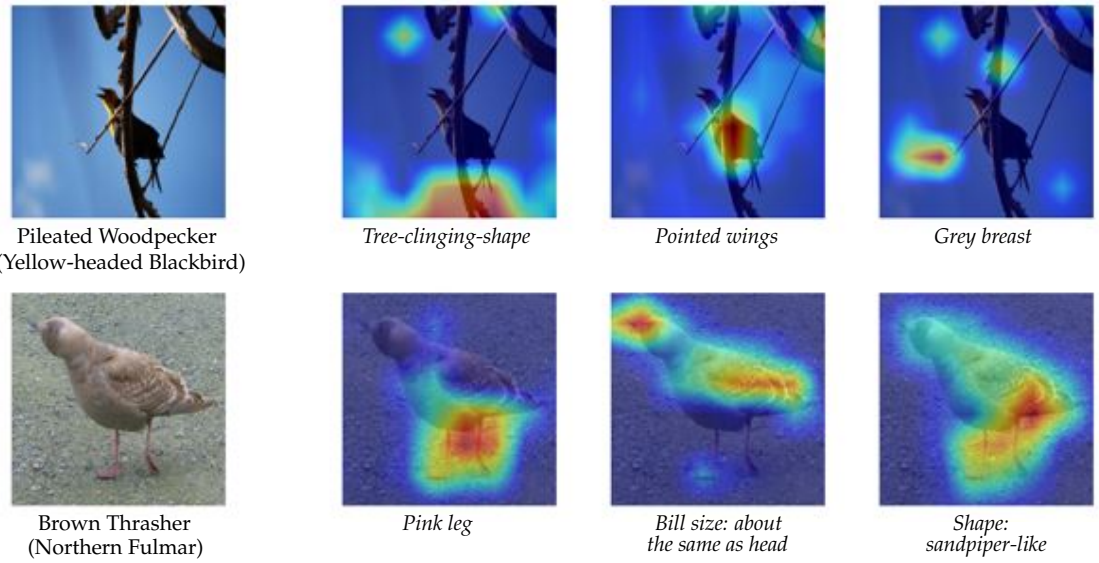


FIGURE 3.16: Explainability of the proposed approach for incorrect class prediction. Left: randomly selected test images in dataset CUB. Right: the top three most salient attributes helping the neural network make the incorrect classification and the corresponding attribute-wise saliency maps.

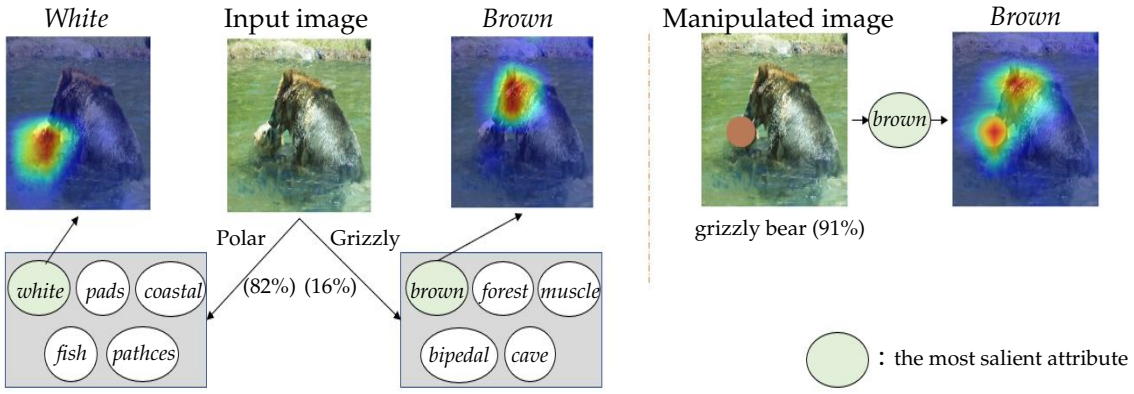


FIGURE 3.17: Explainability of the proposed approach for incorrect prediction. Left: A grizzly bear which is misclassified as a polar bear. The top five salient attributes are shown in ellipses for the highest probable class (i.e. polar bear) and second class (i.e. grizzly bear). The most salient attributes (i.e., white and brown) and their saliency maps provide insights into the prediction. Right: A manipulated grizzly bear image (obtained by replacing the area related to the attribute "white" with a brown patch) which is then correctly classified by our approach with high confidence (91%).

3.5.4.6 Information of linguistic attributes

We now study the information of linguistic attributes, which will give us some guidance about the relationship between each class and its attributes.

For a set of classes $\mathcal{Y}$ (in our case the 50 species in the dataset AwA2), let Y be a random variable describing our classes, and define the probability of a class having class label

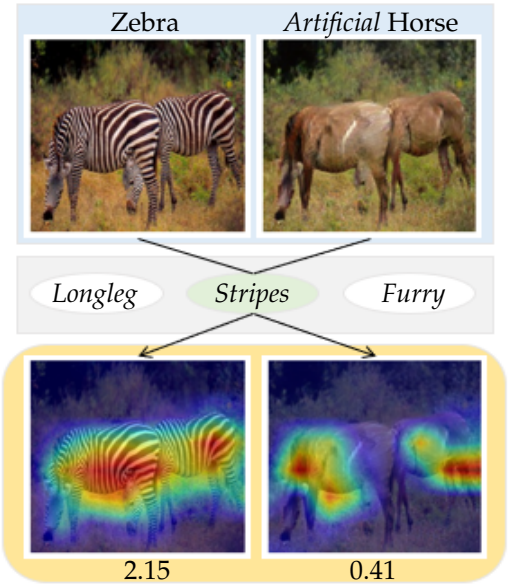


FIGURE 3.18: Effectiveness of linguistic attributes in our approach. "Stripes" is one of the salient attributes for zebra with the value of 2.15 and its saliency map reasonably highlights the body of the zebra. For the artificial horse image, the attribute "stripes" is of value 0.41 and its saliency map is meaningfully unrelated.

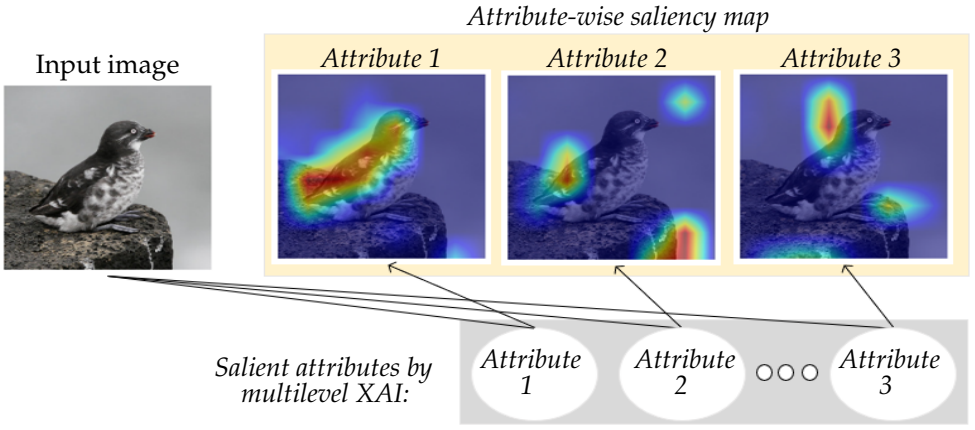


FIGURE 3.19: Explainability of the proposed approach in the scenario of attributes shuffling (*cf.* Figure 3.1). Model accuracy reaches the same level as when we use the true linguistic attribute values, whereas the attribute-wise saliency maps are meaningless to humans, illustrating the importance of prior knowledge of the attribute values.

$y \in \mathcal{Y}$ as $\mathbb{P}[Y = y]$. Then the entropy (or uncertainty) in the class label is

$$H_Y = -\sum_{y \in \mathcal{Y}} \mathbb{P}[Y = y] \log(\mathbb{P}[Y = y]). \tag{3.4}$$

To compute the mutual information for a linguistic attribute, $\tilde{y}^k$, we need to know $\mathbb{P}\left[Y \mid \tilde{y}^k\right]$, which can be estimated from a proper image set $\mathcal{V} \subseteq \mathcal{X}$. If we choose an image $\boldsymbol{X}_i \in \mathcal{V}$, we can feed it into our network (i.e., see Figure 3.20) to find its attribute value, $\tilde{y}_i^k, 1 \leq k \leq K$. To simplify the calculation, let us set a threshold on the attribute

value so that if it is above the threshold we treat $\tilde{y}_i^k = 1$, otherwise $\tilde{y}_i^k = 0$. For each $y \in \mathcal{Y}$ and each $\alpha \in \{0, 1\}$, we can estimate various probabilities, e.g.,

$$\mathbb{P}\left[Y = y, \tilde{y}^k = \alpha\right] \approx \frac{\sum_{\boldsymbol{X}_i \in \mathcal{V}} \left[\!\left[\boldsymbol{X}_i \text{ in class } y\right]\!\right] \left[\!\left[\tilde{y}_i^k = \alpha\right]\!\right]}{|\mathcal{V}|},$$
$$\mathbb{P}\left[\tilde{y}^k = \alpha\right] \approx \frac{\sum_{\boldsymbol{X}_i \in \mathcal{V}} \left[\!\left[\tilde{y}_i^k = \alpha\right]\!\right]}{|\mathcal{V}|}, \tag{3.5}$$

where $\left[\!\left[\text{predicate}\right]\!\right]$ is an indicator function (equal to 1 if the predicate is true and 0 otherwise). We can then compute

$$\mathbb{P}\left[Y = y \mid \tilde{y}^k = \alpha\right] = \frac{\mathbb{P}\left[Y = y, \tilde{y}^k = \alpha\right]}{\mathbb{P}\left[\tilde{y}^k = \alpha\right]}. \tag{3.6}$$

The conditional entropy of the classes given the (binarised) linguistic attribute is given by

$$H_{Y|\tilde{y}^k} = -\sum_{\tilde{y}^k \in \{0,1\}} \sum_{y \in \mathcal{Y}} \mathbb{P}\left[Y = y, \tilde{y}^k = \alpha\right] \log\left(\mathbb{P}\left[Y = y \mid \tilde{y}^k = \alpha\right]\right). \tag{3.7}$$

The mutual information on the classes due to the linguistic attributes is given by

$$I\left(Y; \tilde{y}^k\right) = H_Y - H_{Y|\tilde{y}^k}. \tag{3.8}$$

Calculating the mutual information by attributes $\tilde{y}^k$ predicted by our approach and Eqn (3.8) gives us a measure of how important each attribute is in differentiating the classes, see Table 3.5. Some of them seem useless by themselves as seen in Table 3.5; however, they could be important in combination with others.

An alternative way to calculate the mutual information is by following the same steps above but obtaining $\tilde{y}^k$ values from the attribute-class matrix given in Table 3.1 instead of dataset $\mathcal{V}$. These alternative mutual information values represent the importance of each attribute for human experts who created Table 3.1. The calculated alternative mutual information is also presented in Table 3.5.

We now have two mutual information values per attribute, i.e., one is by using the dataset $\mathcal{V}$ (here we use the test set) and the other is by using the prior knowledge given in Table 3.1, see Table 3.5. Comparison between these two values for each attribute is helpful to see the discrepancy between what attributes people think are important and the ones that our trained DNNs learn. To give an example, "small" reduces the uncertainty by 0.98 bits when using the table values given by experts. However, it takes the value of 0.13 when using images, which suggests that "small" is not one of the best attributes for the trained DNN to differentiate the given classes, although people think it is. Further discussion is in Section 3.7 below.

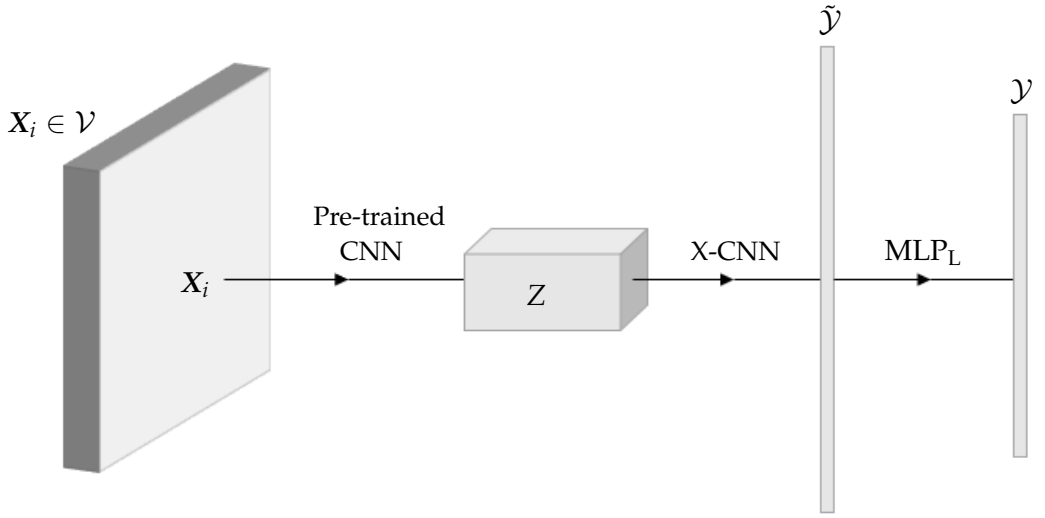


FIGURE 3.20: A simplified version of the proposed multilevel XAI architecture. An image $\boldsymbol{X}_i$ with a class label in $\mathcal{Y}$ is drawn from the dataset $\mathcal{V} \subseteq \mathcal{X}$. It is then passed to a pre-trained CNN. After passing through the last CNN layer, Z, it is passed to X-CNN, which produces a linguistic representation in $\tilde{\mathcal{Y}}$. Finally, it is passed to the MLP_L with a softmax output, giving a predicted label in $\mathcal{Y}$ for $\boldsymbol{X}_i$.

The pseudo-code for the training and test steps of our approach, i.e., X-MLP/X-CNN, is given in Algorithm 1. The pseudo-code for the data perturbation used to train the MLP_L is given in Algorithm 2.

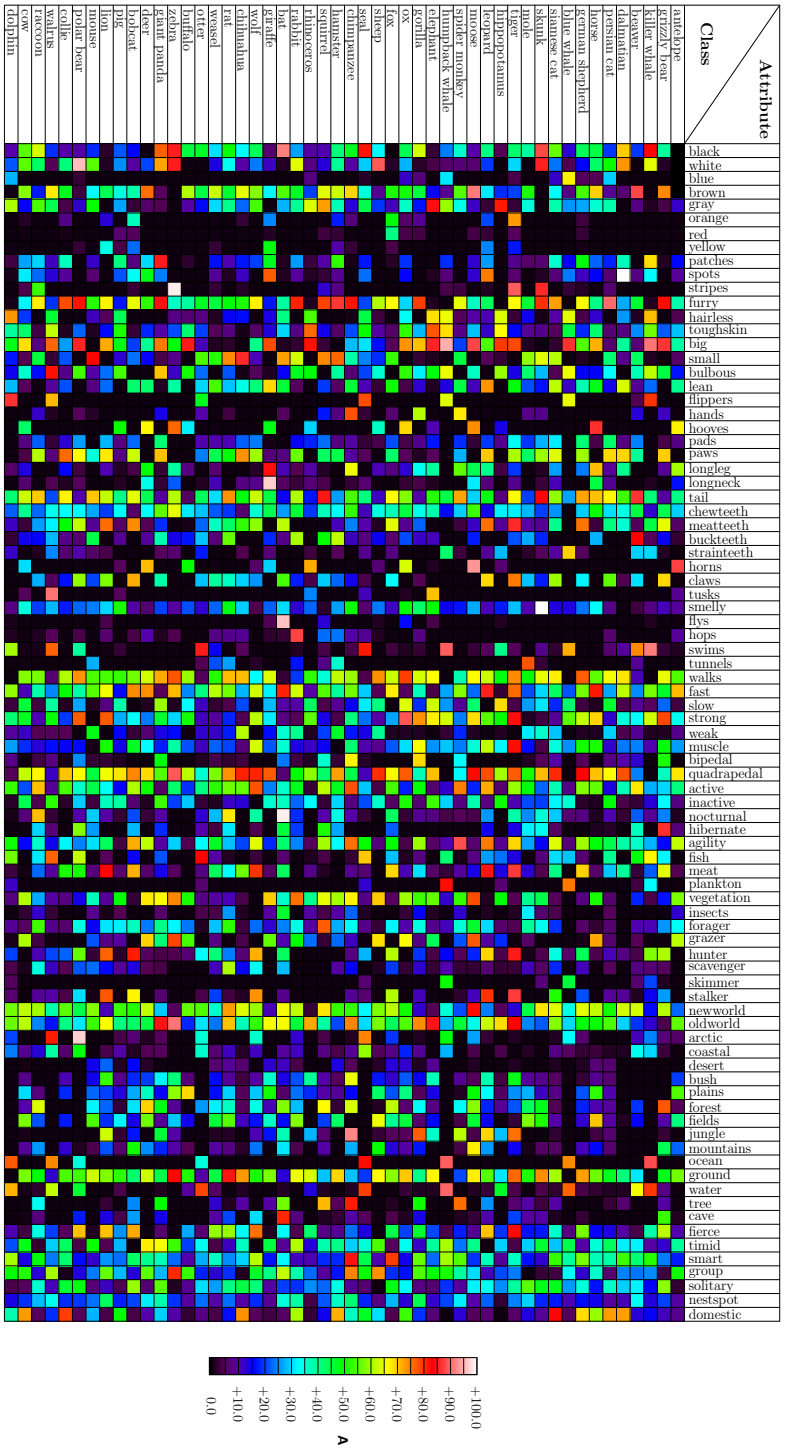
The full attribute-class matrix $\mathbf{A}$ annotated by experts for the AwA2 dataset is given in Table 3.3. The attribute-class prediction (averaged over the test samples for each class) by our approach for the AwA2 dataset is given in Table 3.4 (*cf.* Table 3.3).

3.6 Discussion and Limitations

To the best of our knowledge, the proposed approach is totally novel to XAI. It also raises a number of questions that are not commonly addressed in this context. Many of these open research questions we believe are important to the further development of XAI. Some of these are outlined below.

Linguistic alignment. We train the X-CNN network to find some visual features that separate the set of animals with high scores for a linguistic feature from those with low scores. We hope in doing so that the feature we learn captures some salient aspect of the linguistic feature. The extent to which we succeed we call the "linguistic alignment". Because we are learning this correlation in an unsupervised manner, we are not guaranteed that the visual feature aligns correctly with the linguistic label. This is likely to improve if we were to use more classes. On a small database of animals dominated

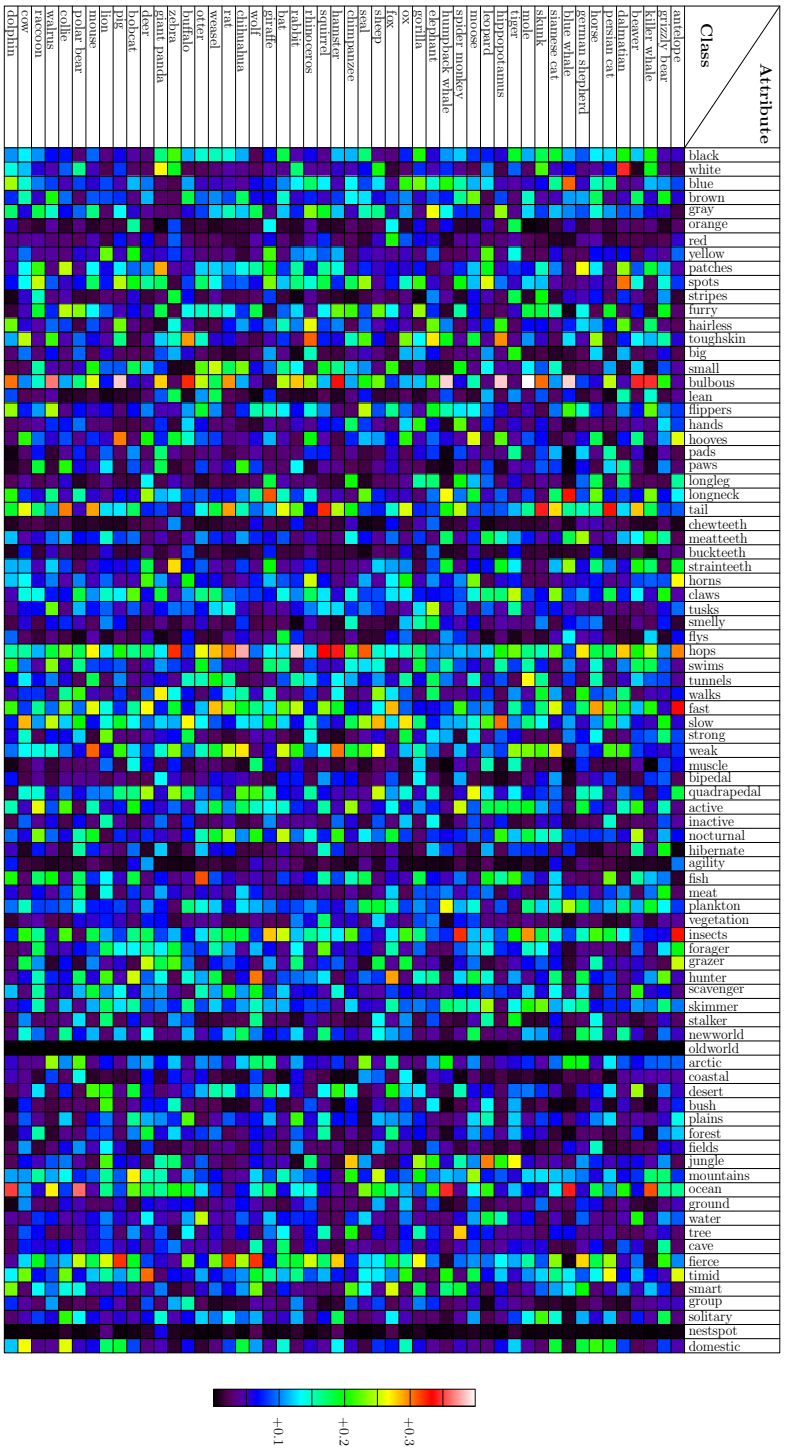
TABLE 3.3: Full attribute-class matrix **A** for the AwA2 dataset.



by undulates and fish, it would be easy to confuse the linguistic term "hooves" with "legs". Such confusion is much less likely if the dataset contained other animals, such as dogs or primates.

The linguistic alignment is complicated as a DNN is likely to assess circumstantial or contextual evidence for the existence of an attribute. As hooves are highly correlated

TABLE 3.4: Attribute-class prediction (averaged over the test samples for each class) by our approach for the AwA2 dataset (*cf.* Table 3.3).



with legs, it would not be surprising if the legs were considered highly salient to the presence of hooves in an image. This appears to be common in many of the attributes, where saliency maps appear to take in a much larger area than the feature described by the linguistic attribute. At some level, this clearly makes sense. A hoof-like object that is not attached to a leg is unlikely to be a hoof. Similarly, where a hoof is occluded, there

TABLE 3.5: Mutual information of attributes calculated by using test images and the prior knowledge given by experts in Table 3.3 separately. Symbol *** indicates the value is less than 0.01.

Attributes \ Info. via	*Test images*	*Experts*	Attributes \ Info. via	*Test images*	*Experts*
Black	0.38	0.95	*Muscle*	0.31	0.98
White	0.10	0.99	*Bipedal*	***	0.63
Blue	0.33	0.40	*Quadrupedal*	0.08	0.58
Brown	0.49	0.92	*Active*	***	0.82
Gray	0.02	0.99	*Inactive*	0.49	0.99
Orange	0.09	0.40	*Nocturnal*	0.43	0.88
Red	0.52	0.14	*Hibernate*	0.49	0.82
Yellow	0.40	0.40	*Agility*	0.42	0.92
Patches	0.14	0.88	*Fish*	0.09	0.92
Spots	***	0.79	*Meat*	0.46	0.97
Stripes	0.53	0.40	*Plankton*	0.35	0.32
Furry	***	0.76	*Vegetation*	0.23	0.99
Hairless	0.54	0.82	*Insects*	0.36	0.40
Toughskin	***	0.99	*Forager*	0.30	0.99
Big	0.34	0.95	*Grazer*	0.38	0.92
Small	0.13	0.98	*Hunter*	***	0.92
Bulbous	0.21	0.99	*Scavenger*	0.53	0.52
Lean	0.50	1	*Skimmer*	0.69	0.24
Flippers	0.15	0.58	*Stalker*	***	0.72
Hands	0.15	0.32	*Newworld*	0.30	0.68
Hooves	0.16	0.79	*Oldworld*	0.45	0.52
Pads	0.19	0.88	*Arctic*	0.07	0.68
Paws	0.51	0.99	*Coastal*	0.08	0.63
Longleg	0.50	0.85	*Desert*	0.59	0.14
Longneck	0.29	0.46	*Bush*	0.48	0.76
Tail	0.17	0.76	*Plains*	0.21	0.97
Chewteeth	***	0.76	*Forest*	0.04	0.98
Meatteeth	0.60	0.99	*Fields*	0.37	0.95
Buckteeth	***	0.79	*Jungle*	0.52	0.76
Strainteeth	0.38	0.52	*Mountains*	0.35	0.79
Horns	0.14	0.63	*Ocean*	0.39	0.63
Claws	0.15	0.98	*Ground*	***	0.68
Tusks	0.18	0.32	*Water*	0.14	0.72
Smelly	***	0.99	*Tree*	0.29	0.68
Flys	0.35	0.14	*Cave*	0.27	0.40
Hops	0.36	0.32	*Fierce*	0.25	0.98
Swims	0.32	0.72	*Timid*	***	0.92
Tunnels	0.14	0.46	*Smart*	0.51	0.90
Walks	0.57	0.72	*Group*	0.29	0.97
Fast	0.06	0.63	*Solitary*	0.31	0.98
Slow	0.55	0.97	*Nestspot*	0.20	0.97
Strong	0.24	0.90	*Domestic*	***	0.94
Weak	0.14	0.72			

may be enough context to infer that the animal is very likely to be hooved. However, this contextual evidence reduces the linguistic alignment. This is a feature of explainability rather than a fault of our approach. However, an important line of research in XAI is to separate circumstantial and contextual evidence from direct evidence.

The nature of explainability. Explanations for classifications are not unique. This was shown when we randomised the elements in the matrix, $\mathbf{A}$, between classes and linguistic attributes. In doing so, it seems highly unlikely that any attribute has a simple linguistic description. Yet, we can train our model with these random attributes

and still obtain classification levels of around 90% in AwA2. This seems at first sight counter-intuitive, although given that we have 85 continuous features they have the potential to carry sufficient information to separate the classes with high accuracy. What separates, at least, some of the true linguistic attributes from other attributes is the information content of the linguistic attributes; that is the linguistic attributes that have a high mutual information in regard to the classes. However, some of these attributes are hard for a DNN to learn.

What attributes DNNs learn. We have shown examples of attributes such as "rufous bill" that appear to be well captured by the networks. However, through using an independent test set we find that some of the linguistic attributes appear not to have been learned by the network. An example of this is "big", clearly an attribute that would be useful for humans to distinguish an elephant from a squirrel. Because of the nature of the training set, where objects tend to be resized to fill most of the image, this turns out to carry little information about the classes. However, the lack of success of these attributes is very informative regarding how DNNs perform a discriminative task. All the linguistic attributes used in the data are chosen by experts because linguistically they carry considerable mutual information about the classes. The failure of DNNs to exploit some of these terms obtained by our approach conveys important insights that directly address the issue of what information a neural network is actually learning – a core concern of XAI.

Tangible *vs* abstract attributes. The linguistic attributes used in this work (that we inherited from the zero-shot learning community) interestingly incorporate both tangible and abstract attributes. For the tangible attributes such as "horn", we would expect the corresponding saliency map to highlight the horn (although as we have argued it may highlight areas that are important contextual clues to the presence of a horn). The more abstract attributes such as "domestic" or "fast" are less easily attributed to a particular area of the image. They may however be highly informative for example in differentiating between cat and lion. When these attributes are informative then it is clearly important to understand whereabouts in the image these attributes are inferred. Again our approach goes some way towards addressing this issue.

Atomic and compound attributes. In our approach, we have treated all attributes as atomic. Consequently, pointy, fluffy and large ears would all be treated as separate linguistic attributes. However, each attribute would correspond to the same area of the image, and in many cases, it seems more natural to treat attributes as compound entities. We have not attempted to do this, but if we wish to scale up our approach to larger datasets, this seems to us to be an important area of future research.

3.7 Conclusion

High-performance DNNs are highly desirable when they can reason about their decisions. We presented a new XAI methodology—multilevel XAI—with self-explainable models delivering human-like multilevel explanations alongside the class probabilities. Explaining why a certain prediction is made using linguistic terms and attribute-wise saliency maps without requiring per-image ground-truth explanations in the training phase makes the proposed technique efficient and inexpensive. The results in explainability demonstrated by the match between image features and class embeddings greatly empower the explainability of DNNs while preserving their prediction ability at a reasonable level. Given the importance of XAI and the power of the newly introduced approach, we believe this could spark new avenues in XAI and shed light on developing and applying AI in more sensible ways.

Chapter 4

Semantic Proportions-based Semantic Segmentation

Although we obtain heatmaps that indicate regions important for classification thanks to our multilevel XAI approach, the precision of the heatmaps is not always high which may lead to ambiguity in the explanations. One way to address this is to use semantic segmentation technology to aid explanations. To explore the possibility of using this, we made a diversion into the field of semantic segmentation. Motivated by the ability of XAI heatmaps to identify relevant regions, we explored whether we can learn to perform semantic segmentation with minimum information. This leads us to develop SPSS (semantic proportions-based semantic segmentation). Here, we show that we can learn surprisingly accurate semantic segmentation using only knowledge of the proportion of each class in an image. This work contributes to the field of semantic segmentation outside of XAI. Nevertheless, we see this as an important first step in showing the potential of using semantic segmentation as part of an XAI framework.

Semantic segmentation is a critical task in computer vision aiming to identify and classify individual pixels in an image, with numerous applications in for example autonomous driving and medical image analysis. However, semantic segmentation can be highly challenging particularly due to the need for large amounts of annotated data. Annotating images is a time-consuming and costly process, often requiring expert knowledge and significant effort; moreover, saving the annotated images could dramatically increase the storage space. In this chapter, we propose a novel approach for semantic segmentation, requiring the rough information of individual semantic class proportions, shortened as *semantic proportions*, rather than the necessity of ground-truth segmentation maps. This greatly simplifies the data annotation process and thus will significantly reduce the annotation time, cost and storage space, opening up new possibilities for semantic segmentation tasks where obtaining the full ground-truth segmentation maps may not be feasible or practical. Our proposed method of utilising

semantic proportions can *(i)* further be utilised as a booster in the presence of ground-truth segmentation maps to gain performance without extra data and model complexity, and *(ii)* also be seen as a parameter-free plug-and-play module, which can be attached to existing DNNs designed for semantic segmentation. Extensive experimental results demonstrate the good performance of our method compared to benchmark methods that rely on ground-truth segmentation maps. Utilising semantic proportions suggested in this work offers a promising direction for future semantic segmentation research.

4.1 Introduction

Semantic segmentation is the task of partitioning an image into different regions depending on their semantic classes/categories. It is widely used in a variety of fields such as autonomous driving [108], medical imaging [156, 157], augmented reality [158] and robotics [159]. Impressive improvements have been shown in those areas with the recent development of DNNs, benefiting from the availability of extensive annotated segmentation datasets at a large scale [160, 161]. However, creating such datasets can be expensive and time-consuming due to the usual need to annotate pixel-wise labels as it takes between 54 and 79 seconds per object [125], thus requiring a couple of minutes per image with a few objects. Moreover, requiring full supervision is rather impractical in some cases, for example, in medical imaging where expert knowledge is required. Annotating 3D data for semantic segmentation is even more costly and time-consuming due to the additional complexity and dimensionality of the data, which generally requires voxel (i.e., point in 3D space) annotation. Skilled annotators from outsourcing companies that are dedicated to data annotation may be needed for specific requests to ensure annotation accuracy and consistency, adding further to the cost [162]. In addition, saving the annotated data could also be expensive given the substantial amount of storage space generally needed.

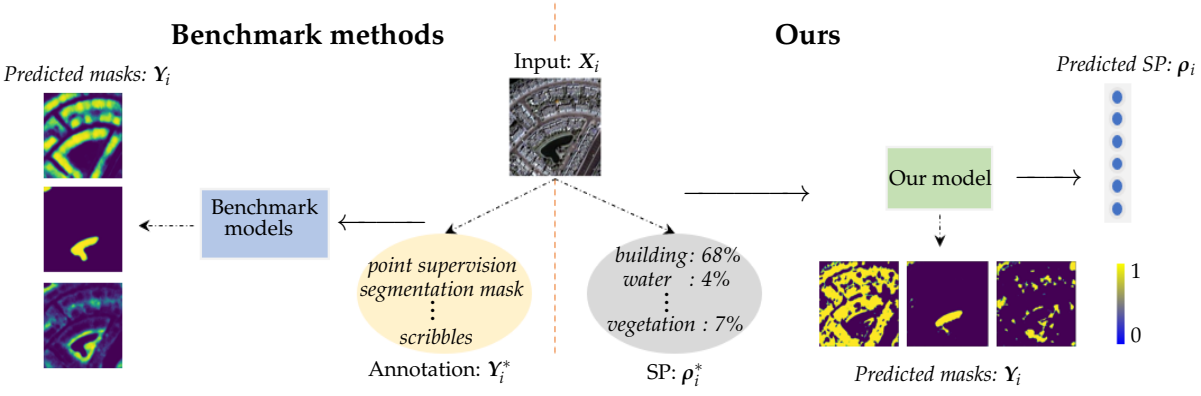


FIGURE 4.1: Difference between the proposed semantic segmentation approach and benchmark methods.

Different approaches have been proposed to reduce the fine-grained level (e.g. pixel-wise) annotation costs. One line of research is to train segmentation models in a weakly supervised manner by requiring image-level labels [163, 164], scribbles [122], eye tracks [165], or point supervision [125, 166] rather than costly segmentation masks of individual semantic classes. In contrast, in this chapter, we propose to utilise the proportion (i.e., percentage information) of each semantic class present in the image for semantic segmentation. For simplicity, we call this type of annotation *semantic (class) proportions* (SP). To the best of our knowledge, this is the first time utilising SP for semantic segmentation. This innovative way, different from the existing ways (see e.g. Figure 4.1), could significantly simplify and reduce the human involvement required for data annotation and storage space in semantic segmentation. Our proposed approach by utilising the SP annotation can achieve comparable and sometimes even better performance in comparison to benchmark methods with full supervision utilising ground-truth segmentation masks. Moreover, we show that our method can sometimes provide free performance improvement in the presence of ground-truth maps as it can serve as a plug-and-play module, which can easily be added on top of existing DNNs trained for segmentation tasks.

Our main contributions are: i) propose a new semantic segmentation methodology and a plug-and-play module, utilising SP annotations; ii) conduct extensive experiments on representative benchmark datasets from distinct fields to demonstrate the effectiveness and robustness of the proposed approach; and iii) draw an insightful discussion for semantic segmentation with weakly annotated data and future directions.

4.2 Related Work

Supervision levels in semantic segmentation. In recent years, more and more researchers have focused on reducing the annotation cost for semantic segmentation tasks. One way is to use weakly supervised learning techniques that require less precise or less expensive forms of supervision. For instance, the work in [164] proposed to utilise image-level labels, the work in [117, 119] used bounding boxes, and the methods in [122, 123] fed scribbles as labels instead of precise annotations to conduct semantic segmentation. Those approaches can significantly reduce the annotation cost, as they require less manual effort to annotate the data. However, there is always a trade-off between the annotation cost and the model performance, i.e., models trained with higher levels of supervision generally perform better than weakly supervised models. Active learning is an alternative approach to reduce the annotation cost by selecting the most informative samples to annotate based on the current model's uncertainty. With the selected most informative samples, active learning can reduce the amount of data that needs to be labelled, thus reducing the annotation cost [167]. It is worth mentioning that this is actually similar to the way we propose for the SP degraded by clustering presented

in Section 4.5.2.2. Reducing the annotation cost could also be achieved by generating synthetic data that can be used to augment the real-world data [168]. Synthetic data can be generated using e.g. computer graphics or other techniques to simulate realistic images and labels.

DNNs for semantic segmentation. The work in [112] made a breakthrough by proposing fully convolutional networks (FCNs) for semantic segmentation. FCNs utilise CNNs to transform input images into a probability map, where each entry of the probability map represents the likelihood of the corresponding image pixel belonging to a particular class. This approach allows the model to learn spatial features and eliminate the need for hand-crafted features. Following FCN, several variants have been proposed to improve the segmentation performance. For example, SegNet [169] is a modification of FCN employing an encoder-decoder architecture to achieve better performance; and DeepLab [113] introduced a novel technique called atrous spatial pyramid pooling to capture multi-scale information from the input image. U-Net [170], one of the architectures used in our proposed methodology, is a type of CNN consisting of a contracting path and an expansive path. The skip connections in U-Net allow the network to retain and reuse high-level feature representations learned in the contracting path, helping to improve segmentation accuracy. The U-Net architecture has been widely used for biomedical image segmentation tasks such as cell segmentation [171], organ segmentation [172] and lesion detection [173, 174], due to its ability to accurately segment objects within images while using relatively few training samples. Furthermore, its modular architecture and efficient training make it adaptable to a wide range of segmentation tasks. Therefore, to demonstrate our methodology utilising SP, we employ a modified and relatively basic version of the U-Net architecture as the backbone of our models for most of the experiments.

4.3 Methodology

Notation. Let $\mathcal{X}$ be a set of images. Without loss of generality, we assume each image in $\mathcal{X}$ contains no more than C semantic classes. $\forall \boldsymbol{X}_i \in \mathcal{X}$, $\boldsymbol{X}_i \in \mathbb{R}^{M \times H}$, where $M \times H$ is the image size. Let $\mathcal{X}_\mathrm{T} \subset \mathcal{X}$ and $\mathcal{X}_\mathrm{V} \subset \mathcal{X}$ be the training and validation (test) sets, respectively; and let $\Omega_\mathrm{T} \subset \mathbb{N}$ be the set containing the indexes of the images in $\mathcal{X}_\mathrm{T}$. $\forall \boldsymbol{X}_i \in \mathcal{X}_\mathrm{T}$, annotations are available. The most general annotation is the ground-truth segmentation maps, say $\{\boldsymbol{Y}^*_{ij}\}_{j=1}^C$, for $\boldsymbol{X}_i$, where each $\boldsymbol{Y}^*_{ij} \in \mathbb{R}^{M \times H}$ is a binary mask for the semantic class j of $\boldsymbol{X}_i$. For simplicity, let $\boldsymbol{Y}^*_i$ be a tensor formed by $\{\boldsymbol{Y}^*_{ij}\}_{j=1}^C$, where its j-th channel is $\boldsymbol{Y}^*_{ij}$. Note that the ground-truth segmentation maps are not required in our approach for semantic segmentation in this paper unless specifically stated; instead, they are mainly used by benchmark methods for comparison purposes. Analogously, let $\boldsymbol{Y}_i$ be the predicted segmentation maps following the same format as $\boldsymbol{Y}^*_i$. Let $\boldsymbol{\rho}^*_i = (\rho^*_{i1}, \cdots, \rho^*_{iC})$ be the given SP annotation of image $\boldsymbol{X}_i \in \mathcal{X}_\mathrm{T}$, which will be

mainly used to train our approach, where each $\rho_{ij}^* \in [0,1]$ is the SP of the j-th semantic class of $\boldsymbol{X}_i$ and $\sum_{j=1}^{C} \rho_{ij}^* = 1$.

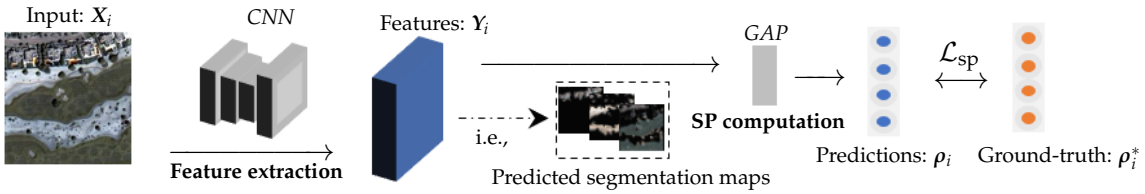


FIGURE 4.2: The SPSS (SP-based semantic segmentation) architecture. In the training stage, features are firstly extracted by a CNN from the input; and then the extracted features are through a GAP layer calculating the SP. After training using the loss function $\mathcal{L}_{\text{sp}}$, the proposed SPSS architecture can force the extracted features to be the prediction of the class-wise segmentation masks.

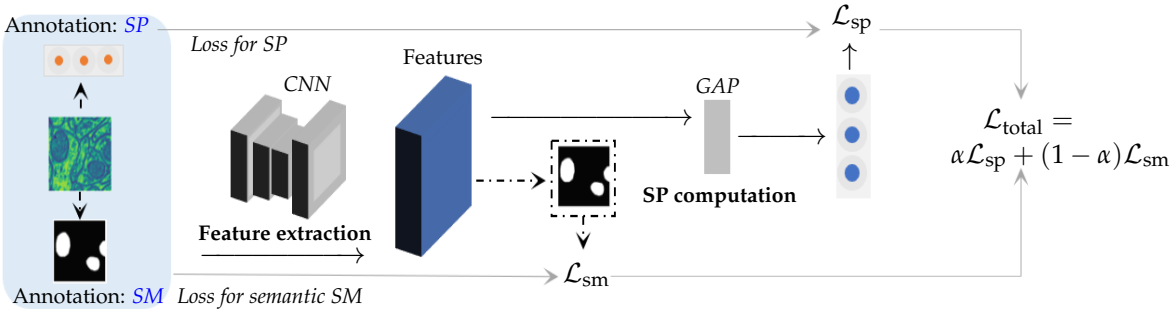


FIGURE 4.3: The SPSS+ architecture (*cf.* the SPSS architecture in Figure 4.2). In contrast, $\mathcal{L}_{\text{total}}$ (see Eq. (4.3)), a weighted average of $\mathcal{L}_{\text{sp}}$ and $\mathcal{L}_{\text{sm}}$, is calculated during training. After training, the SPSS+ architecture can force the extracted features to be the prediction of the class-wise segmentation masks.

Loss function. Two types of loss functions are introduced in the architectures of our method. One is based on the mean squared error (MSE). MSE is commonly used to evaluate the performance of regression models where there are numerical target values to predict. We employ MSE to measure the discrepancy between the ground-truth SP and the predicted ones. For ease of reference, we call this loss function $\mathcal{L}_{\text{sp}}$ throughout the chapter, i.e.,

$$\mathcal{L}_{\text{sp}} = \frac{1}{|\Omega_{\text{T}}|} \sum_{i \in \Omega_{\text{T}}} \|\boldsymbol{\rho}_i^* - \boldsymbol{\rho}_i\|^2, \tag{4.1}$$

where $\boldsymbol{\rho}_i$ is the predicted SP for image $\boldsymbol{X}_i \in \mathcal{X}_{\text{T}}$ and $|\Omega_{\text{T}}|$ is the cardinality of set Ω_{T}. The other loss function, which will be deferred in Section 4.3.2, is defined based on the binary cross-entropy (BCE). BCE is a commonly used loss function in binary classification problems and measures the discrepancy between the predicted probabilities and the true binary ones. Below we define the BCE function as

$$\mathcal{L}_{\text{sm}} = \frac{1}{|\Omega_{\text{T}}|} \sum_{i \in \Omega_{\text{T}}} \sum_{j=1}^{C} -(\boldsymbol{Y}_{ij}^* \log(\boldsymbol{Y}_{ij}) \; + \; (1 - \boldsymbol{Y}_{ij}^*) \log(1 - \boldsymbol{Y}_{ij})), \tag{4.2}$$

where $\boldsymbol{Y}_{ij}$ is the predicted segmentation map for the j-th semantic class of image $\boldsymbol{X}_i \in \mathcal{X}_{\mathrm{T}}$.

4.3.1 Proposed SP-based Semantic Segmentation Architecture

The proposed SPSS architecture is shown in Figure 4.2. It contains two main parts. The first part of the SPSS architecture is feature extraction. Employing a CNN is a common approach in current state-of-the-art semantic segmentation methods. In our SPSS, a CNN (or other type of DNNs) is utilised as its backbone to extract high-level image features $\boldsymbol{Y}_i$ from the input image $\boldsymbol{X}_i$. The second part of the SPSS architecture is a GAP layer, which takes the image features $\boldsymbol{Y}_i$ to generate the SP, $\boldsymbol{\rho}_i$, for the input image $\boldsymbol{X}_i$. The SPSS architecture is then trained by using the loss function $\mathcal{L}_{\mathrm{sp}}$ defined in Eq. (4.1). After training the SPSS architecture, the extracted features $\boldsymbol{Y}_i$ of the trained CNN are, surprisingly, the prediction of the class-wise segmentation masks; that is how the SPSS architecture performs semantic segmentation by just using the SP rather than the ground-truth segmentation maps.

We remark that both parts in the SPSS architecture except for utilising SP are well-known and commonly employed for e.g. computer vision tasks. To the best of our knowledge, it is, for the first time, to combine them for semantic segmentation in reducing the need for labour-intensive (fine-grained) ground-truth segmentation masks to the (coarse-grained) SP level.

4.3.2 A Booster: SPSS+

The proposed SPSS architecture in Figure 4.2 only uses the SP annotation for semantic segmentation, which is quite cheap in terms of annotation generation. Moreover, SPSS is also very flexible. For example, i) the proposed loss function $\mathcal{L}_{\mathrm{sp}}$ using SP can be employed as a plug-and-play module in different DNNs; and ii) SPSS can be enhanced directly when additional annotation information is available. Below we give a showcase regarding how to use SP and pixel-level annotations jointly to enhance the SPSS architecture, see Figure 4.3. For ease of reference, we call the proposed booster in Figure 4.3 *SPSS+*.

The total loss for the SPSS+ architecture is

$$\mathcal{L}_{\mathrm{total}} = \alpha \mathcal{L}_{\mathrm{sp}} + (1 - \alpha)\mathcal{L}_{\mathrm{sm}}, \tag{4.3}$$

where α is an adjustable weight to determine the trade-off between $\mathcal{L}_{\mathrm{sp}}$ and $\mathcal{L}_{\mathrm{sm}}$. The SPSS+ architecture uses the loss $\mathcal{L}_{\mathrm{total}}$, which considers the annotations of the SP and segmentation masks for training. Similar to the SPSS architecture (in Figure 4.2), the

extracted features $\mathbf{Y}_i$ of the trained CNN in the SPSS+ architecture are the prediction of the class-wise segmentation masks, i.e., the semantic segmentation results.

Our SPSS can generally achieve comparable performance against benchmark semantic segmentation methods. SPSS+ works as a performance booster and improves the segmentation ability of SPSS without extra training data or model complexity. More details regarding the extensive validation and comparison are given in Section 4.5.

4.4 Data and Settings

4.4.1 Data

The proposed SP-based methodology for semantic segmentation is showcased on four different datasets described below.

(i) Satellite images of Dubai, i.e., `Aerial Dubai`. This is an open-source aerial imagery dataset presented as part of a Kaggle competition[1]. The dataset includes 8 tiles and each tile has 9 images of various sizes and their corresponding ground-truth segmentation masks for 6 classes, *i.e., building, land, road, vegetation, water and unlabeled*.

(ii) Medical imaging dataset `ISIC` (International Skin Imaging Collaboration). This is a comprehensive collection of dermoscopic images specifically curated for the study and analysis of skin lesions [175, 176]. It contains 2594 training, 100 validation and 1,000 test images with high-resolution capturing various types of skin lesions, including benign and malignant conditions. Each image in the dataset is accompanied by expert annotations including detailed segmentation masks outlining the precise boundaries of the lesions. These annotations are crucial for segmentation methods to accurately delineate the lesion from the surrounding skin. The `ISIC` dataset is frequently used in research and competitions, such as the `ISIC` Challenge, to benchmark and advance segmentation algorithms. However, obtaining fine-grained pixel-level segmentation masks is expensive and our SPSS model shows comparable performance despite being trained with dramatically less expensive SP rather than full masks.

(iii) Medical imaging dataset `Electron Microscopy`[2]. It contains 165 slices of microscopy images with a size of 768×1024. The primary aim of this medical dataset is to identify and classify mitochondria pixels. This dataset is quite challenging since its semantic classes are severely imbalanced, i.e., the size of the mitochondria in most slices is very small (e.g. see the right column of Figure 4.4 and Figure 4.7).

(iv) Medical imaging dataset `LGG Brain MRI` from The Cancer Genome Atlas (TCGA) and The Cancer Imaging Archive (TCIA). We used the version made available by Buda

[1] https://www.kaggle.com/datasets/humansintheloop/semantic-segmentation-of-aerial-imagery
[2] https://www.epfl.ch/labs/cvlab/data/data-em/

et al. [177] on Kaggle[3], where the authors selected 120 patients from the TCGA lower-grade glioma collection[4] which had available preoperative imaging data including at least a fluid-attenuated inversion recovery (FLAIR) sequence. The dataset includes roughly 4000 brain MRI images of 110 patients from 5 institutions. Figure 4.4 presents some example images for the three medical imaging datasets.

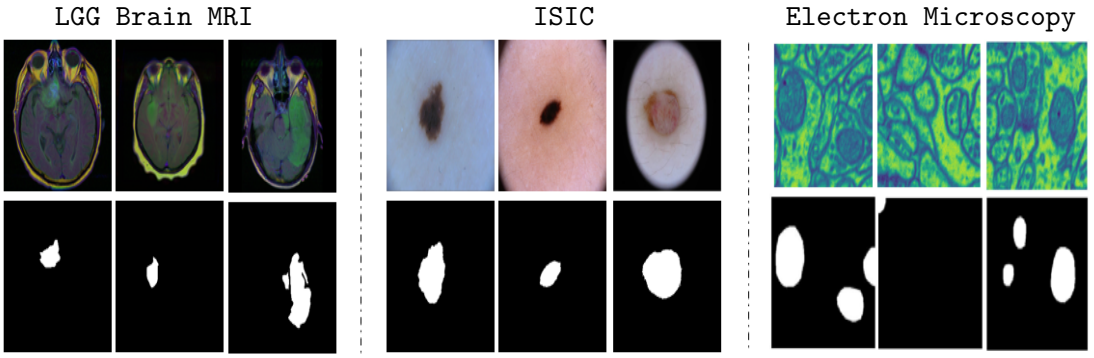


FIGURE 4.4: Example images and ground-truth segmentation masks of the three employed medical imaging datasets.

For the use of medical imaging datasets `ISIC`, `LGG Brain MRI` and `Electron Microscopy`, an ethical clearance from the University of Southampton was obtained with the ERGO number 100585.

4.4.1.1 Data preprocessing

The `Aerial Dubai` and `Electron Microscopy` datasets contain large images that were preprocessed into smaller patches for analysis. Specifically, each image in the `Aerial Dubai` dataset was divided into 224×224 pixel patches, resulting in a total of 1,647 images. For the `Electron Microscopy` dataset, images were divided into 256×256 pixel patches, yielding 1,980 images. The images in the `LGG Brain MRI` dataset, originally sized at 256×256 pixels, were centre-cropped to 144×144 pixels. Subsequently, images from all datasets including `ISIC` were then resized to 288×288 pixels. This preprocessing ensures uniformity in image sizes across different datasets, facilitating consistent and effective analysis.

4.4.2 Experimental Settings

Benchmark methods with different CNN backbones (e.g., U-Net [170] or Feature Pyramid Network (FPN) [178] with VGG16 [58] and ResNet34 [60]) are trained end-to-end for semantic segmentation using the ground-truth segmentation masks, comparing to

[3] https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation
[4] https://cancergenome.nih.gov/cancersselected/lowergradeglioma

ours using the SP. For a fair comparison, the same training images are used to train all the models.

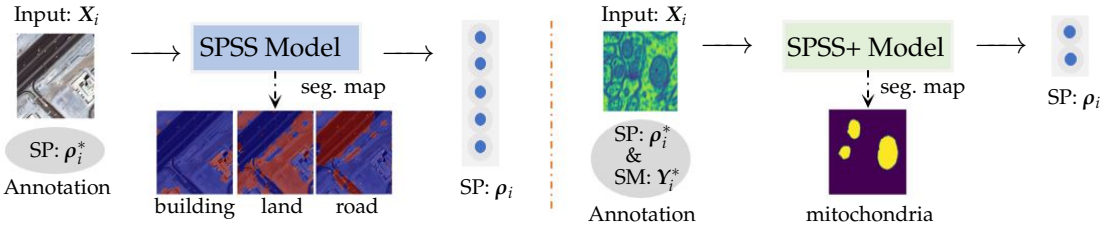


FIGURE 4.5: Diagrams of the proposed models SPSS and SPSS+ on the datasets `Aerial Dubai` (*left*) and `Electronic Microscopy` (*right*; significant class imbalance), respectively.

4.4.2.1 Deep neural architecture details

- We employed U-Net [170] and FPN [178] architectures with pre-trained weights from VGG16 [58] and ResNet34 [60] on the `Aerial Dubai` dataset. For the medical imaging datasets and all the ablation experiments presented in Section 4.5, we consistently utilized a U-Net with VGG16 weights.
- To adapt U-Net and FPN for predicting SP rather than fine-grained masks, a 1×1 convolutional layer with n filters is employed to match the C number of the semantic classes. Thus n is set to 6 and 1 to output feature maps of the size $288 \times 288 \times 6$ and $288 \times 288 \times 1$ respectively for the `Aerial Dubai` and medical imaging datasets. Note that there is no need to set n to 2 for the binary segmentation problem with medical imaging datasets. Finally, a GAP layer is added on top to get n float to be used as the predicted SP values.
- To obtain segmentation maps during the test stage, we extract the feature maps prior to the GAP layer and visualise them per semantic class (*cf.* Figures 4.2 and 4.3).

4.4.2.2 Training setup

For all experiments, an 80/20 split for the training/test, Adam optimizer with a learning rate of 10^{-3}, and a batch size of 16 were chosen. The number of epochs was set to 100 with early stopping applied with patience set to 10 based on the validation loss. All the experiments were implemented on a personal laptop with the following specifications: i7-8750H CPU, GeForce GTX 1060 GPU and 16GB RAM. Training of SPSS and SPSS+ takes around 30 minutes and 40 minutes, respectively.

4.5 Experiments

We highlight that the main aim here is to show that semantic segmentation can be achieved with significantly weaker annotations, i.e., the SP annotation, rather than segmentation accuracy enhancement only. Recall that the difference between SPSS and SPSS+ is just the way of using the annotations for their training, i.e., SPSS+ addresses scenarios in which ground-truth segmentation maps are available. Figure 4.5 illustrates the difference by utilising the SPSS and SPSS+ architectures on the datasets `Aerial Dubai` and `Electronic Microscopy`, respectively. To demonstrate the effectiveness of our semantic segmentation approach, we evaluate performance using mean Intersection over Union (IoU) and F1 scores.

4.5.1 Segmentation Performance Comparison

Quantitative comparison. Tables 4.1 and 4.2 give the quantitative results of our method and the benchmark methods for the `Aerial Dubai` and the three medical imaging datasets, respectively. Well-known evaluation metrics, i.e., Mean IoU and F1 scores are employed. Estimated errors in the mean are obtained by training the models three times with randomly initialised weights. Tables 4.1 and 4.2 show that SPSS performs comparably to the benchmark methods for all tasks, demonstrating the utility of the SP annotation for semantic segmentation that our methodology introduces. Moreover, SPSS+, i.e., using both ground-truth maps and SP, outperforms the benchmark methods for all the cases except for using the FPN with VGG16 backbone, indicating the usefulness of involving the SP annotation. Note again that SPSS+ does not require any additional data collection or increase in model complexity, hence offering performance improvements for semantic segmentation tasks nearly for free. Without loss of generality, U-Net with VGG16 is adopted in our method for the rest of the experiments.

TABLE 4.1: Quantitative semantic segmentation results (Mean IoU and F1 scores) on the `Aerial Dubai` dataset.

Model	U-Net				FPN			
Backbone	VGG16		ResNet34		VGG16		ResNet34	
Metric	Mean IoU	F1	Mean IoU	F1	Mean IoU	F1	Mean IoU	F1
Benchmark	71.3 ± 1.2	88.3 ± 0.7	69.2 ± 0.8	86.1 ± 1.2	$\mathbf{68.5 \pm 0.5}$	$\mathbf{82.1 \pm 0.3}$	67.2 ± 0.8	81.3 ± 0.8
SPSS	64.2 ± 0.6	83.7 ± 0.4	64.4 ± 0.4	80.6 ± 0.8	60.5 ± 0.2	77.2 ± 0.4	61.7 ± 0.6	77.5 ± 1.1
SPSS+	$\mathbf{71.6 \pm 0.6}$	$\mathbf{88.7 \pm 0.6}$	$\mathbf{70.4 \pm 0.5}$	$\mathbf{86.4 \pm 0.3}$	67.7 ± 1.2	80.5 ± 0.5	$\mathbf{69.2 \pm 1.0}$	$\mathbf{82.5 \pm 0.7}$

TABLE 4.2: Quantitative semantic segmentation results (Mean IoU scores) on the medical imaging datasets using U-Net with VGG16 backbone.

Method \ Data	`ISIC`	`Mithocondria`	`Brain MRI`
Benchmark	78.4 ± 0.3	83.7 ± 0.6	72.3 ± 0.2
SPSS	73.2 ± 0.5	76.5 ± 0.2	69.5 ± 0.6
SPSS+	$\mathbf{79.1 \pm 0.1}$	$\mathbf{84.3 \pm 0.5}$	$\mathbf{72.8 \pm 0.4}$

Qualitative comparison. Figure 4.6 shows the qualitative results of our method and the benchmark method for the `Aerial Dubai` dataset. Surprisingly, the class-wise segmentation maps that our method achieves (middle of Figure 4.6) are visually significantly better than that of the benchmark method (right of Figure 4.6) in terms of the binarisation ability, indicating the effectiveness of the loss $\mathcal{L}_{\text{sp}}$ (defined in Eq. (4.1)) using the SP annotation we introduce. For the significant class imbalance dataset `Electronic Microscopy`, Figure 4.7 shows the qualitative results of our method and the benchmark method for some challenging cases. Again, our method exhibits superior performance against the benchmark method. For example, our method can accurately segment the mitochondria on the top-left corner of the second image despite employing much less annotation, but the benchmark method completely misses it despite being trained using the ground-truth segmentation masks. This again validates the effectiveness of the SP annotation for semantic segmentation. Moreover, due to the great binarisation ability of the loss $\mathcal{L}_{\text{sp}}$ using SP, it may serve as an auxiliary loss functioning as a plug-and-play module even in scenarios where ground-truth segmentation masks are available to enhance the segmentation performance of many existing methods as SPSS+ does.

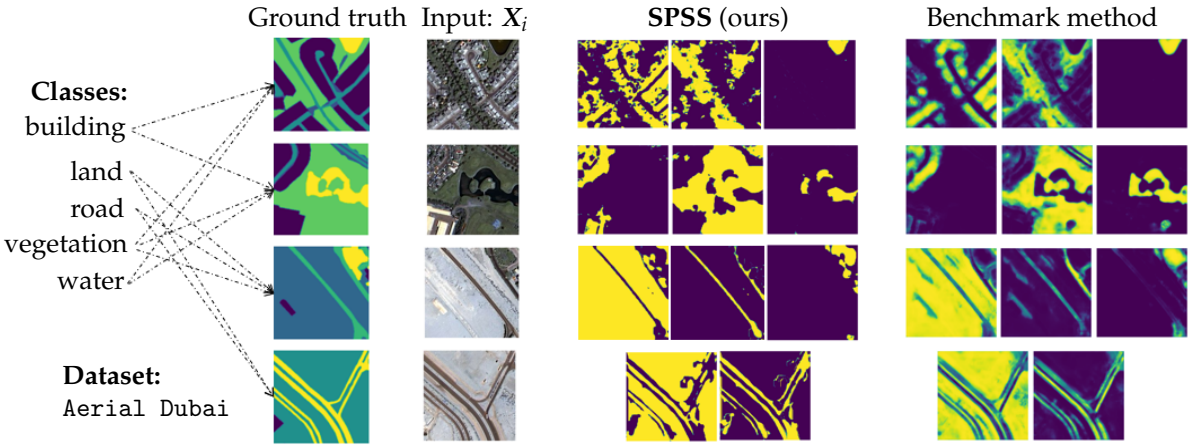


FIGURE 4.6: Qualitative semantic segmentation comparison between our SPSS method (*middle*) and the benchmark method (*right*).

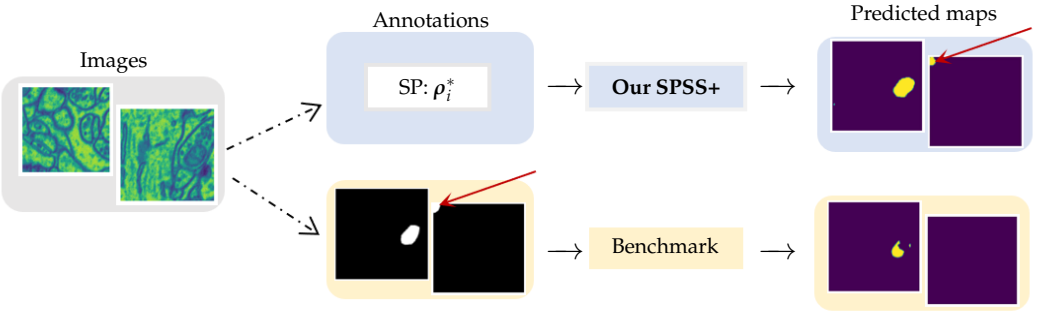


FIGURE 4.7: Comparison between our SPSS+ method (*upper*) and the benchmark method (*lower*) on some images from the `Electronic Microscopy` dataset.

4.5.2 Sensitivity Analysis

Obtaining precise SP annotations may be challenging and, as a result, annotators may provide rough estimates instead. We showcase that rough estimated SP is quite sufficient for our model to achieve good performance (further results are deferred in Section 4.5.3). Below we first investigate the robustness of our models corresponding to the quality of the SP. Two extreme ways of degrading the SP are examined: one is adding noises to the SP directly and the other is assigning images in individual clusters to the same SP.

4.5.2.1 SP degraded by different noise

We first conduct sensitivity analysis of our method SPSS by systematically adding Gaussian noise to the SP for the `Aerial Dubai` dataset. Let $\mathcal{N}(0,\sigma)$ be the normal distribution with 0 mean and standard deviation σ. For the given SP $\boldsymbol{\rho}_i^* = (\rho_{i1}^*, \cdots, \rho_{iC}^*)$ of $\forall \boldsymbol{X}_i \in \mathcal{X}_\mathrm{T}$, let $\tilde{\boldsymbol{\rho}}_i^* = (\tilde{\rho}_{i1}^*, \cdots, \tilde{\rho}_{iC}^*)$, where

$$\tilde{\rho}_{ij}^* = \rho_{ij}^* + \mathcal{N}(0,\sigma), \;\; j = 1, \cdots, C. \tag{4.4}$$

Then the softmax operator is used to normalise $\tilde{\boldsymbol{\rho}}_i^*$, and the normalised $\tilde{\boldsymbol{\rho}}_i^*$ is used as the new SP to train our model. Here the standard deviation σ controls the level of the Gaussian noise being added to the SP; e.g., $\sigma = 0.1$ represents 10% Gaussian noise. Table 4.3 showcases the robustness of our methodology, as it continues performing well even with the SP degraded by quite high levels of noise. Our method suffers a drop in performance of $\sim$ 4% for 10% Gaussian noise being added to the SP. Our method still works significantly above random guessing even with the SP which is degraded by 50% Gaussian noise. This shows that our method is quite robust corresponding to the SP, which means the annotators could in practice spend much less effort for providing rough SP rather than the precise SP.

TABLE 4.3: Performance of our model in terms of Mean IoU trained by using the SP degraded by Gaussian noise.

	Dataset `Aerial Dubai`							
Noise (%)	0	5	10	15	20	30	40	50
Mean IoU	64.2	62.4	60.1	57.8	52.2	48.3	43.4	38.3

For medical imaging datasets, the SP of the positive class region, i.e., ρ_{i1}^*, is degraded by a different noise generation process to present diverse noise injection scenarios. Noise is added in a controlled manner utilising the uniform distribution $\mathcal{U}(a,b)$ bounded by a and b, ensuring that the degraded SP remains within a specified range, i.e.,

$$\tilde{\rho}_{i1}^* = \rho_{i1}^* + \lambda\, \mathcal{U}(a,b)\rho_{i1}^*, \tag{4.5}$$

Algorithm 3 Noisy SP $\tilde{\boldsymbol{\rho}}_i^*$ Generation

Input: Ground-truth SP $\boldsymbol{\rho}_i^*$ of image $\boldsymbol{X}_i$, standard deviation σ, lower bound a, and upper bound b.
Output: Noisy SP $\tilde{\boldsymbol{\rho}}_i^*$
if length$(\boldsymbol{\rho}_i^*) == 1$ **then** ▷ E.g., medical imaging datasets
    Randomly select λ from $\{-1, 1\}$;
    $\tilde{\rho}_{i1}^* = \rho_{i1}^* + \lambda\, \mathcal{U}(a, b)\rho_{i1}^*$;
else ▷ E.g., `Aerial Dubai` dataset
    for $j = 1$ to length$(\boldsymbol{\rho}_i^*)$ **do**
        $\tilde{\rho}_{ij}^* = \rho_{ij}^* + \mathcal{N}(0, \sigma)$;
    end for
end if
return $\tilde{\boldsymbol{\rho}}_i^*$

where λ is a parameter with value -1 or 1 selected randomly. The above way ensures that the degraded SP is relative to the size of the original SP controlled by bounds a and b. The above steps are also summarised in Algorithm 3. The results presented in Table 4.4 again show that our method SPSS is robust against a high level of noise imposed on the SP.

TABLE 4.4: Performance of our model in terms of Mean IoU trained by using the degraded SP for medical imaging datasets.

Noise ($[a, b]$) \ Data	`ISIC`	`Mithocondria`	`Brain MRI`
Noise free	73.2	76.5	69.5
$[0, 0.5]$	70.1	70.5	62.5
$[0, 1]$	67.3	66.2	60.1
$[0.5, 1]$	69.3	64.2	63.1

4.5.2.2 SP degraded by clustering

We now conduct the sensitivity analysis of our method by degrading the SP of the training images by clustering. The degradation procedures are: i) clustering the set of the given SP, i.e., $\{\boldsymbol{\rho}_i^*\}_{i\in\Omega_{\mathrm{T}}}$, into K clusters by K-means; ii) clustering the training set $\mathcal{X}_{\mathrm{T}}$ into the same K clusters, say $\mathcal{X}_{\mathrm{T}}^k, k = 1, \ldots, K$, corresponding to the SP clusters; and iii) assigning all the training images in cluster $\mathcal{X}_{\mathrm{T}}^k$ the same SP which is randomly selected from the SP of one image in this cluster; see also Figure 4.8 for illustration. Obviously, implementing this way of degrading the SP, all the images' SP in the training set $\mathcal{X}_{\mathrm{T}}$ are changed except for K (i.e., the number of clusters) images if every training image has different SP annotation in the original SP set. The smaller the number K, the more severe the SP degradation.

The performance of our method regarding the SP degraded by clustering is shown in Table 4.5, indicating again the robustness of our methodology corresponding to the SP. For example, after just using $K = 100$ images' SP for the whole training set $\mathcal{X}_{\mathrm{T}}$, the

Mean IoU of our method only drops by $\sim$ 2.5%; and just using $K = 5$ images' SP for the whole training set, our method can still work to some extent (i.e., the Mean IoU just drops less than half). This again shows that our method is indeed quite robust corresponding to the SP. This suggests one possible strategy to reduce effort is to cluster images (for example from patients with a similar level of disease) and then estimate SP on representative images in the cluster.

TABLE 4.5: Performance of our model in terms of Mean IoU trained by using the SP degraded by clustering.

	Dataset `Aerial Dubai`					
# Clusters K	100	50	30	20	10	5
Mean IoU	61.7	59.4	56.5	51.2	47.4	38.3

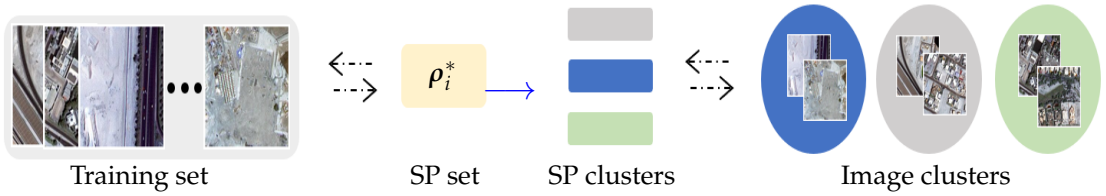


FIGURE 4.8: Diagram of the SP annotation degraded by clustering. Images are clustered corresponding to the SP clusters which are achieved by applying K-means on the SP set. An SP annotation for one image in each image cluster is then randomly selected from that cluster and is assigned to all the images in that image cluster.

4.5.3 Further Comparison and Analysis

For demonstration purposes, the SP information used in the previous experiments is simply obtained from the given annotated ground-truth segmentation masks. Certainly, in practice, we need the estimated SP information directly from annotators rather than from the ground-truth segmentation masks and thus significantly simplify the data annotation process. Below we showcase that rough estimated SP directly from annotators can indeed be obtained efficiently and cheaply and is quite sufficient for our models to achieve good performance.

To directly obtain the SP annotations (in the absence of ground-truth masks), 52 images were randomly picked from the `Aerial Dubai` dataset, and then three annotators (the author of this thesis and two supervisors) estimated the SP for the provided images. The estimated SP scores were then averaged. Afterwards, data augmentation techniques such as flipping and rotation were applied to obtain 416 images for training. Further details of the annotation process are given below.

The three annotators were asked to annotate a small batch containing 52 images from the `Aerial Dubai` dataset each with the size of 288×288 to show the efficiency of the SP annotation process compared to the pixel-wise annotation, as well as the excellent

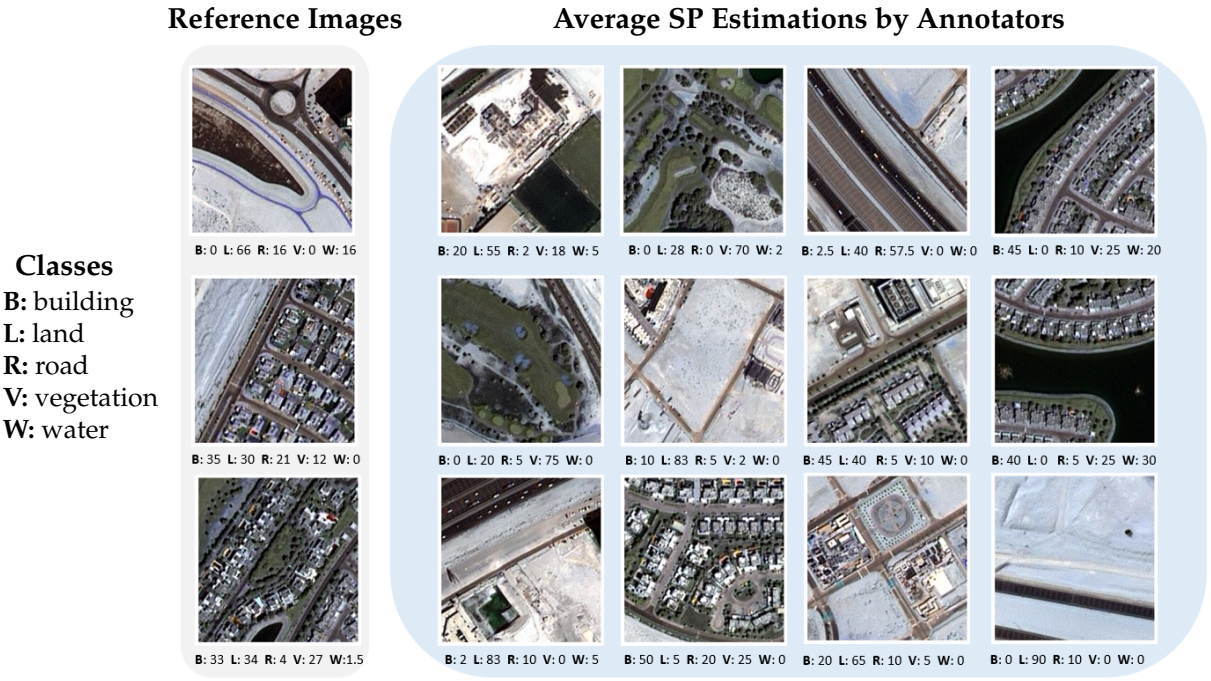


FIGURE 4.9: Showcase of the SP annotation process by annotators directly. Three annotators were asked to annotate a batch with 52 images for training. Left: reference images whose SP information is calculated from the pixel-wise annotated ground-truth segmentation maps. Right: some randomly selected images with their average SP estimations by the three annotators.

semantic segmentation ability of the proposed SPSS model compared to the benchmark model (with the ground-truth segmentation maps).

- The annotators were provided with three reference images whose SP information is simply obtained via the pixel-wise segmentation maps, see the left of Figure 4.9. The reference images could be helpful for annotators to adjust their estimations; for instance, for the last image in the first row of Figure 4.9 regarding the SP estimations, it is clear that the water area is a little larger than that in the first reference image, which helps the annotators to estimate a proportion with a larger value than that for the water area in the reference image (i.e., 20% vs. 16%). The average estimation of the three annotators for the water area in the mentioned image is around 20%, which is quite close to the value obtained by its ground-truth map, i.e., 21.3%, showing the efficiency of the SP annotation directly by annotators in this manner. Moreover, our sensitivity experiments showed that obtaining precise SP information for training is not a must for our SPSS model to perform well, making the SP annotation process even more efficient and relaxing given its tolerance of rough deviation in the SP estimations.
- After each annotator completes their SP annotation, the average SP annotation of the three annotators is obtained for the 52 images.

- Finally, two types of augmentation strategies were carried out to increase the training dataset size. Each image was flipped horizontally and rotated by 90, 180 and 270 degrees clockwise. The rotations were also applied to every flipped image. Therefore, 8 images were obtained for every image, and a training dataset consisting of 416 images in total was formed. Note that, since the SP information is irrelevant to the position of the content in an image, the estimated SP for one image is also applied to all of its 7 augmented versions.

Table 4.6 highlights the time and memory cost to produce the SP annotations compared to producing the ground-truth segmentation masks.

Pixel annotation for a single image with 5 objects takes roughly 330 seconds which is around 16 times more than the time required for SP annotation[5]. Regarding memory, a mask with the size of 224×224 takes up around 148 kB. With compression, this value can drop to as low as 4 kB, which is still roughly 200 times larger than the SP which consists of only 5 numbers. This huge efficiency brought by our proposed SP strategy is quite significant, particularly for big datasets which are required for semantic segmentation.

TABLE 4.6: Comparison between the annotation styles of obtaining the segmentation masks and the SP in terms of time and memory. The `Aerial Dubai` dataset is used.

Annotation style	Average time per image	Memory per image	
		Original	Compressed
Segmentation masks	$\sim$ 330s	$\sim$ 148 kB	$\sim$ 4 kB
SP (via annotators)	$\sim$ 20s	$\sim$ 0.02 kB	

We now further compare the semantic segmentation performance between the benchmark model with ground-truth segmentation maps and our SPSS with the SP simply obtained from the ground-truth segmentation maps and the rough SP produced by the annotators, separately. Table 4.7 presents the results on the same test set used in Table 4.1. The results are quite impressive as SPSS with the rough SP estimations surpasses not only the way of using the SP obtained by the ground-truth maps but also the benchmark model trained using the costly ground-truth maps.

TABLE 4.7: Quantitative comparison on the `Aerial Dubai` dataset with rough estimated SP annotations.

Model	Mean IoU	Per-class F1 score					Mean accuracy
		Building	Land	Road	Vegetation	Water	
Segmentation masks	39.5 ± 1.3	$\mathbf{52.7 \pm 1.2}$	84.8 ± 0.6	2.4 ± 0.6	43.2 ± 1.3	75.4 ± 0.5	67.9 ± 1.1
SP (via seg. masks)	37.9 ± 0.8	39.8 ± 1.3	84.6 ± 0.3	4.5 ± 0.2	41.3 ± 0.8	77.2 ± 0.9	67.4 ± 0.3
SP (via annotators)	$\mathbf{41.6 \pm 1.3}$	46.2 ± 0.7	$\mathbf{85.7 \pm 1.3}$	$\mathbf{26.6 \pm 2.1}$	$\mathbf{44.3 \pm 0.8}$	$\mathbf{75.6 \pm 0.3}$	$\mathbf{68.7 \pm 0.4}$

[5] Average time taken for per-pixel annotation is estimated based on [125].

4.6 Discussion and Limitations

SP (semantic proportions) for each training image is required as annotation/label information for the presented semantic segmentation model. In this work, we obtained these proportions from both the segmentation maps available for the chosen datasets and three annotators directly to demonstrate the effectiveness and robustness of our proposed SP-based methodology. We would like to stress that the reason why we benefited from the existing segmentation maps, which seems controversial to our main aim at first glance, is to show that the proposed methodology is feasible in the presence of SP. Arguably, reasonable proportions can be simply extracted from the ground-truth segmentation maps if they are annotated properly. Therefore, obtaining SP from the readily available maps to achieve our aim is sensible. Clearly, our goal is to train our proposed model when the segmentation maps are unavailable. It is evident from our experiments that obtaining SP annotation could be much cheaper than obtaining precise segmentation maps, particularly for data volumes in high dimensions. There are obviously various ways to obtain SP readily in the absence of the segmentation maps, such as by employing mechanical turks. There may exist applications such as estimating the density of housing in a particular area where information may be extracted from other studies or even obtained from pre-trained large language models, e.g., GPT-3 [144].

The results that we present in Section 4.5 are promising and one may wonder if the exact proportions are a must, which would make the proposed setting as expensive as the traditional one. To demonstrate that this is not the case and that our methodology only needs rough SP, we presented a sensitivity analysis regarding SP, where we added various amounts of noise to the extracted SP and demonstrated that the model performs satisfactorily well when trained with noisy SP. We also presented sensitivity analysis by investigating degraded SP by clustering to further support the robustness of our methodology when the precise SP is unavailable. The analysis suggests that our methodology not only works well with rough SP but also with rough SP for only some representative images from the whole training set, indicating its need for significantly less annotation effort.

Additional annotations. In many scenarios, different types of annotations may exist. This raises the question of whether it is feasible for semantic segmentation methods to use the combination of different types of annotations to boost their performance. In this regard, our proposed semantic segmentation methodology based on SP delivers quite promising results.

For datasets where the ground-truth segmentation maps are available, the SP annotation can be calculated directly. In these cases, an additional loss function using the SP scores can be used as demonstrated by the SPSS+ model we have proposed. The results shown in Tables 4.1 and 4.2 demonstrated the good performance of SPSS+. The

enhanced performance of our method by utilising both annotation types may benefit from our introduced loss function $\mathcal{L}_{\text{total}}$ in Eq. (4.3). It contains the $\mathcal{L}_{\text{sp}}$ loss defined in Eq. (4.1), which measures the MSE between the predicted SP and the given SP. The visualisation results in Figure 4.6 showed that our $\mathcal{L}_{\text{sp}}$ loss may produce better segmentation than the loss directly measuring the segmentation maps (that the benchmark method uses) in terms of the binarisation ability. Therefore, combining the $\mathcal{L}_{\text{sp}}$ loss with the $\mathcal{L}_{\text{sm}}$ loss and then forming the $\mathcal{L}_{\text{total}}$ loss could boost the semantic segmentation performance, e.g. see the visualisation given in Figure 4.7.

Limitations. SP provides much less information than standard segmentation annotations. In some scenarios, for example, with a large number of classes or where some classes represent only a tiny proportion of any image, the semantic proportions might not provide enough information for the network to infer the classes. Thus the utility of SP will be problem-dependent. In many ways, the surprising observation for us was to discover how powerful SP is on a range of problems given how little information we are providing to the network. Although SP will not be a solution for all segmentation problems, we believe that its relative cheapness means that it may be the method of choice in a number of applications where semantic segmentation is required, but the resources to hand annotation images are limited.

In this work, we proposed a new semantic segmentation methodology by introducing the SP annotation. In the scenario of quite limited annotation, using SP for semantic segmentation can already achieve competitive results. If additional annotations are available, our method can easily utilise them for a performance boost. Moreover, for existing segmentation methods that use different types of annotations, we also suggest involving SP in these methods; e.g., our proposed $\mathcal{L}_{\text{sp}}$ loss could serve as a type of regularisation given its effectiveness in binarisation.

4.7 Conclusion

Semantic segmentation methodologies generally require costly annotations such as the ground-truth segmentation masks in order to achieve satisfying performance. Motivated by reducing the annotation time and cost for semantic segmentation, we in this chapter presented a new methodology SPSS, relying on the SP annotation instead of the costly ground-truth segmentation maps. Extensive experiments validated the great potential of the proposed methodology in reducing the time and cost required for annotation, making it more feasible for large-scale applications. Furthermore, this innovative design opens up new opportunities for semantic segmentation tasks where obtaining the ground-truth segmentation maps may not be feasible or practical. We believe that

the use of the SP annotation suggested in this chapter offers a new and promising avenue for future research in the field of semantic segmentation, with evident and wide real-world applications.

In the context of XAI, we believe that these results suggest that semantic segmentations could plausibly be used as a more accurate way to represent saliency information. An obvious next step would be to use the saliency maps we generated in Chapter 3 to train a semantic segmentation network. This might then provide a much more detailed explanation of how a DNN interprets an image. Due to lack of time, this next step will have to be left as future work.

Chapter 5

Concept-Based Explainable Artificial Intelligence: Measures and Benchmarks

In this chapter, we return to our main theme of XAI. Here, we explore new methods to evaluate the performance of XAI and particular alignment between linguistic labels and features found by the DNNs.

Concept-based explanation methods, such as CBMs (concept bottleneck models) and CAVs (concept activation vectors), aim to improve the interpretability of ML models by linking their decisions to human-understandable concepts. These methods hold great promise but rely on the critical assumption that such concepts can be accurately attributed to the network's feature space. However, this foundational assumption has not been rigorously validated, mainly because the field lacks standardised measures and benchmarks to assess the existence and spatial alignment of such concepts. To address this, we propose three measures: the *CGIM (concept global importance measure)* to measure the global concept alignment, the *CEM (concept existence measure)* to test whether a concept identified by an examined methodology truly exists in the image, and the *CLM (concept location measure)* to evaluate whether the identified concept is spatially aligned with the corresponding human-understood region in a test image. To enable CLM, we also introduce *CoAM (concept activation mapping)*, a technique for visualising concept activations. We use the benchmark `Caltech-UCSB Bird (CUB)` [55] dataset and we benchmark post-hoc CBMs [37] on this dataset to illustrate their capabilities and challenges. Through qualitative and quantitative experiments, we demonstrate that, in many cases, even the most important concepts determined by post-hoc CBMs are not present in input images; moreover, when they are present, their saliency maps fail to align with the expected regions by either activating across an entire object or misidentifying relevant concept-specific regions. We analyse the root causes of these

limitations, such as the natural correlation of concepts. Our findings underscore the need for more careful application of concept-based explanation techniques especially in settings where spatial interpretability is critical.

5.1 Introduction

In recent years interest in XAI methods has grown substantially because of the desire to exploit the success of newly developed ML methods to new areas of our lives [16, 18, 142, 179, 180]. In an attempt to make XAI more understandable to the layman there has been a growing drive to develop techniques that provide explanations in terms of human-understandable concepts [36, 41, 49, 50, 98, 101]. One of the big challenges of concept-based XAI methods that is of paramount importance yet lacks of research is to ensure that the concepts identified as important to making a decision properly align with human understanding of the concepts.

In this chapter, we propose three new measures for measuring this alignment on a large dataset. The first is the *CGIM*, which measures the global concept alignment by the XAI techniques. The second is the *CEM*, which measures whether the concepts identified as important for making a classification exist in an image. For example, if the horn is identified as the most important concept for deciding the image is a rhinoceros then we should expect the horn to be visible in the image. The third measure is the *CLM*, which measures whether the excitable region of the feature maps used to determine an important concept is close to the location where we would expect the concept to be. In the example above, we would expect the heatmap representing the area of the feature map that corresponds to the horn concept should be located around the horn. Using these three measures we create a benchmark problem using the `CUB` dataset [55]. This is a rich dataset that provides 200 bird classes together with 112 binary concepts and the locations of many parts of the bird given for each image. Using this dataset and our new measures we can test the performance of concept-based XAI methods over the whole dataset of 11,800 images.

To illustrate the usefulness of our measures we examine a prominent example of a concept-based XAI system known as the post-hoc CBMs [37]. This method is designed to provide explanations of classifiers based on DNNs. The method is a synthesis of two approaches – traditional CBMs [41] and CAVs [36] – for concept-based explanations. Traditional CBMs are a relatively straightforward approach to introducing human-understandable concepts into XAI. In traditional CBMs, we start from a network trained to classify a set of classes and replace the final few layers with a new set of layers that are trained to predict human-understandable concepts, which provides a "concept bottleneck". From this concept representation, a fully connected layer is trained to predict the classes. Given a new image, it is then straightforward to see

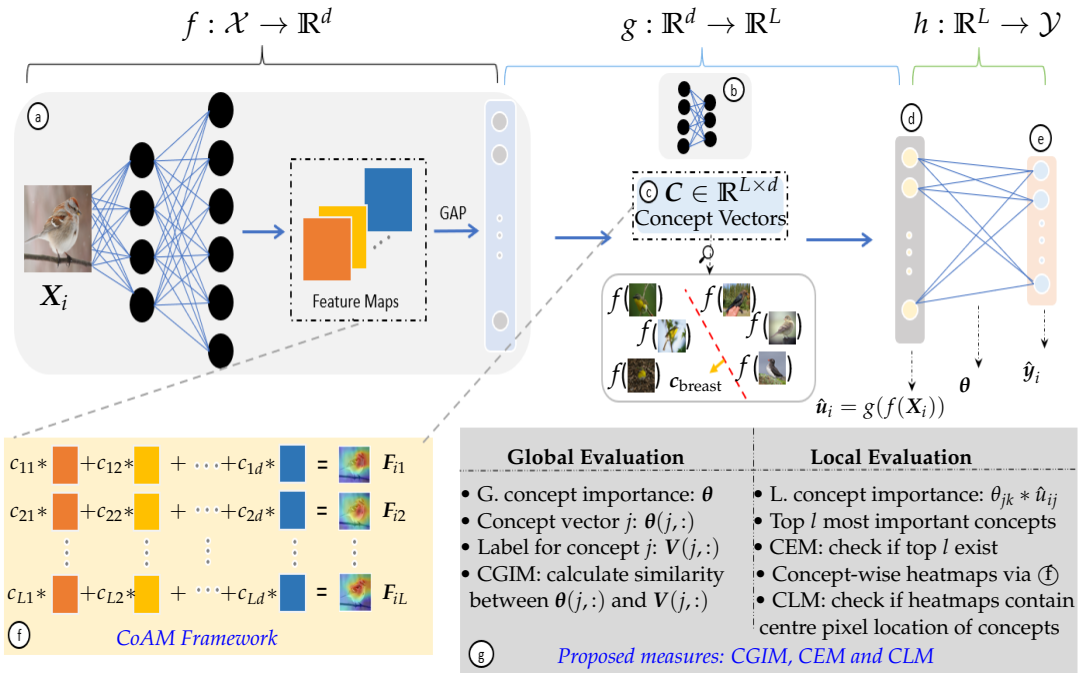


FIGURE 5.1: **Overview of CAVs, CBMs, post-hoc CBMs and the proposed techniques.** Feature extractor ⓐ, concept prediction block ⓑ, CAVs ⓒ, concept bottleneck ⓓ, classifier ⓔ, and our proposed CoAM framework ⓕ. A traditional (without concept bottleneck) classification model consists of ⓐ + ⓔ, and ⓒ is introduced post-hoc to explain its predictions via CAVs. ⓐ + ⓑ + ⓓ + ⓔ forms the steps for traditional CBMs training, whereas ⓐ + ⓒ + ⓓ + ⓔ forms the post-hoc CBMs. Our proposed CoAM framework is ⓕ, weighing pre-GAP feature maps with CAVs for concept visualisation. ⓖ presents the example steps of our proposed measures: CGIM, CEM and CLM.

which concepts are important in making the prediction [41]. The disadvantage of traditional CBMs is that in order to train the network it requires that every image is annotated with the set of concepts that are visible in the image. Although there exists a few datasets where such annotations are given, generally it would be prohibitively expensive to annotate a large dataset.

There has therefore been a drive to find cheaper methods to learn concepts. One example proposed by Aysel et al. [50] is to use annotations for the classes rather than individual images. A second family of models that were developed to provide concept-based explanations is known as CAV methods [36]. These methods take a pre-trained network and probe the internal representation to determine the directions in that representation that align with human-understandable concepts. One approach for doing this is to take two subsets of the images, one class where the concept is present and the other class where the concept is absent. From examining the difference in the representations between the two classes we can determine CAVs. This method is an example of a "post-hoc" XAI method as it seeks to explain the decisions of a pretrained network without changing that network.

The post-hoc CBMs [37] combine CAVs with the traditional CBMs. It takes a pre-trained network and feeds each channel in the last convolution layers into a GAP layer. It uses the GAP representation to learn a set of CAVs. To do so, for each concept, it chooses m positive and negative example images which it then trains an SVM (support vector machine) to separate (m is of the order of 100 images). Each SVM discriminant vector is taken as a CAV. These CAVs are then used to determine the degree to which a concept is present in an image. From this, a concept bottleneck can be trained. This is the post-hoc CBMs that we study in this chapter. The network is illustrated in the top row of Figure 5.1, and the bottom row illustrates the new measures that we propose to evaluate the alignment of the concepts with human understanding of the concepts.

Although post-hoc CBMs sacrifice some performance accuracy in predicting classes compared to traditional DNNs (i.e., the ones without a concept bottleneck), they provide a relatively cheap way to obtain human-interpretable concepts. However, for this explanation to be useful, the concepts need to be accurately aligned with human understanding of the concept. We use the new measures and new benchmarks for evaluating this alignment. As we will see the alignment is surprisingly poor, which highlights the necessity of introducing new measures for assessing this alignment. The main contributions of the paper are as follows.

- We propose the CoAM to visualise concept activations.
- We propose three quantitative measures: i) CGIM, to test the global concept alignment by XAI methods; ii) CEM, to test whether a concept being identified by XAI methods exists in the image; and iii) CLM, to test whether a concept being identified by XAI methods is spatially aligned with the human concept.
- We benchmark the post-hoc CBMs [37] using the proposed measures to evaluate the alignment of concept-based XAI techniques on a benchmark dataset and conduct insightful discussion.

The rest of the chapter is structured as follows. Section 5.2 recalls related work on concept-based explainability. Section 5.3 provides an overview of CBMs and CAVs as foundational methodologies. Section 5.4 introduces our proposed measures CGIM, CEM, and CLM, including our CoAM framework for concept activation visualisation. The thorough experimental results are presented in Section 5.5, followed by a detailed discussion of the findings in Section 5.6 and limitations. Finally, Section 5.7 concludes the chapter.

5.2 Related Work

This section recalls concept-based methodologies for XAI and examines existing variants of CAM highlighting the need for a dedicated approach to concept visualisation (which can be addressed by our CoAM).

Network dissection [49] is one of the well-known concept-based approaches, where individual neurons in a network are examined to identify their correspondence to human-understandable concepts like *edges, textures*, or *objects*. By aligning neuron activations with segmentation-annotated images, network dissection quantifies how well a model's internal representations map to meaningful concepts. However, this method is computationally expensive and data-intensive, requiring large and richly labelled datasets to accurately associate neurons with interpretable concepts. Despite its valuable insights, these limitations have prompted the development of more efficient and flexible methods, such as CAVs and CBMs.

Testing with CAVs (TCAV) framework [36] introduced CAVs to explain model predictions based on high-level human-interpretable concepts. CAVs represent directions in the latent space of a model corresponding to specific concepts, allowing for sensitivity analysis. By perturbing an input in the direction of a concept vector, TCAV measures how much the model's prediction depends on that specific concept, offering quantitative insights into the reliance on different concepts for a given task. TCAV has been applied in several fields to assess whether models depend on sensitive attributes like gender or race when making decisions. Recent adaptations have improved the computational efficiency and robustness of CAVs when applied to large-scale models [181]. However, TCAV can only unveil the global effect of concepts on examined classes and not on individual samples. Therefore, it is unable to directly assess the concept predictions or provide spatial concept localisation for individual images.

CBMs [41] offers a significantly different approach to interpretability. They enforce that intermediate representations of the model correspond to human-understandable concepts, such as attributes (e.g., *colour, shape, part*) of objects in an image. By constraining the model to predict based on these explicit concepts, CBMs inherently provide an interpretable mechanism for understanding decisions. This makes it easier to debug and correct errors by diagnosing the model's performance on individual concepts. Recent work in CBMs has focused on improving robustness, especially when concept labels are noisy or incomplete. For instance, in Label-free CBMs [182], a method was proposed using unsupervised techniques to learn concept bottlenecks, thereby extending the applicability of CBMs to scenarios where manual labelling is expensive or impractical. Despite their interpretability, CBMs typically lack the ability to provide spatial visualisations, limiting their usefulness in tasks that require precise localisation of important concepts.

The multilevel XAI method in [50] offers solutions for both expensive annotation needs and single-level output drawbacks of CBMs. The cost-effective solution to CBMs is achieved by only requiring class-wise concept annotations rather than per-image. Moreover, the multilevel XAI method provides concept-wise heatmaps by-product handling the single-level limitation of CBMs. To be more precise, different from other CBM approaches, the explanations by the multilevel XAI method are not only raw concept values but also each concept comes with its saliency map that highlights the region in the image activated by that concept. The authors in [50] have also shown the possibility of concept intervention on the input dimension, which is much more intuitive than the concept dimension. To give an example, in other CBMs, one may tweak the concept value, say, "*white*" at the bottleneck layer to flip the prediction, say, from polar bear to grizzly bear. In the multilevel XAI method, one can convert the white colour region in the image to brown to achieve the same flipping, which is more intuitive and reliable.

A breakthrough in visual explanations came with the introduction of CAM [38], which provides spatial localisation by computing class-specific activation maps that highlight the regions of an image most relevant for a given prediction. CAM operates by utilising the output of GAP layers in CNNs, enabling the generation of heatmaps that represent regions crucial for the final classification. This approach was generalised in Grad-CAM [39], which makes use of the gradients flowing into the final convolutional layer to visualise where the model "looks" when making a decision. Grad-CAM extends CAM to more general architectures without requiring specific layers like GAP. However, Grad-CAM does not always provide sharp localisation, especially when multiple objects are present in the image. Grad-CAM++ [88] addresses this limitation by refining the localisation to better handle multiple instances of objects, offering a more fine-grained interpretation. Further extensions include Score-CAM [75], which eliminates the dependency on gradients, instead using the activations themselves to weigh different regions of the input. This addresses some of the instability associated with gradient-based methods but comes with increased computational overhead. Other advancements like Ablation-CAM [183] explore removing parts of the model and input to measure their impact on predictions, thus improving interpretability.

5.3 Preliminary

In this section, we first present the notations used throughout the paper and then demonstrate how CAVs are generated as proposed by [36]. After that, we introduce the notations for traditional and post-hoc CBMs to familiarise the readers with these methodologies and their differences. Finally, the global and local concept importance notions are discussed.

Let $\mathcal{X}$ be the set of images, $\mathcal{U}$ be the set of concept labels, and $\mathcal{Y} = \{1, 2, \cdots, K\}$ be the set of K class labels. Let $\mathcal{S} = \{(\boldsymbol{X}_i, \boldsymbol{u}_i, \boldsymbol{y}_i, \Lambda_i, \mathcal{P}_i) \mid \boldsymbol{X}_i \in \mathcal{X}, \boldsymbol{u}_i \in \mathcal{U}, \boldsymbol{y}_i \in \mathcal{Y}, i = 1, 2, \ldots, N\}$ be the training set with N samples, where $\boldsymbol{u}_i \in \{0, 1\}^L$ is the concept label vector with L different concepts for image $\boldsymbol{X}_i \in \mathbb{R}^{M_1 \times M_2 \times M_3}$ ($M_3 = 3$ for RGB images), and $\boldsymbol{y}_i \in \mathbb{R}^K$ (a one-hot vector) denotes the class label of image $\boldsymbol{X}_i$, Λ_i is the set containing the indexes of activated concepts for image $\boldsymbol{X}_i$ (i.e., the indexes of the components in $\boldsymbol{u}_i$ with value 1), and $\mathcal{P}_i = \{p_{i1}, \cdots, p_{iL}\}$ is the set holding centre pixel coordinates p_{ij} of concept j with $j = 1, \ldots, L$ for image $\boldsymbol{X}_i$. Let $f : \mathcal{X} \to \mathbb{R}^d$ be a d-dimensional feature extractor, which can be any trained DNNs such as ResNet [60] and VGG [58]. From block ⓐ in Figure 5.1, we see that the feature vector $f(\boldsymbol{X}_i)$ consists of the post-GAP features (i.e., the features right after the GAP layer). Let $\boldsymbol{E}_i \in \mathbb{R}^{H \times W \times d}$ represent the pre-GAP feature maps (i.e., the features right before the GAP layer), where H, W and d denote the height, width and depth (i.e., the number of channels). The k-th channel of $\boldsymbol{E}_i$ is represented as $\boldsymbol{E}_i(:, :, k) \in \mathbb{R}^{H \times W}$.

CAVs. Following [36] and [37], for $j = 1, \ldots, L$, to generate the CAV $\boldsymbol{c}_j \in \mathbb{R}^d$ for the j-th concept, two sets of image embeddings through f are needed, i.e., $\mathcal{N}_j^{\text{pos}}$ for positive examples and $\mathcal{N}_j^{\text{neg}}$ for negative ones. In detail, set $\mathcal{N}_j^{\text{pos}}$ consists of embeddings of N_p images (positive examples) that contain the j-th concept, and set $\mathcal{N}_j^{\text{neg}}$ consists of embeddings of N_n randomly chosen images (negative examples) that do not contain the concept. Sets $\mathcal{N}_j^{\text{pos}}$ and $\mathcal{N}_j^{\text{neg}}$ are then used to train an SVM with $\boldsymbol{c}_j$ being the obtained normal vector to the hyperplane separating sets $\mathcal{N}_j^{\text{pos}}$ and $\mathcal{N}_j^{\text{neg}}$. All together, these L number of CAVs form a concept bank $\boldsymbol{C} = (\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L)^\top \in \mathbb{R}^{L \times d}$. For an image $\boldsymbol{X}_i$, the feature vector $f(\boldsymbol{X}_i)$ is to be projected onto the concept space by $\boldsymbol{C}$, i.e., $\boldsymbol{C}f(\boldsymbol{X}_i) \in \mathbb{R}^L$, which is the concept value vector $\hat{\boldsymbol{u}}_i$ to be fed to the classifier.

Traditional vs. post-hoc CBMs. After the feature vector $f(\boldsymbol{X}_i)$ is obtained for image $\boldsymbol{X}_i$, the traditional CBMs predict concepts by the concept prediction block, while the post-hoc CBMs project the feature vector $f(\boldsymbol{X}_i)$ onto the concept space using the concept bank $\boldsymbol{C}$, see Figure 5.1. Let

$$\hat{\boldsymbol{u}}_i = (\hat{u}_{i1}, \hat{u}_{i2}, \cdots, \hat{u}_{iL})^\top = g(f(\boldsymbol{X}_i)) \tag{5.1}$$

be the obtained concept vector for image $\boldsymbol{X}_i$ and g be the projection function. Then, for traditional CBMs, as the ground-truth concept label vector $\boldsymbol{u}_i$ for image $\boldsymbol{X}_i$ is available, g (i.e., the concept prediction block) is achieved/trained by minimising the below binary cross-entropy loss function

$$\mathcal{L}_g = \sum_i \mathcal{L}_g(\hat{\boldsymbol{u}}_i, \boldsymbol{u}_i). \tag{5.2}$$

In contrast, for post-hoc CBMs, where the ground-truth concept label vector $\boldsymbol{u}_i$ is not available, the obtained concept vector $\hat{\boldsymbol{u}}_i$ corresponding to $\boldsymbol{X}_i$ is directly obtained by setting

$$g(f(\boldsymbol{X}_i)) = \boldsymbol{C}f(\boldsymbol{X}_i). \tag{5.3}$$

Finally, the obtained concept vector $\hat{\boldsymbol{u}}_i$ is used to predict the final classes via a single classification layer

$$h : \mathbb{R}^L \rightarrow \mathcal{Y} \tag{5.4}$$

for both the traditional and post-hoc CBMs. In detail,

$$h(\hat{\boldsymbol{u}}_i) = \boldsymbol{\theta}^\top \hat{\boldsymbol{u}}_i + b, \tag{5.5}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{L \times K}$ holds the weights and b is the bias. Function h is trained for the final classification with the categorical cross-entropy loss function

$$\mathcal{L}_h = \sum_i \mathcal{L}_h(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_i), \tag{5.6}$$

where $\hat{\boldsymbol{y}}_i = h(g(f(\boldsymbol{X}_i)))$ is the class prediction of $\boldsymbol{X}_i$.

Global vs. local concept importance. After training, it is to use the trained model to make a prediction for every test image, and then rank the concepts and present the highest l of them as explanations. In this regard, it is crucial to differentiate between the global and local importance of concepts for a task as they may play key roles in different scenarios. Global importance is the overall effect of concepts for a given class. For instance, in the post-hoc CBMs setting, the classifier h is a single layer with weights $\boldsymbol{\theta}$ mapping concept values to the final classes (also see the right of Figure 5.1) and each parameter of this layer is proposed as the *global importance* of a concept that they weigh for an examined class. By analysing each parameter, say θ_{jk}, one can assess the overall effect of the concept j for class k. Moreover, tuning these parameters may allow the model to debug as presented in [37]. The *local importance* of concepts on the other hand is their influence on individual class predictions rather than on the entire class. The CBM and its variants focus on *local concept interventions* [36], which is the process of tweaking the predicted/projected concept values in $\hat{\boldsymbol{u}}_i$ at the concept bottleneck layer, i.e., ⓓ in Figure 5.1, to flip a single class prediction when needed. An effective way to determine what concept values to intervene on is an active area of research [97, 98, 100].

One, however, should note that the magnitude of the concept values at the bottleneck layer is not the same as the *local importance*. This is because a class prediction score by h is the $\boldsymbol{\theta}$-weighted sum of concept values, and the parameters of $\boldsymbol{\theta}$ may greatly increase or decrease the individual concept effects on the final classification. Therefore, defining the concept importance solely based on their values in $\hat{\boldsymbol{u}}_i$ is misleading. We will address this issue in our proposed methodology in Section 5.4.

5.4 Proposed Methodology

There is a significant gap in the field regarding the evaluation of the explainability power of the well-known concept-based methodologies. To fill this gap and assess the existence and correctness of the concepts given as highly important by XAI techniques, we propose our CoAM framework (see ⓕ in Figure 5.1 for an overview), which allows concept visualisation. Moreover, we also propose the CGIM to test the global concept alignment by XAI methods, the CEM to evaluate the existence of the concepts, and the CLM to reveal whether the highly important concepts correspond to the correct regions in a given test image.

5.4.1 Concept Activation Mapping

We below propose the CoAM framework, which generates concept activation maps revealing the parts of an image that correspond to these concepts. As we know, for post-hoc CBMs, the pre-GAP feature maps $\boldsymbol{E}_i \in \mathbb{R}^{H\times W\times d}$ (which contain spatial information) for the examined image $\boldsymbol{X}_i$ become the post-GAP feature vector $f(\boldsymbol{X}_i)$ after the GAP layer, which is then linked to the CAVs, $\boldsymbol{c}_j = (c_{j1}, \cdots, c_{jd})^\top, j = 1, \ldots, L$.

Our introduced concept activation maps, say $\boldsymbol{F}_{ij}$, for $\boldsymbol{X}_i$ corresponding to the j-th concept for $j = 1, \ldots, L$, are calculated by

$$\boldsymbol{F}_{ij} = \frac{1}{d}\sum_{k=1}^{d} c_{jk}\boldsymbol{E}_i(:,:,k) \in \mathbb{R}^{H\times W}, \tag{5.7}$$

i.e., weighing the pre-GAP feature maps of $\boldsymbol{X}_i$ by the j-th CAV; see block © in Figure 5.1. The CoAM framework is also summarised in Algorithm 4, with $\boldsymbol{F}_i$ being the output, where $\boldsymbol{F}_i(:,:,j) = \boldsymbol{F}_{ij}$ for $j = 1, 2, \ldots, L$. Since the size of each $\boldsymbol{F}_{ij}$ is significantly smaller than that of $\boldsymbol{X}_i$, to visualise the concept activation maps in a better way and for localisation assessment, we upsample them to the original image size of $\boldsymbol{X}_i$, denoted by $\bar{\boldsymbol{F}}_{ij}$, and overlay them on $\boldsymbol{X}_i$. This will tell us what parts of the input image contribute to the individual concepts. Algorithm 5 gives the details of the final feature visualisation pseudo-code.

5.4.2 Concept Global Importance Measure

We firstly introduce the *global importance score* of concept j for class k as θ_{jk} [the (j,k)-th entry of $\boldsymbol{\theta}$], i.e., the weight in the classifier h mapping the j-th concept to the k-th class, for $j = 1, 2, \ldots, L$ and $k = 1, 2, \ldots, K$. Let $\boldsymbol{V} \in \mathbb{R}^{L\times K}$ be the ground-truth concept matrix for all the classes provided by annotators, where the entry of its j-th row and k-th column V_{jk} is the ground-truth value of the j-th concept for the k-th class. One

might consider directly comparing θ_{jk} and V_{jk} for global evaluation of the correctness of θ_{jk}. However, this is inappropriate because these values are on different scales; in particular, $\boldsymbol{V}$ contains values between 0 and 1, while $\boldsymbol{\theta}$ can take any real value as it represents layer weights. To address this issue, we propose to compare the entire j-th row vectors $\boldsymbol{\theta}(j,:)$ and $\boldsymbol{V}(j,:)$ by calculating their similarity for $j = 1, 2, \ldots, L$.

Our *first type* CGIM is defined as

$$\rho_j^{\text{CGIM}_1} := \phi(\boldsymbol{\theta}(j,:), \boldsymbol{V}(j,:)), \;\; j = 1, 2, \ldots, L, \tag{5.8}$$

where ϕ is the function for similarity calculation. In this paper, we use the cosine similarity (measuring the alignment between two vectors regardless of their magnitudes) for ϕ. Therefore, $\rho_j^{\text{CGIM}_1}$ is a similarity score between -1 and 1 for the j-th concept. Ideally, $\rho_j^{\text{CGIM}_1}$ is expected to be close to 1 if the obtained $\boldsymbol{\theta}(j,:)$ is meaningful.

Analogous to the first type CGIM in Eqn (5.8), we also introduce the concept global explanations based on the average say $\hat{\boldsymbol{u}}_k^*$ of the concept vectors $\hat{\boldsymbol{u}}_i$ of $\forall \boldsymbol{X}_i \in \mathcal{X}_{\text{T}}^k$, where $\mathcal{X}_{\text{T}}^k$ is the set that consists of all the test images with correct predicted class $1 \leq k \leq K$ and $|\mathcal{X}_{\text{T}}^k| = N_k$. Then, form $\hat{\boldsymbol{U}}^* = (\hat{\boldsymbol{u}}_1^*, \hat{\boldsymbol{u}}_2^*, \cdots, \hat{\boldsymbol{u}}_K^*) \in \mathbb{R}^{L \times K}$, i.e., the obtained average concept matrix. Our *second type* CGIM is then defined as

$$\rho_j^{\text{CGIM}_2} := \phi(\hat{\boldsymbol{U}}^*(j,:), \boldsymbol{V}(j,:)), \;\; j = 1, 2, \ldots, L. \tag{5.9}$$

If we consider both the weight matrix $\boldsymbol{\theta}$ and the obtained average concept matrix $\hat{\boldsymbol{U}}^*$, we have our *third type* CGIM, which is defined as

$$\rho_j^{\text{CGIM}_3} := \phi(\hat{\boldsymbol{U}}_{\boldsymbol{\theta}}^*(j,:), \boldsymbol{V}(j,:)), \;\; j = 1, 2, \ldots, L, \tag{5.10}$$

where $\hat{\boldsymbol{U}}_{\boldsymbol{\theta}}^* = \boldsymbol{\theta} \odot \hat{\boldsymbol{U}}^*$ with $\odot$ being the pointwise multiplication operator.

The above proposed CGIM scores $\rho_j^{\text{CGIM}_1}$, $\rho_j^{\text{CGIM}_2}$, and $\rho_j^{\text{CGIM}_3}$ are for each concept $1 \leq j \leq L$. They can also be readily modified analogously so that we can calculate CGIM scores for each class $1 \leq k \leq K$, i.e.,

$$\rho_k^{\text{CGIM}_1} := \phi(\boldsymbol{\theta}(:,k), \boldsymbol{V}(:,k)), \tag{5.11}$$

$$\rho_k^{\text{CGIM}_2} := \phi(\hat{\boldsymbol{U}}^*(:,k), \boldsymbol{V}(:,k)), \tag{5.12}$$

$$\rho_k^{\text{CGIM}_3} := \phi(\hat{\boldsymbol{U}}_{\boldsymbol{\theta}}^*(:,k), \boldsymbol{V}(:,k)). \tag{5.13}$$

5.4.3 Concept Existence Measure

We now define the *local importance score* of concept j for class k as $\theta_{jk}\hat{u}_{ij}$; note that $\hat{u}_{ij}$ is the obtained j-th concept value of test image $\boldsymbol{X}_i$ and θ_{jk} is the weight in classifier h linking the j-th concept and the k-th class prediction. We rank the total L concepts for

Algorithm 4 Concept Activation Mapping (CoAM)

Input:

- Pre-GAP feature maps $\boldsymbol{E}_i \in \mathbb{R}^{H\times W\times d}$ for image $\boldsymbol{X}_i$
- Concept bank $\boldsymbol{C} \in \mathbb{R}^{L\times d}$

Output: Concept activation map $\boldsymbol{F}_i \in \mathbb{R}^{H\times W\times L}$

1: **for** each concept j in the $\boldsymbol{C}$ **do**
2: Compute the weighted map $\boldsymbol{F}_{ij}$ with Eqn (5.7)
3: Set $\boldsymbol{F}_i(:,:,j) = \boldsymbol{F}_{ij}$
4: **end for**
5: **return** $\boldsymbol{F}_i$

test image $\boldsymbol{X}_i$ based on their contribution to the final classification prediction k using the local importance score $\theta_{jk}\hat{u}_{ij}$, and let $\boldsymbol{q}_i = (q_{i1}, q_{i2}, \cdots, q_{iL})^\top$ represent the ranked indexes of the concepts for $\boldsymbol{X}_i$. Therefore, if $q_{is} = m$, it means the m-concept is ranked at the s place for $s = 1, 2, \ldots, L$ based on the descending order of the magnitude of $\theta_{mk}\hat{u}_{im}$ among $\{\theta_{jk}\hat{u}_{ij}\}_{j=1}^{L}$.

Recall that Λ_i is the set containing the indexes of activated concepts for image $\boldsymbol{X}_i$. Our CEM is defined as

$$\rho_l^{\text{CEM}} := \frac{1}{l}\sum_{j=1}^{l} \mathbf{1}_{\Lambda_i}(q_{ij}), \tag{5.14}$$

assessing if the first $l \leq L$ concepts (i.e., the first l components) in $\boldsymbol{q}_i$ exist in the examined image $\boldsymbol{X}_i$, where $\mathbf{1}_{\Lambda_i}$ is an indicator function defined as

$$\mathbf{1}_{\Lambda_i}(x) = \begin{cases} 1, & \text{if } x \in \Lambda_i; \\ 0, & \text{otherwise.} \end{cases} \tag{5.15}$$

Obviously, the CEM ρ_l^{CEM} is an accuracy score between 0 and 1 evaluating the existence of highly important concepts in image $\boldsymbol{X_i}$, thanks to the set Λ_i containing the indexes of activated concepts. CEM reveals the reliability of explanations generated by a trained model; for example, $\rho_l^{\text{CEM}} = 0$ means none of the l highly important concepts exists in the examined image, whereas $\rho_l^{\text{CEM}} = 1$ means all of the l highly important concepts exist in the examined image. We remark that ρ_l^{CEM} can also be obtained in the same manner by using θ_{jk} or $\hat{u}_{ij}$ instead of $\theta_{jk}\hat{u}_{ij}$ as the local importance score for comparison purpose.

5.4.4 Concept Location Measure

After checking whether the obtained important concepts of image $\boldsymbol{X_i}$ exist in the ground-truth set Λ_i with CEM and generating concept activation maps with CoAM, we now

Algorithm 5 Feature Visualisation in CoAM

Input:

- Boolean flag *colored* for generating colored heatmaps.
- Threshold value *threshold* for binary heatmaps.
- Opacity level β for superimposed heatmaps.
- Input image $\boldsymbol{X}_i \in \mathbb{R}^{M_1 \times M_2 \times M_3}$.
- Concept activation map $\boldsymbol{F}_i \in \mathbb{R}^{H \times W \times L}$ of $\boldsymbol{X}_i$. ▷ L is the number of concepts

Output: Set of superimposed images $\bar{\mathcal{S}} \in \mathbb{R}^{M_1 \times M_2 \times M_3 \times L}$.

Initialize an empty list of superimposed images $\bar{\mathcal{S}}$
for each spatial projection map j in $\boldsymbol{F}_i$ **do**
    heatmap = `resize` ($\boldsymbol{F}_{ij}$, (M_1, M_2)) ▷ Generate heatmap with size of (M_1, M_2)
    if *colored* **then** ▷ Apply a colormap to the heatmap
        jet_heatmap = `apply_colormap` (heatmap, "jet") ▷ Convert the heatmap to an RGB image
        superimposed_img = $\beta \cdot$ jet_heatmap + $\boldsymbol{X}_i$ ▷ Overlay heatmap on the original image $\boldsymbol{X}_i$
        Append superimposed_img to $\bar{\mathcal{S}}$
    else ▷ Generate binary heatmap using the threshold value
        binary_heatmap = `binary_threshold` (heatmap, *threshold*)
        superimposed_img = $\boldsymbol{X}_i \odot$ binary_heatmap ▷ Overlay heatmap on the original image $\boldsymbol{X}_i$
        Append superimposed_img to $\bar{\mathcal{S}}$
    end if
end for
return $\bar{\mathcal{S}}$

propose CLM to assess whether the obtained concepts of image $\boldsymbol{X_i}$ correspond to the correct region in $\boldsymbol{X_i}$.

Note that this check could be rigorously done by calculating the IoU (intersection over union) score if a ground-truth segmentation map per concept is available. However, the absence of these ground-truth maps makes this way impractical. In contrast, it will be much easier to mark some pixels, e.g. the coordinate information of the centre pixel for each important semantic area in an image, and then link the coordinate information to each concept. One useful label available for this purpose is the coordinate information of the centre pixel for each concept, i.e., $\mathcal{P}_i$, for image $\boldsymbol{X}_i$.

The proposed CLM checks whether the concept-wise activation heatmap $\bar{\boldsymbol{F}}_{ij}$ for concept j generated by CoAM contains the ground-truth centre location p_{ij}. For the $l \leq L$ most important concepts of $\boldsymbol{X}_i$ obtained in $\boldsymbol{q}_i$, our CLM is defined as

$$\rho_l^{\text{CLM}} := \frac{1}{l} \sum_{j=1}^{l} \mathbf{1}_{\Omega_{ij}}(p_{ij}), \tag{5.16}$$

where Ω_{ij} is the visual region of concept j of $\boldsymbol{X}_i$. Obviously, ρ_l^{CLM} is an accuracy score between 0 and 1 evaluating the alignment between the obtained individual concept heatmaps and their actual region in the image $\boldsymbol{X}_i$. In particular, $\rho_l^{\text{CLM}} = 0$ means none of the l highly important concepts corresponds to the correct region in the image, whereas a $\rho_l^{\text{CLM}} = 1$ score means all the l highly important concepts correspond to the correct region in the image. Finally, we remark that there are many ways to obtain the visual region Ω_{ij}. In this paper, we use thresholding on the concept-wise activation heatmap $\bar{\boldsymbol{F}}_{ij}$ with threshold τ to obtain Ω_{ij}.

5.5 Experiments

We in this section benchmark results and evaluate the performance of the post-hoc CBMs using our proposed measures. The benchmark fine-grained bird classification dataset, `CUB` [55], with concept annotations such as *wing colour*, *beak shape* and *feather pattern* is employed for the experiments. It consists of 200 different classes and 112 binary concept labels for around 11,800 images. Additionally, the central pixel locations of 12 different body parts are provided and used for concept localisation assessment by the proposed CLM. Following [37], we employ a ResNet-18 [60] trained on the `CUB` dataset[1] as the feature extractor f. CAVs are calculated as explained in Section 5.4 to create a concept bank $\boldsymbol{C}$ (also see © in Figure 5.1). Finally, a single layer h with weights $\boldsymbol{\theta} \in \mathbb{R}^{112\times 200}$ is trained for the classification.

5.5.1 Post-hoc CBMs Reproduction

By employing the same model as the feature extractor and following the same steps for CAVs and classifier training, we reproduce the results of post-hoc CBMs [37] with various hyperparameter combinations. There are two hyperparameters to tune during the SVM training for CAV learning, i.e., N_p and N_n (the number of positive and negative images per concept), which we set to 50 and 100, respectively. The other hyperparameter is the regularisation term λ in SVM, which controls the trade-off between maximising the margin that separates classes and minimising classification errors on the training data. A low λ value allows the model to prioritise a wider margin, even if some data points are misclassified, making the model more robust to noise and potentially improving its generalisation of new data. In contrast, a high λ value forces the SVM to minimise the training error, making it less tolerant of misclassifications and resulting in a narrower margin. While a high λ can lead to more accurate training performance, it may also increase the risk of overfitting, as the model becomes more sensitive to individual data points. Thus, λ helps balance the SVM's complexity and flexibility, impacting its ability to generalise well.

[1] The trained CUB model is available at https://github.com/osmr/imgclsmob.

TABLE 5.1: Classification accuracy of the reproduced post-hoc CBMs with different settings for the parameters λ, N_p, and N_n.

λ \ $N_p = N_n$	50	100
0.001	26.7	52.2
0.01	34.1	44.9
0.1	29.1	41.5
1	25.5	**59.1**
10	25.3	58.7
Traditional model w/o bottleneck	75.4	

We train SVM with λ values ranging from 0.001 to 10. Table 5.1 shows the classification accuracy of the classifier h with various concept banks obtained by these hyperparameter combinations. For the experiments in the rest of the paper, we employ the model with the best classification accuracy 59.1%, which is achieved when $N_p = N_n = 100$ and $\lambda = 1$. This result is very close to the accuracy 58.8% reported in the seminal work [37]. Note that there is more than 15% accuracy loss in comparison to the traditional model, i.e., the one without a concept bottleneck (i.e., ⓐ + ⓔ in Figure 5.1), for the sake of obtaining an interpretable model via concept bottleneck.

5.5.2 Global Importance Evaluation

We now investigate the quality of the global explanations of the post-hoc CBMs. Recall that the entries of $\boldsymbol{\theta} \in \mathbb{R}^{112\times 200}$ are considered as the global importance scores, determining the importance of a concept for an examined class. Ideally, these weights should closely align with human annotations in $\boldsymbol{V} \in \mathbb{R}^{112\times 200}$, i.e., the so-called ground truth. Intuitively, we expect the CGIM scores $\rho_j^{\text{CGIM}_1}$, $\rho_j^{\text{CGIM}_2}$, and $\rho_j^{\text{CGIM}_3}$ of $\boldsymbol{\theta}$, $\hat{\boldsymbol{U}}^*$, and $\hat{\boldsymbol{U}}_{\boldsymbol{\theta}}^*$ corresponding to $\boldsymbol{V}$ for each concept $1 \leq j \leq 112$ (and analogously for each class $1 \leq k \leq 200$) to be close to 1 if the obtained $\boldsymbol{\theta}$, $\hat{\boldsymbol{U}}^*$, and $\hat{\boldsymbol{U}}_{\boldsymbol{\theta}}^*$ are meaningful.

The calculated CGIM scores of the post-hoc CBMs for each concept $1 \leq j \leq 112$ and for each class $1 \leq k \leq 200$ are respectively presented in Table 5.2 and Tables 5.3 and 5.4. To better visualise and interpret results, Figure 5.2 showcases the histograms of the obtained CGIM scores across a range between -1 (maximum dissimilarity) and 1 (maximum similarity), regarding individual classes and concepts. Again, in an ideal scenario, it would be expected the CGIM scores $\rho_k^{\text{CGIM}_1}$ and $\rho_j^{\text{CGIM}_1}$ in the top row of Figure 5.2 to be a single prominent bar at the value of 1, or at the very least, a clear accumulation of bars towards the right end of the histogram (approaching 1), if the results of the post-hoc CBMs are meaningful/correct. Obviously, this is not the case. For example, the class and concept histograms in Figure 5.2 (a)–(b) show that many bars are distributed across the range from -1 to 1 with a noticeable number of values on the negative side, indicating a tendency towards negative correlation for some classes and

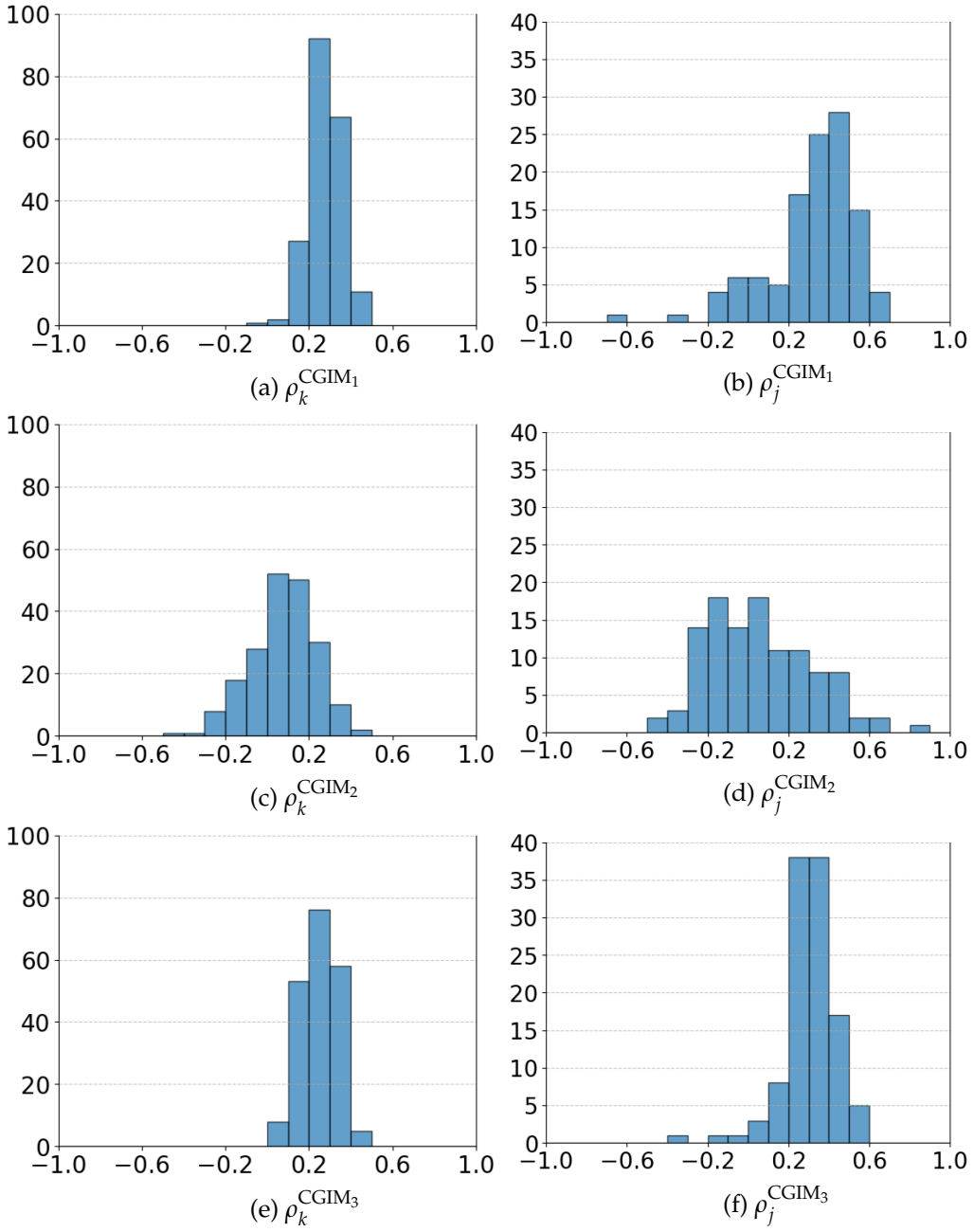


FIGURE 5.2: Histograms of the CGIM scores of the post-hoc CBMs. Plots on the left and right columns show the results for classes and concepts, respectively. A full list of the CGIM scores can be found in Table 5.2 for the concepts and in Tables 5.3 and 5.4 for the classes.

concepts, which is contrary to the expected accumulation near 1. The class and concept histograms in terms of $\rho_k^{\text{CGIM}_2}$ and $\rho_j^{\text{CGIM}_2}$ in Figure 5.2 (c)–(d) and $\rho_j^{\text{CGIM}_3}$ and $\rho_k^{\text{CGIM}_3}$ in Figure 5.2 (e)–(f) again disclose the same issue of the post-hoc CBMs.

A deeper analysis is also conducted by investigating the specific concepts and classes with significantly low or negative CGIM scores presented in Tables 5.2, 5.3 and 5.4. For instance, the $\rho_j^{\text{CGIM}_1}$ score for concept ($j = 51$) *black eye colour* in Table 5.2 is a

TABLE 5.2: Full list of CGIM scores for concepts in CUB dataset [55] with reproduced post-hoc CBMs [37].

Concept \ CGIM	$\rho_j^{\text{CGIM}_1}$	$\rho_j^{\text{CGIM}_2}$	$\rho_j^{\text{CGIM}_3}$	Concept \ CGIM	$\rho_j^{\text{CGIM}_1}$	$\rho_j^{\text{CGIM}_2}$	$\rho_j^{\text{CGIM}_3}$
1: Dagger beak	0.54	0.05	0.41	57: Yellow forehead colour	0.41	0.22	0.47
2: Hooked seabird beak	0.45	0.08	0.34	58: Black forehead colour	0.39	0.25	0.35
3: All-purpose beak	0.37	0.67	0.46	59: White forehead colour	0.52	−0.13	0.07
4: Cone beak	0.47	0.23	0.43	60: Brown under tail colour	−0.01	−0.16	0.23
5: Brown wing colour	0.22	−0.05	0.28	61: Grey under tail colour	0.36	−0.21	0.30
6: Grey wing colour	−0.13	−0.25	0.31	62: Black under tail colour	−0.11	0.35	0.24
7: Yellow wing colour	0.42	−0.04	0.42	63: White under tail colour	0.02	−0.22	0.23
8: Black wing colour	0.42	−0.02	0.33	64: Buff under tail colour	0.46	−0.21	0.22
9: White wing colour	−0.05	0.08	0.30	65: Brown nape colour	0.22	0.00	0.39
10: Buff wing colour	0.22	0.12	0.30	66: Grey nape colour	0.27	0.34	0.38
11: Brown upper-part colour	0.09	0.03	0.26	67: Yellow nape colour	0.35	0.23	0.52
12: Grey upper-part colour	−0.36	0.16	0.23	68: Black nape colour	0.37	0.00	0.20
13: Yellow upper-part colour	0.46	0.09	0.45	69: White nape colour	0.38	0.16	0.28
14: Black upper-part colour	−0.04	0.48	0.30	70: Buff nape colour	0.49	−0.24	0.14
15: White upper-part colour	0.32	−0.17	0.23	71: Brown belly colour	0.56	−0.15	0.22
16: Buff upper-part colour	0.27	−0.13	0.20	72: Grey belly colour	0.52	−0.12	0.41
17: Brown underpart colour	0.47	0.02	0.36	73: Yellow belly colour	0.20	0.43	0.27
18: Grey underpart colour	0.19	0.04	0.37	74: Black belly colour	0.48	0.21	0.46
19: Yellow underpart colour	0.40	0.55	0.51	75: White belly colour	0.01	0.33	0.21
20: Black underpart colour	0.55	0.05	0.29	76: Buff belly colour	0.48	−0.31	−0.07
21: White underpart colour	−0.04	0.03	0.26	77: Rounded wing shape	0.05	0.03	0.38
22: Buff underpart colour	0.18	−0.10	0.20	78: Pointed wing shape	0.51	−0.16	0.34
23: Solid breast pattern	0.38	0.39	0.40	79: Small size	−0.09	0.37	0.28
24: Striped breast pattern	0.33	0.00	0.33	80: Medium size	0.23	0.06	0.29
25: Multi-coloured breast pattern	0.51	−0.19	0.25	81: Very small size	0.57	−0.21	0.26
26: Brown back colour	0.43	0.04	0.23	82: Duck-like shape	0.42	0.41	0.57
27: Grey back colour	0.38	−0.21	0.18	83: Perching-like shape	−0.18	0.47	0.11
28: Yellow back colour	0.31	0.03	0.32	84: Solid back pattern	0.53	−0.13	0.20
29: Black back colour	0.22	0.60	0.38	85: Striped back pattern	0.33	−0.09	0.33
30: White back colour	0.34	−0.23	0.20	86: Multi-coloured back pattern	0.25	−0.38	0.39
31: Buff back colour	0.01	−0.24	0.27	87: Solid tail pattern	0.64	0.41	0.45
32: Notched tail shape	0.21	−0.10	0.31	88: Striped tail pattern	0.42	−0.30	0.21
33: Brown upper tail colour	0.33	−0.10	0.23	89: Multi-coloured tail pattern	0.25	−0.41	0.21
34: Grey upper tail colour	0.08	−0.19	0.23	90: Solid belly pattern	0.35	0.47	0.37
35: Black upper tail colour	0.44	0.50	0.50	91: Brown primary colour	0.12	0.19	0.26
36: White upper tail colour	−0.13	0.19	0.25	92: Grey primary colour	0.45	0.00	0.27
37: Buff upper tail colour	0.51	−0.01	0.39	93: Yellow primary colour	0.12	0.44	0.31
38: Head pattern eyebrow	0.41	−0.12	0.37	94: Black primary colour	0.47	0.11	0.39
39: Head pattern plain	0.67	−0.05	0.34	95: White primary colour	0.58	−0.05	0.26
40: Brown breast colour	0.39	0.19	0.23	96: Buff primary colour	0.32	−0.29	0.26
41: Grey breast colour	0.58	−0.05	0.35	97: Grey leg colour	0.55	0.22	0.43
42: Yellow breast colour	0.47	0.26	0.46	98: Black leg colour	0.43	−0.09	0.29
43: Black breast colour	0.22	0.00	0.25	99: Buff leg colour	0.17	0.00	0.27
44: White breast colour	0.32	−0.19	0.30	100: Grey bill colour	0.43	0.06	0.41
45: Buff breast colour	0.32	−0.26	0.17	101: Black bill colour	0.36	0.39	0.38
46: Grey throat colour	0.38	−0.11	0.37	102: Buff bill colour	0.52	−0.43	−0.13
47: Yellow throat colour	0.26	0.23	0.29	103: Blue crown colour	0.37	0.27	0.50
48: Black throat colour	0.52	−0.03	0.28	104: Brown crown colour	0.39	0.16	0.32
49: White throat colour	0.45	0.14	0.32	105: Grey crown colour	0.39	−0.24	0.19
50: Buff throat colour	0.42	−0.29	0.22	106: Yellow crown colour	0.25	0.20	0.36
51: Black eye colour	−0.63	0.82	−0.33	107: Black crown colour	0.45	0.18	0.39
52: Head size beak	0.40	0.26	0.38	108: White crown colour	0.42	−0.13	0.01
53: Shorten than head size beak	−0.08	0.43	0.23	109: Solid wing pattern	0.61	0.39	0.60
54: Blue forehead colour	0.26	0.36	0.50	110: Spotted wing pattern	0.48	0.04	0.49
55: Brown forehead colour	0.43	−0.20	0.32	111: Striped wing pattern	0.24	−0.12	0.37
56: Grey forehead colour	0.62	−0.20	0.06	112: Multi-coloured wing pattern	0.27	0.17	0.40

large negative value, i.e., -0.63. Similarly, the $\rho_j^{\text{CGIM}_1}$ score close to 0 for class ($k = 18$)

TABLE 5.3: Full list of CGIM scores for classes in CUB dataset [55] with reproduced post-hoc CBMs [37].

Class \ CGIM	$\rho_k^{\text{CGIM}_1}$	$\rho_k^{\text{CGIM}_2}$	$\rho_k^{\text{CGIM}_3}$	Class \ CGIM	$\rho_k^{\text{CGIM}_1}$	$\rho_k^{\text{CGIM}_2}$	$\rho_k^{\text{CGIM}_3}$
1: Black footed Albatross	0.24	−0.20	0.03	51: Horned Grebe	0.27	−0.24	0.24
2: Laysan Albatross	0.30	0.04	0.11	52: Pied billed Grebe	0.32	−0.12	0.32
3: Sooty Albatross	0.28	−0.15	0.17	53: Western Grebe	0.25	−0.04	0.24
4: Groove billed Ani	0.28	0.37	0.31	54: Blue Grosbeak	0.20	0.20	0.31
5: Crested Auklet	0.25	0.18	0.26	55: Evening Grosbeak	0.40	0.24	0.35
6: Least Auklet	0.26	0.07	0.14	56: Pine Grosbeak	0.14	−0.01	0.20
7: Parakeet Auklet	0.25	0.12	0.20	57: Rose breasted Grosbeak	0.31	0.17	0.31
8: Rhinoceros Auklet	0.36	0.00	0.22	58: Pigeon Guillemot	0.36	0.05	0.20
9: Brewer Blackbird	0.29	0.21	0.28	59: California Gull	0.27	0.16	0.24
10: Red winged Blackbird	0.27	0.36	0.38	60: Glaucous winged Gull	0.37	0.04	0.26
11: Rusty Blackbird	0.17	−0.09	0.32	61: Heermann Gull	0.26	0.12	0.32
12: Yellow headed Blackbird	0.36	0.32	0.41	62: Herring Gull	0.27	0.01	0.17
13: Bobolink	0.27	0.23	0.34	63: Ivory Gull	0.33	0.33	0.30
14: Indigo Bunting	0.15	0.16	0.23	64: Ring billed Gull	0.34	0.20	0.29
15: Lazuli Bunting	0.16	−0.07	0.27	65: Slaty backed Gull	0.25	−0.04	0.23
16: Painted Bunting	0.28	0.01	0.36	66: Western Gull	0.16	0.15	0.20
17: Cardinal	0.22	0.00	0.17	67: Anna Hummingbird	0.16	−0.15	0.17
18: Spotted Catbird	−0.02	−0.38	0.14	68: Ruby throated Hummingbird	0.28	−0.22	0.17
19: Gray Catbird	0.27	0.17	0.28	69: Rufous Hummingbird	0.19	−0.09	0.16
20: Yellow breasted Chat	0.39	0.01	0.33	70: Green Violetear	0.20	−0.14	0.12
21: Eastern Towhee	0.31	0.00	0.33	71: Long tailed Jaeger	0.24	−0.16	0.13
22: Chuck will Widow	0.23	0.05	0.32	72: Pomarine Jaeger	0.18	−0.28	0.36
23: Brandt Cormorant	0.27	0.02	0.16	73: Blue Jay	0.30	0.02	0.36
24: Red faced Cormorant	0.22	0.15	0.24	74: Florida Jay	0.32	−0.06	0.24
25: Pelagic Cormorant	0.05	0.06	0.16	75: Green Jay	0.32	0.13	0.38
26: Bronzed Cowbird	0.22	0.19	0.32	76: Dark eyed Junco	0.22	−0.05	0.20
27: Shiny Cowbird	0.33	0.25	0.24	77: Tropical Kingbird	0.28	0.13	0.36
28: Brown Creeper	0.30	0.11	0.27	78: Gray Kingbird	0.17	0.06	0.29
29: American Crow	0.28	0.38	0.28	79: Belted Kingfisher	0.25	0.01	0.19
30: Fish Crow	0.29	0.44	0.33	80: Green Kingfisher	0.27	−0.04	0.23
31: Black billed Cuckoo	0.26	0.04	0.21	81: Pied Kingfisher	0.14	0.41	0.19
32: Mangrove Cuckoo	0.23	−0.22	0.17	82: Ringed Kingfisher	0.21	−0.18	0.17
33: Yellow billed Cuckoo	0.26	−0.02	0.29	83: White breasted Kingfisher	0.39	0.13	0.28
34: Gray-crowned Rosy Finch	0.33	0.14	0.21	84: Red legged Kittiwake	0.30	0.17	0.25
35: Purple Finch	0.05	0.00	0.16	85: Horned Lark	0.31	0.13	0.23
36: Northern Flicker	0.35	0.02	0.29	86: Pacific Loon	0.48	−0.13	0.17
37: Acadian Flycatcher	0.32	−0.15	0.33	87: Mallard	0.27	−0.14	0.23
38: Great Crested Flycatcher	0.37	0.00	0.30	88: Western Meadowlark	0.43	0.05	0.38
39: Least Flycatcher	0.14	−0.15	0.16	89: Hooded Merganser	0.46	−0.03	0.38
40: Olive sided Flycatcher	0.21	−0.20	0.25	90: Red breasted Merganser	0.24	−0.10	0.29
41: Scissor tailed Flycatcher	0.25	−0.01	0.25	91: Mockingbird	0.22	−0.03	0.10
42: Vermilion Flycatcher	0.12	−0.01	0.07	92: Nighthawk	0.32	−0.12	0.26
43: Yellow bellied Flycatcher	0.37	−0.22	0.18	93: Clark Nutcracker	0.32	0.20	0.31
44: Frigatebird	0.25	0.04	0.18	94: White breasted Nuthatch	0.24	0.11	0.34
45: Northern Fulmar	0.14	0.12	0.16	95: Baltimore Oriole	0.27	0.07	0.30
46: Gadwall	0.32	−0.04	0.36	96: Hooded Oriole	0.27	0.17	0.19
47: American Goldfinch	0.38	0.37	0.37	97: Orchard Oriole	0.30	0.08	0.17
48: European Goldfinch	0.45	0.13	0.32	98: Scott Oriole	0.43	0.23	0.43
49: Boat tailed Grackle	0.32	0.08	0.25	99: Ovenbird	0.23	0.00	0.13
50: Eared Grebe	0.26	−0.10	0.14	100: Brown Pelican	0.22	−0.45	0.06

TABLE 5.4: Full list of CGIM scores for classes in CUB dataset [55] with reproduced post-hoc CBMs [37] (Table 5.3 continued).

Class \ *CGIM*	$\rho_k^{\text{CGIM}_1}$	$\rho_k^{\text{CGIM}_2}$	$\rho_k^{\text{CGIM}_3}$	*Class* \ *CGIM*	$\rho_k^{\text{CGIM}_1}$	$\rho_k^{\text{CGIM}_2}$	$\rho_k^{\text{CGIM}_3}$
101: White Pelican	0.19	0.03	0.05	151: Black-capped Vireo	0.25	0.02	0.27
102: Western Wood Pewee	0.26	0.05	0.23	152: Blue-headed Vireo	0.35	0.03	0.30
103: Sayornis	0.25	−0.16	0.13	153: Philadelphia Vireo	0.19	0.00	0.20
104: American Pipit	0.27	0.09	0.27	154: Red-eyed Vireo	0.20	0.00	0.32
105: Whip poor will	0.24	−0.21	0.34	155: Warbling Vireo	0.34	0.10	0.14
106: Horned Puffin	0.37	0.24	0.35	156: White-eyed Vireo	0.31	0.05	0.25
107: Common Raven	0.32	0.24	0.33	157: Yellow-throated Vireo	0.21	0.16	0.13
108: White-necked Raven	0.35	0.22	0.35	158: Bay-breasted Warbler	0.32	−0.12	0.22
109: American Redstart	0.28	0.12	0.24	159: Black-and-white Warbler	0.35	0.15	0.30
110: Geococcyx	0.24	−0.10	0.24	160: Black-throated Blue Warbler	0.26	0.02	0.27
111: Loggerhead Shrike	0.33	0.28	0.31	161: Blue-winged Warbler	0.26	0.36	0.35
112: Great Grey Shrike	0.28	0.19	0.31	162: Canada Warbler	0.27	0.06	0.17
113: Baird's Sparrow	0.24	0.12	0.31	163: Cape May Warbler	0.28	−0.05	0.30
114: Black-throated Sparrow	0.33	0.08	0.22	164: Cerulean Warbler	0.10	−0.10	0.21
115: Brewer's Sparrow	0.25	0.23	0.12	165: Chestnut-sided Warbler	0.27	0.02	0.35
116: Chipping Sparrow	0.28	0.02	0.21	166: Golden-winged Warbler	0.40	0.20	0.39
117: Clay-colored Sparrow	0.22	0.07	0.16	167: Hooded Warbler	0.26	0.17	0.43
118: House Sparrow	0.36	0.11	0.24	168: Kentucky Warbler	0.12	0.17	0.14
119: Field Sparrow	0.14	0.07	0.19	169: Magnolia Warbler	0.41	0.19	0.35
120: Fox Sparrow	0.22	0.10	0.22	170: Mourning Warbler	0.33	0.06	0.24
121: Grasshopper Sparrow	0.30	0.08	0.34	171: Myrtle Warbler	0.34	0.01	0.22
122: Harris's Sparrow	0.22	−0.01	0.31	172: Nashville Warbler	0.27	0.18	0.33
123: Henslow's Sparrow	0.34	0.10	0.27	173: Orange-crowned Warbler	0.21	0.00	0.18
124: Le Conte's Sparrow	0.36	0.09	0.29	174: Palm Warbler	0.18	−0.05	0.09
125: Lincoln Sparrow	0.28	0.22	0.21	175: Pine Warbler	0.23	0.23	0.19
126: Nelson's Sharp-tailed Sparrow	0.31	−0.13	0.17	176: Prairie Warbler	0.20	0.16	0.24
127: Savannah Sparrow	0.35	0.13	0.32	177: Prothonotary Warbler	0.33	0.37	0.42
128: Seaside Sparrow	0.15	−0.24	0.07	178: Swainson's Warbler	0.31	0.05	0.20
129: Song Sparrow	0.37	0.20	0.29	179: Tennessee Warbler	0.17	0.00	0.20
130: Tree Sparrow	0.38	0.10	0.20	180: Wilson's Warbler	0.20	0.25	0.34
131: Vesper Sparrow	0.16	0.12	0.17	181: Worm-eating Warbler	0.33	0.02	0.32
132: White-crowned Sparrow	0.40	0.12	0.33	182: Yellow Warbler	0.33	0.37	0.34
133: White-throated Sparrow	0.18	0.00	0.27	183: Northern Waterthrush	0.23	0.08	0.24
134: Cape Glossy Starling	0.40	0.14	0.34	184: Louisiana Waterthrush	0.27	0.01	0.19
135: Bank Swallow	0.17	−0.15	0.19	185: Bohemian Waxwing	0.29	0.23	0.28
136: Barn Swallow	0.37	0.09	0.22	186: Cedar Waxwing	0.40	0.12	0.25
137: Cliff Swallow	0.29	−0.13	0.12	187: American Three-toed Woodpecker	0.32	0.14	0.20
138: Tree Swallow	0.33	0.17	0.30	188: Pileated Woodpecker	0.31	0.27	0.26
139: Scarlet Tanager	0.30	0.10	0.09	189: Red-bellied Woodpecker	0.20	0.10	0.27
140: Summer Tanager	0.21	0.10	0.16	190: Red-cockaded Woodpecker	0.28	0.26	0.26
141: Arctic Tern	0.23	0.17	0.26	191: Red-headed Woodpecker	0.39	0.22	0.33
142: Black Tern	0.22	−0.07	0.27	192: Downy Woodpecker	0.27	0.24	0.27
143: Caspian Tern	0.12	0.20	0.26	193: Bewick Wren	0.20	0.25	0.30
144: Common Tern	0.17	0.11	0.26	194: Cactus Wren	0.39	0.17	0.29
145: Elegant Tern	0.22	0.23	0.25	195: Carolina Wren	0.33	0.27	0.30
146: Forsters Tern	0.17	0.25	0.24	196: House Wren	0.26	0.30	0.28
147: Least Tern	0.38	0.20	0.33	197: Marsh Wren	0.33	0.07	0.21
148: Green tailed Towhee	0.30	0.06	0.27	198: Rock Wren	0.24	0.19	0.22
149: Brown Thrasher	0.34	0.10	0.30	199: Winter Wren	0.20	0.32	0.16
150: Sage Thrasher	0.37	0.12	0.25	200: Common Yellowthroat	0.30	−0.04	0.30

TABLE 5.5: Concept existence assessment of the reproduced post-hoc CBMs under CEM for the top l most important concepts.

Image	CEM based on	$l = 1$	$l = 3$	$l = 5$
Entire test set	θ_{jk}	39.2	37.9	37.1
	$\hat{u}_{ij}$	84.3	80.1	77.2
	$\theta_{jk}\hat{u}_{ij}$	49.3	44.3	41.2
Correct class set	θ_{jk}	48.5	46.8	44.8
	$\hat{u}_{ij}$	85.4	82.1	80.7
	$\theta_{jk}\hat{u}_{ij}$	55.4	49.1	45.8

spotted catbird and class ($k = 25$) *pelagic cormorant* in Table 5.3 indicates that these classes share no similarities to their ground truth. For the first time, the low and negative valued CGIM scores occurring for many concepts and classes raise concerns about the reliability and quality of the explanations of the post-hoc CBMs.

5.5.3 Concept Existence Evaluation

After analysing the global importance evaluation based on the classifier's weights and average concept predictions, we, in this section, focus on the local importance analysis. The first step in this regard is to assess the concept existence qualitatively and quantitatively.

5.5.3.1 Qualitative observations

When a set of concepts is presented as highly important for a prediction by a trained model, it is essential to qualitatively verify whether these concepts really exist in the image. In Figure 5.3, we present random images from the test set with the top 5 most important concepts for their prediction outputted by the reproduced post-hoc CBM [37]. As shown in Figure 5.3, many of those highly important concepts do not actually exist in the given images. For instance, for an *American Redstart* in the first column, the most important concept is given as *white throat*; this is incorrect because the bird has a *black throat*, which can be clearly seen in the input image. Similarly, for the *brown pelican* image in the second column, the fifth most important concept is given as *shorten than head bill*; this is not the case as the pelican has a much longer bill than its head.

5.5.3.2 Quantitative test by CEM

We calculate the CEM score over the entire test set. The full results are presented in Table 5.5 in terms of ranking the importance of the concepts based on i) the weights of the classifier θ_{jk}, ii) the projected concept values $\hat{u}_{ij}$, and iii) their combination $\theta_{jk}\hat{u}_{ij}$, for the top l most important concepts with l set to 1, 3, and 5. The results show that

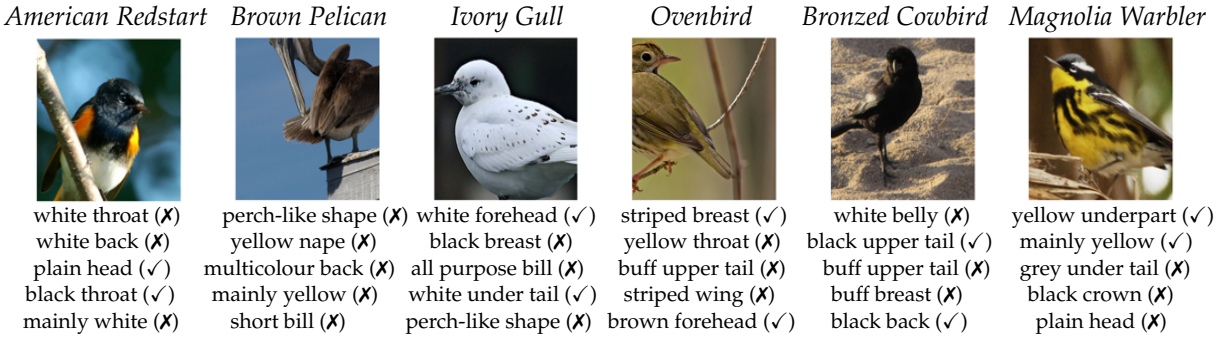


FIGURE 5.3: Randomly selected test images from different classes and the top 5 most important concepts for their classification by the post-hoc CBMs. In particular, symbol ✓ is for concept existence in the ground-truth label, while symbol ✗ indicates the concept absence in the ground-truth label.

TABLE 5.6: The number of concepts, grouped based on their types, and mapped to the parts (see Table 5.7 for more details).

Type / *Part*	**Color**	**Pattern**	**Shape**	***Total***
Back	6	3	—	9
Beak	3	2	4	9
Belly	6	1	—	7
Breast	6	3	—	9
Crown	6	—	—	6
Head	6	2	—	8
Eye	1	—	—	1
Leg	3	—	—	3
Wing	6	4	2	12
Nape	6	—	—	6
Tail	10	3	1	14
Throat	5	—	—	5
Others	18	—	5	23
Total	82	18	12	112

the CEM score based on $\hat{u}_{ij}$ is significantly higher than the others, which is intuitive at first glance as the highest values after concept projection are highly likely to be present in the ground-truth label. However, as detailed in Section 5.4, the concept values in $\hat{\boldsymbol{u}}_i$ do not independently determine the final class prediction; instead, these values are weighted by their respective weights in $\boldsymbol{\theta}$, which can significantly alter their overall impact. Relying solely on the projected concept values in $\hat{\boldsymbol{u}}_i$ may therefore lead to misleading conclusions. Hence, we build our argument based on $\theta_{jk}\hat{u}_{ij}$ rather than solely on $\hat{u}_{ij}$ or θ_{jk}. Strikingly, as shown in Table 5.5, the single most important concept (i.e., when $l = 1$) only exists in the images around 55% of the times when the image is correctly classified. This score drops to 49% when the test is done on the entire test set. Moreover, the CEM score is even lower when l is set to 3 and 5.

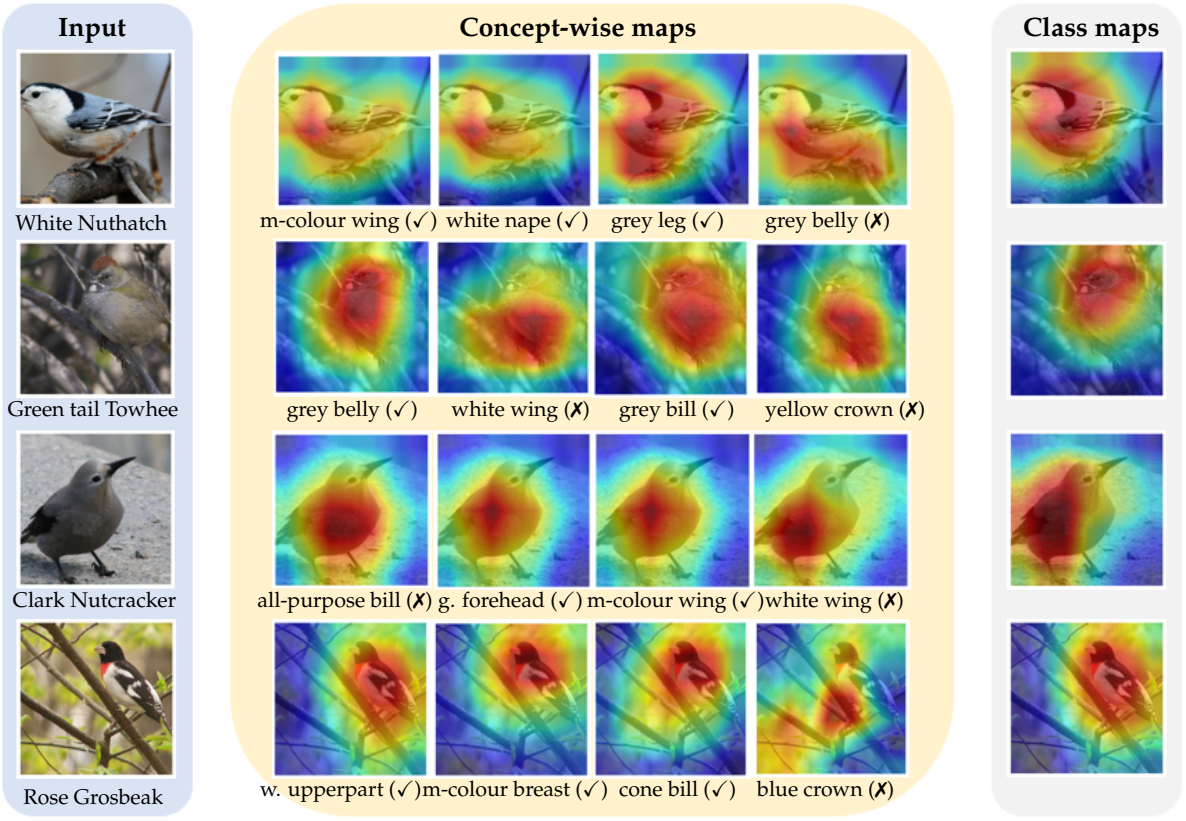


FIGURE 5.4: Class and concept visualisation with our CoAM. All images (on the left) are correctly classified and their class-wise saliency maps are given on the right. The four most important concepts under CEM for the given classifications and their individual saliency maps are given in the middle. In particular, symbol ✓ is for concept existence in the ground-truth label, while ✗ shows the concept absence in the ground-truth label.

5.5.4 Concept Localisation Evaluation

We now assess the concept of localisation qualitatively using our proposed CoAM and quantitatively by our proposed CLM.

TABLE 5.7: Details of the concepts and the parts they are mapped to.

Type / *Part*	**Color**	**Pattern (length for beak)**	**Shape**
Back	*brown, grey, yellow, black, white, buff*	*solid, striped, multi-coloured*	—
Beak	*grey, black, buff*	*same-as-head, shorter-than-head*	*dagger, hooked-seabird, all-purpose, cone*
Belly	*brown, grey, yellow, black, white, buff*	*solid*	—
Breast	*brown, grey, yellow, black, white, buff*	*solid, striped, multi-coloured*	—
Crown	*blue, brown, grey, yellow, black, white*	—	—
Head	*blue, brown, grey, yellow, black, white*	*eyebrow, plain*	—
Eye	*black*	—	—
Leg	*grey, black, buff*	—	—
Wing	*brown, grey, yellow, black, white, buff*	*solid, spotted, striped, multi-coloured*	*rounded, pointed*
Nape	*brown, grey, yellow, black, white, buff*	—	—
Tail	*brown, grey, black, white, buff*	*solid, striped, multi-colored*	*notched*
Throat	*grey, yellow, black, white, buff*	—	—

5.5.4.1 Qualitative observations

By visualizing concept heatmaps for different concepts using our proposed CoAM, we identified several recurring patterns. Figure 5.4 presents some examples of class and concept visualisation by using our CoAM. In many cases, the concept activation maps cover broad image regions, often extending beyond the expected concept areas. For instance, when detecting the concept *grey leg* in a *white breast nuthatch* image, the concept map covers the entire body of the bird rather than focusing on the specific region around the *leg*, as shown in the first row in Figure 5.4. Moreover, many of the fine-grained concepts such as *crown* or *tail pattern* are often not correctly localised. For instance, the heatmap highlights a region around the *leg* for *blue crown* concept as given in the last row of Figure 5.4.

5.5.4.2 Quantitative test by CLM

To be able to calculate the CLM score, the centre pixel coordinates for individual concepts are needed. In the `CUB` dataset, the centre pixel coordinates are only available for 12 broader body parts such as *beak, throat*, and *leg*. Fortunately, most of the 112 concepts are related to one of the 12 body parts, allowing us to match each concept to its closest body part and hence exploit the corresponding body-part coordinates for concepts. For instance, we match the *hooked seabird beak* concept with the *beak* part and the *solid wing* concept with the *wing*; see Tables 5.6 and 5.7 for details. We ignore concepts that are not related to a specific part such as overall size, shape and colour information, which leaves us with 89 out of 112 concepts for the CLM evaluation.

Recall that after obtaining the activation map for the j-th concept $\bar{\boldsymbol{F}}_{ij}$ of $\boldsymbol{X}_i$, CLM checks if the centre pixel location p_{ij} falls into the highest activated region Ω_{ij}. Here Ω_{ij} is formed by the $\alpha(M_1 M_2)/12$ number of pixels in terms of the largest pixel intensities in $\bar{\boldsymbol{F}}_{ij}$, where α is a hyperparameter that allows changing the region's size. For example, $\alpha = 1$ means the $1/12$ of the image is scanned, which is the size of a rough area for each of the 12 body parts such as *beak, back* and *throat* as given in Tables 5.6 and 5.7.

Table 5.8 gives the CLM scores for different choices of α. For $\alpha = 1$, only 13.3% of the time the centre pixel for a concept falls into the highly activated region Ω_{ij}. Even increasing α to 6, which means half of the image is scanned, the centre pixels of the individual concepts are still not in the concept locations 41% of the time.

5.6 Discussion

Learning human-understandable concepts is a challenging task. Often concepts are highly correlated with other features. For example, although hooves clearly relate to

TABLE 5.8: Concept localisation assessment of the reproduced post-hoc CBMs under CLM for the top l most important concepts.

Value α for Ω_{ij}	CLM based on	$l = 1$	$l = 3$	$l = 5$
$\alpha = 1$	θ_{jk}	10.8	13.6	13.8
	$\hat{u}_{ij}$	14.9	14.6	14.6
	$\theta_{jk}\hat{u}_{ij}$	13.3	13.5	12.9
$\alpha = 3$	θ_{jk}	29.6	30.1	31.2
	$\hat{u}_{ij}$	39.2	33.7	33.1
	$\theta_{jk}\hat{u}_{ij}$	33.4	32.2	31.8
$\alpha = 6$	θ_{jk}	52.3	50.8	51.6
	$\hat{u}_{ij}$	54.5	56.4	55.9
	$\theta_{jk}\hat{u}_{ij}$	59.0	55.0	53.9

the feet of animals, a group of hooved animals often share other common features (e.g., they are often quadrupeds that feed on grass). Although these other features might help to identify the concepts, it is not very helpful to be told that an important concept for determining that the image represents a cow is hooves if the hooves are not visible in the image. It is therefore important to check that the human-understandable concepts exist in the image and when they exist the network is finding them in the correct location. By providing measures and benchmarks we hope that this will provide an important stimulus to develop models with improved alignment.

Concept-based XAI methodologies showcase either global or local explainability of their proposed techniques, depending on their model setting. For instance, with traditional CBMs, the quality of concept predictions can be assessed just like the final class predictions since the concept labels should be readily available for them to work in the first place. On the other hand, when concept labels are not available as in the post-hoc CBMs case, the main evaluation is on the classifier weights (i.e., $\boldsymbol{\theta}$) as the global explicator. This evaluation is further supported by model editing experiments. To the best of our knowledge, there are no well-known evaluation measures other than model editing experiments, where some concepts are deliberately removed during test time to see how well the said concept is learnt as important for the prediction. However, none of these experiments makes a comparison between these weights and the ground-truth labels (i.e., $\boldsymbol{V}$ for the `CUB` dataset), hindering their reliability. Therefore, we propose CoAM and CGIM to visualise and evaluate the global explainability of concept-based XAI methodologies more rigorously and surface the alignment between the global concept explanations and ground-truth labels. Moreover, when the ground-truth concept labels are not available, for methods such as post-hoc CBMs, they are unable to do concept predictions and instead project concepts to the concept space, which would prevent a direct concept prediction evaluation even if the test set had ground-truth labels for each concept. Our CEM and CLM allow local concept evaluation regardless of the training settings of methodologies, i.e., whether they predict or project concepts to the concept space.

We should note, however, that the measures we provide do not directly measure the usefulness of the concepts as an explanation. Rather it acts as a sanity check that the concepts are correctly identified in the images. Also, success on the benchmark is not necessarily the top objective of a network; for example, the motivation of the post-hoc CBMs [37] was to provide a low-cost means of building traditional CBMs. A traditional CBM that uses per-image annotation will surely have a superior performance on our benchmark, but it may be too costly to train this on other datasets.

There are drawbacks to the measures CEM and CLM that we propose. The CEM can only be used on datasets where we have per-image annotations of the concepts for a test set (note that for our case, only a small number of concepts like the top $l \ll L$ are required per-image, and therefore is cheap). This limits its use to a very small number of datasets. Having a measure limited to one (or a small number of datasets) runs the risk that models are developed that overfit to that particular dataset. Applications to other datasets in other domains would undoubtedly be helpful for widespread use of our proposed measures. These domains would include healthcare where explainability is crucial and necessary for experts to trust AI predictions. Nevertheless, we had to leave this as part of future work and are hopeful to see our measures used in a variety of domains in the near future. The CLM requires knowledge of the location of the concepts. In fact, the concept locations were not given and we had to do a "best guess" approximation of whether the concepts found in the "saliency maps" overlap with the real concept locations. It is also debatable whether the heatmaps we obtained by weighting the feature maps before doing GAP correctly capture the location of the concepts. In our judgment, this seems as fair an estimate of the position as we can make. We feel there is considerable value in visualising the location of a concept through the use of heatmaps. In Chapter 3, we built saliency maps for each concept, but there we aligned each feature map to a concept which prevented cross-contamination between concept locations. By providing visualisations of the parts of the image that activated the concept, it made it much easier to assess the alignment of concepts in that model. We have attempted to provide a similar visualisation for the post-hoc CBMs [37], although as this is not part of the design of that model the visualisation may not be perfect. Finally, reducing the assessment of alignment to a couple of numbers loses a lot of fine-grain detail. As we illustrated, we can get a better understanding of the failure of the network by examining the performance in more detail, for example, by plotting histograms of the CBMs results to identify particularly poor concepts, or by visualising the locations of the features to understand what concepts might be being learnt.

Despite those drawbacks, we believe that proposing a new benchmark for assessing concept alignment has the potential to concentrate the effort of researchers on improving the performance of concept-based XAI systems. As we have illustrated, the performance of post-hoc CBMs is surprisingly poor. Without doing a systematic analysis of this alignment, it is easy to overlook this problem and believe that an XAI system is

more powerful than it actually is. Our hope is that by introducing a new benchmark we can improve the accuracy of future concept-based XAI systems.

Our findings raise important questions about the utility of current concept-based explanation methodologies in providing spatially grounded explanations for image-based tasks. While these models offer some degree of interpretability by linking decisions to human-understandable concepts, their failure to predict and localise concepts correctly can lead to misleading interpretations. This highlights the importance of more rigorous evaluation criteria such as CGIM, CEM and CLM and the development of models that prioritise both concept prediction accuracy and spatial interpretability.

5.7 Conclusion

In this chapter, we proposed three novel measures, i.e., CGIM, CEM and CLM, for concept-based XAI systems. CGIM provides a way to measure the global concept alignment ability of concept-based XAI techniques. CEM and CLM are introduced for local importance evaluation, testing if highly important concepts by XAI techniques exist and can be correctly localised in a given test image, respectively. Employing these three measures, we benchmarked post-hoc CBMs on the `CUB` dataset. Our experiments demonstrated significant limitations in current post-hoc methods, with many concepts and classes found to be weakly or even negatively correlated with their ground-truth labels by CGIM. Moreover, many concepts presented as highly important are not found to be present in test images by CEM, and their concept activations fail to align with the expected regions of the input images by CLM. As the field of XAI continues to evolve, it is essential to ensure that methods not only provide understandable concepts but also accurately predict and localise these concepts within input data. Future work may focus on improving both the concept prediction and spatial localisation capabilities of concept-based XAI methods, ensuring that they can offer reliable and interpretable insights across diverse applications.

Chapter 6

Conclusions and Future Directions

This thesis investigates the black-box nature of DNNs and underscores the importance of the tools and techniques to interpret these complex models. In this context, it provides a comprehensive review of well-known methodologies aimed at enhancing the interpretability, fairness, and transparency of these networks, particularly within the context of computer vision. By identifying key challenges and gaps in the field, we propose targeted solutions to address these limitations. In this chapter, we summarise our findings, highlight the current limitations of our work, and discuss potential strategies to overcome these limitations, outlining promising directions for future research.

6.1 Multilevel XAI

In Chapter 3, we define the types and levels of explanations that are both intuitive and desirable for humans to comprehend and trust machine predictions in the context of computer vision. We argue that explanations should be *multilevel*—combining high-level, human-understandable concepts with their corresponding saliency maps. We demonstrate that many well-known methodologies in the XAI field fall short of achieving this multilevel nature, as they typically produce single-level explanations, such as coarse-grained, object-wide saliency maps or simple lists of concepts. To address this limitation, we introduced multilevel XAI—a novel approach that generates multilevel, human-like explanations. By doing so, we aim to enhance the transparency, accessibility, and interpretability of deep networks.

Multilevel XAI can be viewed as a variant of CBMs, as it introduces an intermediate concept block that bridges high-level features learnt by deep networks and their final class predictions. However, multilevel XAI offers significant advantages over traditional CBMs. Firstly, it requires only *class-wise* concept annotations, drastically reducing annotation costs by several orders of magnitude depending on the task, compared

to the *image-wise* concept annotations typically required by CBMs. Secondly, multilevel XAI generates *concept-wise saliency maps* by product, elevating the level of explanation from a single layer to a more comprehensive two-layer representation. Furthermore, these by-product concept-wise saliency maps enable intervention at the pixel level, offering a more intuitive and user-friendly approach compared to the popular concept-level interventions (see Figures 3.17 and 3.18 for examples).

There are limitations to our multilevel XAI approach. While it significantly reduces annotation costs, it still requires additional annotations, i.e., class-wise concept labels, to generate explanations. These annotations, sourced from human annotators, may necessitate domain-specific expertise, especially for tasks like medical image classification. Reducing this dependency, for instance, by leveraging large language models, is a promising avenue for future research.

Another critical challenge lies in identifying the optimal set of concepts. Ideally, this set should be highly representative of the task-specific classes while ensuring minimal overlap or dependency among the concepts. This remains an open research area that warrants further exploration.

A further limitation involves the generation of concept-wise saliency maps. While the multilevel XAI approach successfully produces these maps for many concepts, generating intuitive and precise maps that clearly delineate regions corresponding to every individual concept remains challenging. These challenges often arise from inherent complexities in explanations, such as distinguishing circumstantial evidence from contextual cues or defining atomic versus compound concepts. In Section 3.6, we delve deeper into these issues and encourage the research community to address them to advance the field of concept-based explanations.

Despite these limitations, the multilevel XAI approach offers significant advantages to the concept-based XAI field. Promising future directions include developing automated methods for concept labelling, identifying optimal concept sets for specific tasks, and refining saliency map generation.

6.2 Semantic Proportions-based Semantic Segmentation

Semantic segmentation tasks are inherently challenging due to the expensive per-pixel annotation requirements. To address this, several forms of weak supervision—such as bounding boxes and scribbles—have been proposed in the field of WSSS (weakly supervised semantic segmentation) to reduce these costs. In Chapter 4, we introduced a novel WSSS methodology, SPSS, which has been demonstrated to be a significantly cost-efficient approach for achieving semantic segmentation while maintaining an acceptable level of accuracy. Our experiments show that SPSS enables the prediction of

high-quality segmentation maps using only rough class proportions per image. This is significant because it requires minimal effort for annotation: just a few numerical values per image (one value per class). By doing so, SPSS reduces the annotation costs by several orders of magnitude in terms of both time and size compared to full per-pixel annotations, as highlighted in Table 4.6.

In the experiments presented in Chapter 4, we obtained semantic class proportion information either from existing segmentation maps or through limited manual annotations by a small group of annotators, as illustrated in Figure 4.9. The latter approach was particularly employed to demonstrate that precise class proportions, as in the former, are not strictly necessary—rough estimates are sufficient for SPSS to achieve satisfactory segmentation predictions. To further validate this, we conducted ablation studies in Section 4.5.2, where noise and cluster experiments demonstrated that SPSS performs reasonably well even when the precise proportions were significantly altered.

Additionally, as shown in Table 4.7, our results indicated that when the number of training images is severely restricted, the models trained using semantic proportions outperform those relying on costly per-pixel annotations. This finding is particularly significant, as having access to a limited number of segmentation-annotated images is a common constraint in real-world applications, further highlighting the practical appeal of the SPSS approach.

Our SPSS methodology does have some limitations. While the pixel accuracies achieved by it are promising, they are not as high as those obtained through per-pixel training—a drawback common to most WSSS techniques. Future work will focus on developing smart solutions, such as alternative training objectives or innovative ways to leverage semantic proportions to help bridge the gap and bring results closer to the optimal accuracy achieved with per-pixel annotations. For instance, combining semantic proportions with other forms of weak supervision, as seen in other methodologies using segmentation maps, could be one of the directions.

While we have demonstrated the low cost of obtaining rough class proportions compared to per-pixel annotations, future work will focus on automating this annotation process to further reduce costs. One promising direction involves leveraging the concept-wise saliency maps generated by the multilevel XAI technique introduced in Chapter 3. This direction requires an initial qualitative and quantitative comparison between the segmentation maps that our SPSS generates and the saliency maps obtained by traditional XAI methodologies for a fairer judgment. More importantly, their collaborative use in the semantic segmentation tasks is promising.

Using saliency maps from XAI methodologies for segmentation tasks is not a novel idea; prior research has successfully employed these maps as a form of weak supervision, particularly when combined with other weak annotations such as class labels or bounding boxes. The additional location information provided by saliency maps

enriches the supervision compared to scenarios where class labels, bounding boxes, or scribbles are used in isolation. However, it is worth noting a key distinction: the objective in our SPSS framework is as simple as predicting a few numerical values (i.e., class proportions) and obtaining the saliency maps by-product, whereas saliency map-based methodologies still aim to make per-pixel predictions. This introduces challenges; saliency maps as ground-truth often include a substantial number of false positives and negatives, making them far from ideal ground-truth segmentation maps. The inherent unreliability of these maps may limit further improvements in segmentation accuracy.

In Chapter 4, we demonstrated the effectiveness of SPSS in domains such as aerial imaging, where class boundaries are relatively distinct, and in medical imaging, where tasks are typically binary. However, semantic proportions may be insufficient in scenarios with a high number of classes in a single image, particularly when several objects occupy similar spatial regions or when certain class proportions are too small to be representative. Thus, SPSS may not be a universal solution for all segmentation challenges. However, its low annotation cost makes it a desirable starting point in cases where per-pixel annotations are infeasible or prohibitively expensive.

Looking forward, future work will include exploring domains and scenarios where SPSS performs effectively, as well as identifying cases where it is less suitable and should be avoided. This exploration will help refine the applicability of SPSS and guide its integration into a broader range of segmentation tasks.

6.3 Concept-Based Explainable Artificial Intelligence: Measures and Benchmarks

Concept-based XAI methodologies have gained popularity in recent years due to their intuitive and human-understandable outputs. However, the field lacks consensus on evaluation standards and widely accepted benchmark datasets, making it challenging to compare and validate techniques. Current works often rely on secondary evaluations. For instance, in some studies where the proposed methodology includes altering the DNN structure, such as by introducing a concept bottleneck to provide concept-level explanations, the evaluation is based on the observation that the model modification does not significantly degrade model prediction performance while providing explanations. Others conduct ablation studies, showing that removing highly important concepts results in significant changes in model predictions.

While these commonly used evaluation approaches provide some insights into the validity of the outputted concepts, they exhibit significant limitations. Specifically, they do not rigorously verify whether the highly important concepts identified by the

trained model truly exist or whether they spatially align with relevant parts of the examined image. We contend that the field requires more rigorous, comprehensive, and standardised evaluation criteria, along with widely accepted benchmark datasets, to enable fair and consistent comparisons.

To address these challenges, Chapter 5 introduces three novel and intuitive measures: CGIM, CEM, and CLM. CGIM facilitates global concept evaluation by assessing concept alignment, while CEM and CLM focus on local concept evaluation, examining concept presence and spatial localisation, respectively. Furthermore, we recommended the CUB [55] as a benchmark dataset for concept-based XAI methodologies.

Employing the CEM is relatively straightforward, as it simply verifies whether the k most important concepts are present in the ground-truth concept labels. In contrast, the CLM presents a greater challenge because most concept-based XAI techniques lack concept-wise saliency maps, offering only single-level outputs that indicate concept relevance without spatial localisation—a core limitation that motivates the multilevel XAI proposed in Chapter 3. To address this gap and enable CLM evaluation, we developed the CoAM framework, which extends single-level explanations into two-level outputs. This two-level structure includes both concept names and their corresponding saliency maps, similar to those produced by multilevel XAI. As a case study, we used CoAM to generate concept-wise saliency maps for the post-hoc CBM method [37]. A promising future direction involves automating the adaptation of CoAM to other concept-based methodologies, further broadening its applicability.

One of the main strengths of concept bottleneck models is that they allow model intervention, which can be used as feedback for model improvement. In this direction, post hoc CBMs stand out as they allow global intervention; unlike traditional CBMs, which only allow local intervention. However, both these well-known methodologies can achieve model intervention at a single level, i.e., by tweaking the predicted concept value before the final classification or changing the concept weight in the classification layer. We take this one step further and achieve model intervention on pixel level both in our proposed Multilevel XAI methodology in Chapter 3 and in Chapter 5 where we introduced CoAM for concepts visualisation for all CBMs. We believe that the model intervention made possible by our methodologies is promising for better analysis of concept-based models and hence their improvement.

Currently, we rely on the CUB dataset as our benchmark due to its inclusion of both concept existence labels and weak spatial annotations, such as centre pixel locations for several concepts. However, relying on a single dataset risks overfitting models to its specific characteristics. Fortunately, obtaining additional benchmarks is feasible, as many existing datasets already include per-image concept labels commonly used for training concept-based models. The primary challenge lies in the lack of spatial annotations required for CLM evaluation. Future efforts should prioritise augmenting

these datasets or annotating new benchmarks with spatial labels to mitigate overfitting and enhance the generalisability of concept-based explainability methods.

By introducing rigorous measures and a benchmark dataset, our work aims to make comparisons between concept-based XAI methodologies fair and standardised. A promising future direction is the creation of a common platform where researchers can upload their trained models and receive automated evaluations of their concept prediction and localisation capabilities. Through these contributions, we strive to establish a unified and standardised framework for evaluating concept-based explainability, addressing critical gaps in the field.

In summary, by proposing the CGIM, CEM and CLM as evaluation measures, developing the CoAM framework for spatial map generation, and promoting `CUB` as a benchmark dataset, we aim to foster consistency and comparability in future research. These efforts pave the way for more robust, interpretable, and impactful XAI methodologies.

References

[1] Chanwoo Kim, Soham U Gadgil, Alex J DeGrave, Jesutofunmi A Omiye, Zhuo Ran Cai, Roxana Daneshjou, and Su-In Lee. Transparent medical image ai via an image–text foundation model grounded in medical literature. *Nature Medicine*, pages 1–12, 2024.

[2] Fei Wang, Lawrence Peter Casalino, and Dhruv Khullar. Deep Learning in Medicine—Promise, Progress, and Challenges. *JAMA Internal Medicine*, 179(3): 293–294, 2019.

[3] Robert Bogue. The role of artificial intelligence in robotics. *Industrial Robot: An International Journal*, 41(2):119–123, 2014.

[4] Mohsen Soori, Behrooz Arezoo, and Roza Dastres. Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*, 3:54–70, 2023.

[5] Steve J Bickley, Ho Fai Chan, and Benno Torgler. Artificial intelligence in the field of economics. *Scientometrics*, 127(4):2055–2084, 2022.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[7] Ravpreet Kaur and Sarbjeet Singh. A comprehensive review of object detection with deep learning. *Digital Signal Processing*, 132:103812, 2023.

[8] Ryo Akita, Akira Yoshihara, Takashi Matsubara, and Kuniaki Uehara. Deep learning for stock prediction using numerical and textual information. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pages 1–6. IEEE, 2016.

[9] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[10] Zeyue Xue, Guanglu Song, Qiushan Guo, Boxiao Liu, Zhuofan Zong, Yu Liu, and Ping Luo. Raphael: Text-to-Image Generation via Large Mixture of Diffusion Paths. *Advances in Neural Information Processing Systems*, 36:41693–41706, 2023.

[11] Shashi Pal Singh, Ajai Kumar, Hemant Darbari, Lenali Singh, Anshika Rastogi, and Shikha Jain. Machine translation using deep learning: An overview. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pages 162–167. IEEE, 2017.

[12] Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtskỳ. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(1):1–15, 2020.

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, 2015.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[15] Erik Lindholm, John Nickolls, Stuart Oberman, and John Montrym. NVIDIA Tesla: A Unified Graphics and Computing Architecture. *IEEE Micro*, 28(2):39–55, 2008.

[16] Vikas Hassija, Vinay Chamola, Atmesh Mahapatra, Abhinandan Singal, Divyansh Goel, Kaizhu Huang, Simone Scardapane, Indro Spinelli, Mufti Mahmud, and Amir Hussain. Interpreting Black-box Models: A Review on Explainable Artificial Intelligence. *Cognitive Computation*, 16(1):45–74, 2024.

[17] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys (CSUR)*, 51(5):1–42, 2018.

[18] Sajid Ali, Tamer Abuhmed, Shaker El-Sappagh, Khan Muhammad, Jose M Alonso-Moral, Roberto Confalonieri, Riccardo Guidotti, Javier Del Ser, Natalia Díaz-Rodríguez, and Francisco Herrera. Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Information Fusion*, 99:101805, 2023.

[19] Hani Hagras. Toward Human-Understandable, Explainable AI. *Computer*, 51(9): 28–36, 2018.

[20] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy*, 23(1): 18, 2020.

[21] David Gunning and David Aha. DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Magazine*, 40(2):44–58, 2019.

[22] Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv preprint arXiv:1702.08608*, 2017.

[23] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A Survey on Neural Network Interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021.

[24] Zachary C Lipton. The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is Both Important and Slippery. *Queue*, 16(3):31–57, 2018.

[25] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.

[26] Arun Das and Paul Rad. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. *arXiv preprint arXiv:2006.11371*, 2020.

[27] Erico Tjoa and Cuntai Guan. A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813, 2020.

[28] Christian Meske and Enrico Bunde. Transparency and Trust in Human-AI-Interaction: The Role of Model-Agnostic Explanations in Computer Vision-Based Decision Support. In *Artificial Intelligence in HCI: First International Conference, AI-HCI 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings 22*, pages 54–69. Springer, 2020.

[29] Zachary C Lipton. The Doctor Just Won't Accept That! *arXiv preprint arXiv:1711.08037*, 2017.

[30] Sofia Bonicalzi. A matter of justice. The opacity of algorithmic decision-making and the trade-off between uniformity and discretion in legal applications of artificial intelligence. *Teoria. Rivista di filosofia*, 42(2):131–147, 2022.

[31] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining Explanations: An Overview of Interpretability of Machine Learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89. IEEE, 2018.

[32] Council of European Union. Complete guide to GDPR compliance. https://gdpr.eu/. Accessed 2024-10-21.

[33] Amina Adadi and Mohammed Berrada. Peeking Inside the Black-box: a Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[34] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint arXiv:1312.6034*, 2013.

[35] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PloS One*, 10 (7):e0130140, 2015.

[36] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *International Conference on Machine Learning*, pages 2668–2677. PMLR, 2018.

[37] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc Concept Bottleneck Models. In *The Eleventh International Conference on Learning Representations*, 2023.

[38] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[39] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[40] Satya M Muddamsetty, NS Jahromi Mohammad, and Thomas B Moeslund. Sidu: Similarity Difference and Uniqueness Method for Explainable AI. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3269–3273. IEEE, 2020.

[41] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept Bottleneck Models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

[42] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

[43] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-Precision Model-Agnostic Explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[44] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

[45] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized Input Sampling for Explanation of Black-Box Models. *arXiv preprint arXiv:1806.07421*, 2018.

[46] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. The What-If Tool: Interactive Probing of Machine Learning Models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):56–65, 2019.

[47] Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilović, et al. AI Explainability 360 Toolkit. In *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*, pages 376–379, 2021.

[48] Erin LeDell and Sebastien Poirier. H2O AutoML: Scalable Automatic Machine Learning. In *Proceedings of the AutoML Workshop at ICML*, volume 2020, page 24, 2020.

[49] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6541–6549, 2017.

[50] Halil Ibrahim Aysel, Xiaohao Cai, and Adam Prugel-Bennett. Multilevel Explainable Artificial Intelligence: Visual and Linguistic Bonded Explanations. *IEEE Transactions on Artificial Intelligence*, 5(5):2055–2066, 2023.

[51] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37 (3):362–386, 2020.

[52] Ravi Aggarwal, Viknesh Sounderajah, Guy Martin, Daniel SW Ting, Alan Karthikesalingam, Dominic King, Hutan Ashrafian, and Ara Darzi. Diagnostic accuracy of deep learning in medical imaging: a systematic review and meta-analysis. *NPJ Digital Medicine*, 4(1):65, 2021.

[53] Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text2Concept: Concept Activation Vectors Directly From Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3744–3749, 2023.

[54] Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive Concept Bottleneck Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5948–5955, 2023.

[55] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[56] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[57] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[58] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[59] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper With Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[61] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[62] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*, 2017.

[63] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 303–310, 2018.

[64] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[65] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt V2: Co-Designing and Scaling ConvNets With Masked Autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16133–16142, 2023.

[66] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[67] Nello Cristianini and Elisa Ricci. Support Vector Machines. In *Encyclopedia of Algorithms*, pages 928–932. Springer, 2008.

[68] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[69] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. IEEE, 1999.

[70] Mohamed Abbas Hedjazi, Ikram Kourbane, and Yakup Genc. On identifying leaves: A comparison of CNN with classical ML methods. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2017.

[71] Christoph Molnar. *Interpretable Machine Learning*. 3 edition, 2025. ISBN 978-3-911578-03-5. URL https://christophm.github.io/interpretable-ml-book.

[72] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

[73] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[74] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object Detectors Emerge in Deep Scene CNNs. *arXiv preprint arXiv:1412.6856*, 2014.

[75] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–25, 2020.

[76] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

[77] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks. *arXiv preprint arXiv:1602.03616*, 2016.

[78] Judy Borowski, Roland S Zimmermann, Judith Schepers, Robert Geirhos, Thomas SA Wallis, Matthias Bethge, and Wieland Brendel. Exemplary Natural Images Explain CNN Activations Better than State-of-the-Art Feature Visualization. *arXiv preprint arXiv:2010.12606*, 2020.

[79] Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah. Curve Detectors. *Distill*, 5(6):e00024–003, 2020.

[80] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding Neural Networks Through Deep Visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[81] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for Simplicity: The All Convolutional Net. *arXiv preprint arXiv:1412.6806*, 2014.

[82] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pages 2018–2025. IEEE, 2011.

[83] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[84] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. In *International Conference on Machine Learning*, pages 3145–3153. PMlR, 2017.

[85] Maximilian Kohlbrenner, Alexander Bauer, Shinichi Nakajima, Alexander Binder, Wojciech Samek, and Sebastian Lapuschkin. Towards Best Practice in Explaining Neural Network Decisions with LRP. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020.

[86] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-Wise Relevance Propagation: An Overview. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 193–209, 2019.

[87] Yeon-Jee Jung, Seung-Ho Han, and Ho-Jin Choi. Explaining CNN and RNN Using Selective Layer-Wise Relevance Propagation. *IEEE Access*, 9:18670–18681, 2021.

[88] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018.

[89] Xuhong Li, Haoyi Xiong, Xingjian Li, Xiao Zhang, Ji Liu, Haiyan Jiang, Zeyu Chen, and Dejing Dou. G-lime: Statistical learning for local interpretations of deep neural networks using global priors. *Artificial Intelligence*, 314:103823, 2023.

[90] Zhengze Zhou, Giles Hooker, and Fei Wang. S-LIME: Stabilized-LIME for Model Explanation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2429–2438, 2021.

[91] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent Individualized Feature Attribution for Tree Ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[92] Adriano Lucieri, Muhammad Naseer Bajwa, Stephan Alexander Braun, Muhammad Imran Malik, Andreas Dengel, and Sheraz Ahmed. On Interpretability of Deep Learning based Skin Lesion Classifiers using Concept Activation Vectors. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2020.

[93] Ramon Correa, Khushbu Pahwa, Bhavik Patel, Celine M Vachon, Judy W Gichoya, and Imon Banerjee. Efficient adversarial debiasing with concept activation vector—Medical image case-studies. *Journal of Biomedical Informatics*, 149: 104548, 2024.

[94] Jonathan Crabbé and Mihaela van der Schaar. Concept Activation Regions: A Generalized Framework for Concept-based Explanations. *Advances in Neural Information Processing Systems*, 35:2590–2607, 2022.

[95] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th International Conference on Computer Vision*, pages 365–372. IEEE, 2009.

[96] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009.

[97] David Steinmann, Wolfgang Stammer, Felix Friedrich, and Kristian Kersting. Learning to Intervene on Concept Bottlenecks. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 46556–46571. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/steinmann24a.html.

[98] Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. A Closer Look at the Intervention Procedure of Concept Bottleneck Models. In *International Conference on Machine Learning*, pages 31504–31520. PMLR, 2023.

[99] Nishad Singhi, Jae Myung Kim, Karsten Roth, and Zeynep Akata. Improving Intervention Efficacy via Concept Realignment in Concept Bottleneck Models. In *European Conference on Computer Vision*, pages 422–438. Springer, 2024.

[100] Moritz Vandenhirtz, Sonia Laguna, Ričards Marcinkevičs, and Julia Vogt. Stochastic Concept Bottleneck Models. *Advances in Neural Information Processing Systems*, 37:51787–51810, 2024.

[101] Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing Leakage in Concept Bottleneck Models. *Advances in Neural Information Processing Systems*, 35: 23386–23397, 2022.

[102] Lijie Hu, Chenyang Ren, Zhengyu Hu, Hongbin Lin, Cheng-Long Wang, Hui Xiong, Jingfeng Zhang, and Di Wang. Editable Concept Bottleneck Models. *arXiv preprint arXiv:2405.15476*, 2024.

[103] Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic Concept Bottleneck Models. *arXiv preprint arXiv:2306.01574*, 2023.

[104] Chenming Shang, Shiji Zhou, Hengyuan Zhang, Xinzhe Ni, Yujiu Yang, and Yuwang Wang. Incremental Residual Concept Bottleneck Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11030–11040, 2024.

[105] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[106] Siwon Kim, Jinoh Oh, Sungjin Lee, Seunghak Yu, Jaeyoung Do, and Tara Taghavi. Grounding Counterfactual Explanation of Image Classifiers to Textual Concept Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10942–10950, 2023.

[107] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep Multimodal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020.

[108] Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, Martin Jagersand, and Hong Zhang. A Comparative Study of Real-time Semantic Segmentation for Autonomous Driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 587–597, 2018.

[109] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. CLIP2Scene: Towards Label-Efficient 3D Scene Understanding by CLIP. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7030, 2023.

[110] Imran Qureshi, Junhua Yan, Qaisar Abbas, Kashif Shaheed, Awais Bin Riaz, Abdul Wahid, Muhammad Waseem Jan Khan, and Piotr Szczuko. Medical image segmentation using deep semantic-based methods: A review of techniques, applications and emerging trends. *Information Fusion*, 90:316–352, 2023.

[111] Tashvik Dhamija, Anunay Gupta, Shreyansh Gupta, Anjum, Rahul Katarya, and Ghanshyam Singh. Semantic segmentation in medical images through transfused convolution and transformer networks. *Applied Intelligence*, 53(1):1132–1148, 2023.

[112] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[113] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[114] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

[115] Zilong Zhong, Zhong Qiu Lin, Rene Bidart, Xiaodan Hu, Ibrahim Ben Daya, Zhifeng Li, Wei-Shi Zheng, Jonathan Li, and Alexander Wong. Squeeze-and-Attention Networks for Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13065–13074, 2020.

[116] Zhaozheng Chen and Qianru Sun. Weakly-Supervised Semantic Segmentation with Image-Level Labels: from Traditional Models to Foundation Models. *arXiv preprint arXiv:2310.13026*, 2023.

[117] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and Semi-supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1742–1750, 2015.

[118] Alexander Kolesnikov and Christoph H Lampert. Seed, Expand and Constrain: Three Principles for Weakly-Supervised Image Segmentation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 695–711. Springer, 2016.

[119] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015.

[120] Jungbeom Lee, Jihun Yi, Chaehun Shin, and Sungroh Yoon. BBAM: Bounding Box Attribution Map for Weakly Supervised Semantic and Instance Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2643–2652, 2021.

[121] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Box-Driven Class-Wise Region Masking and Filling Rate Guided Loss for Weakly Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3136–3145, 2019.

[122] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. ScribbleSup: Scribble-Supervised Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3159–3167, 2016.

[123] Hyeonsoo Lee and Won-Ki Jeong. Scribble2Label: Scribble-Supervised Cell Segmentation via Self-generating Pseudo-Labels with Consistency. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23*, pages 14–23. Springer, 2020.

[124] Paul Vernaza and Manmohan Chandraker. Learning Random-Walk Label Propagation for Weakly-Supervised Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7158–7166, 2017.

[125] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the Point: Semantic Segmentation with Point Supervision. In *European Conference on Computer Vision*, pages 549–565. Springer, 2016.

[126] Yang Liu, Lijin Lian, Ersi Zhang, Lulu Xu, Chufan Xiao, Xiaoyun Zhong, Fang Li, Bin Jiang, Yuhan Dong, Lan Ma, et al. Mixed-unet: Refined class activation mapping for weakly-supervised semantic segmentation with multi-scale inference. *Frontiers in Computer Science*, 4:1036934, 2022.

[127] Clemens Seibold, Johannes Künzel, Anna Hilsmann, and Peter Eisert. From Explanations to Segmentation: Using Explainable AI for Image Segmentation. *arXiv preprint arXiv:2202.00315*, 2022.

[128] Rachel Lea Draelos and Lawrence Carin. Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks. *arXiv preprint arXiv:2011.08891*, 2020.

[129] Rokas Gipiškis, Chun-Wei Tsai, and Olga Kurasova. Explainable AI (XAI) in image segmentation in medicine, industry, and beyond: A survey. *ICT Express*, 2024.

[130] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88:303–338, 2010.

[131] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[132] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[133] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. *Advances in Neural Information Processing Systems*, 30, 2017.

[134] Waddah Saeed and Christian Omlin. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, 263:110273, 2023.

[135] Octavio Loyola-Gonzalez. Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View. *IEEE Access*, 7:154096–154113, 2019.

[136] Will Douglas Heaven. Hundreds of AI tools have been built to catch covid. None of them helped. https://www.technologyreview.com/2021/07/30/1030329/machine-learning-ai-failed-covid-hospital-diagnosis-pandemic, 2021. Accessed: 2022-12-06.

[137] Michael Roberts, Derek Driggs, Matthew Thorpe, Julian Gilbey, Michael Yeung, Stephan Ursprung, Angelica I Aviles-Rivero, Christian Etmann, Cathal McCague, Lucian Beer, et al. Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans. *Nature Machine Intelligence*, 3(3):199–217, 2021.

[138] Phil McCausland. Self-driving Uber car that hit and killed woman did not recognize that pedestrians jaywalk. https://www.nbcnews.com/tech/tech-news/self-driving-uber-car-hit-killed-woman-did-not-recognize-n1079281, 2019. Accessed: 2021-10-05.

[139] Thor Olavsrud. 7 famous analytics and AI disasters. https://www.cio.com/article/190888/5-famous-analytics-and-ai-disasters.html, 2022. Accessed: 2023-01-11.

[140] Anirban Sarkar, Deepak Vijaykeerthy, Anindya Sarkar, and Vineeth N Balasubramanian. A Framework for Learning Ante-Hoc Explainable Models via Concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10286–10295, 2022.

[141] Andreas Holzinger, Anna Saranti, Christoph Molnar, Przemyslaw Biecek, and Wojciech Samek. Explainable AI Methods-a Brief Overview. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pages 13–38. Springer, 2020.

[142] Randy Goebel, Ajay Chander, Katharina Holzinger, Freddy Lecue, Zeynep Akata, Simone Stumpf, Peter Kieseberg, and Andreas Holzinger. Explainable AI: The New 42? In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 295–303. Springer, 2018.

[143] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal Explanations: Justifying Decisions and Pointing to the Evidence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8779–8788, 2018.

[144] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[145] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics*, 9(3):1350–1371, 2015.

[146] Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102:349–391, 2016.

[147] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking Semantic Segmentation: A Prototype View. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2582–2593, 2022.

[148] Wenguan Wang, Cheng Han, Tianfei Zhou, and Dongfang Liu. Visual Recognition with Deep Nearest Centroids. *arXiv preprint arXiv:2209.07383*, 2022.

[149] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-Shot Learning—A Comprehensive Evaluation of the Good, the Bad and the Ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, 2018.

[150] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A Deep Visual-Semantic Embedding Model. *Advances in Neural Information Processing Systems*, 26, 2013.

[151] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot Learning with Semantic Output Codes. *Advances in Neural Information Processing Systems*, 22, 2009.

[152] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-Embedding for Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1425–1438, 2015.

[153] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pages 2152–2161. PMLR, 2015.

[154] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-Based Classification for Zero-Shot Visual Object Categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2013.

[155] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

[156] Saeid Asgari Taghanaki, Kumar Abhishek, Joseph Paul Cohen, Julien Cohen-Adad, and Ghassan Hamarneh. Deep semantic segmentation of natural and medical Images: a review. *Artificial Intelligence Review*, 54:137–178, 2021.

[157] Ruixin Yang and Yingyan Yu. Artificial Convolutional Neural Network in Object Detection and Semantic Segmentation for Medical Imaging Analysis. *Frontiers in Oncology*, 11:638182, 2021.

[158] Huanle Zhang, Bo Han, Cheuk Yiu Ip, and Prasant Mohapatra. Slimmer: Accelerating 3D Semantic Segmentation for Mobile Augmented Reality. In *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 603–612. IEEE, 2020.

[159] Andres Milioto, Philipp Lottes, and Cyrill Stachniss. Real-time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2229–2235. IEEE, 2018.

[160] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[161] Shijie Hao, Yuan Zhou, and Yanrong Guo. A Brief Survey on Semantic Segmentation with Deep Learning. *Neurocomputing*, 406:302–321, 2020.

[162] Kyle Genova, Xiaoqi Yin, Abhijit Kundu, Caroline Pantofaru, Forrester Cole, Avneesh Sud, Brian Brewington, Brian Shucker, and Thomas Funkhouser. Learning 3D Semantic Segmentation with Only 2D Image Supervision. In *2021 International Conference on 3D Vision (3DV)*, pages 361–372. IEEE, 2021.

[163] Pedro O Pinheiro and Ronan Collobert. From Image-Level to Pixel-Level Labeling With Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1713–1721, 2015.

[164] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Zequn Jie, Yanhui Xiao, Yao Zhao, and Shuicheng Yan. Learning to segment with image-level annotations. *Pattern Recognition*, 59:234–244, 2016.

[165] Dim P Papadopoulos, Alasdair DF Clarke, Frank Keller, and Vittorio Ferrari. Training Object Class Detectors from Eye Tracking Data. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 361–376. Springer, 2014.

[166] R Austin McEver and BS Manjunath. PCAMs: Weakly Supervised Semantic Segmentation Using Point Supervision. *arXiv preprint arXiv:2007.05615*, 2020.

[167] Shuai Xie, Zunlei Feng, Ying Chen, Songtao Sun, Chao Ma, and Mingli Song. Deal: Difficulty-aware Active Learning for Semantic Segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[168] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning Semantic Segmentation from Synthetic Data: A Geometrically Guided Input-Output Adaptation Approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1841–1850, 2019.

[169] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[170] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[171] Haigen Hu, Yixing Zheng, Qianwei Zhou, Jie Xiao, Shengyong Chen, and Qiu Guan. MC-Unet: Multi-scale Convolution Unet for Bladder Cancer Cell Segmentation in Phase-Contrast Microscopy Images. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1197–1199. IEEE, 2019.

[172] Hao Chen, Wen Zhang, Xiaochao Yan, Yanbin Chen, Xin Chen, Mengjun Wu, Lin Pan, and Shaohua Zheng. Multi-organ Segmentation Based on 2.5D Semi-supervised Learning. In *MICCAI Challenge on Fast and Low-Resource Semi-supervised Abdominal Organ Segmentation*, pages 74–86. Springer, 2022.

[173] Mehwish Dildar, Shumaila Akram, Muhammad Irfan, Hikmat Ullah Khan, Muhammad Ramzan, Abdur Rehman Mahmood, Soliman Ayed Alsaiari, Abdul Hakeem M Saeed, Mohammed Olaythah Alraddadi, and Mater Hussen Mahnashi. Skin Cancer Detection: A Review Using Deep Learning Techniques. *International Journal of Environmental Research and Public Health*, 18(10):5479, 2021.

[174] Yufeng Cao, April Vassantachart, C Ye Jason, Cheng Yu, Dan Ruan, Ke Sheng, Yi Lao, Zhilei Liu Shen, Salim Balik, Shelly Bian, et al. Automatic detection and segmentation of multiple brain metastases on magnetic resonance image using asymmetric UNet architecture. *Physics in Medicine & Biology*, 66(1):015003, 2021.

[175] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC). *arXiv preprint arXiv:1902.03368*, 2019.

[176] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5(1):1–9, 2018.

[177] Mateusz Buda, Ashirbani Saha, and Maciej A Mazurowski. Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. *Computers in Biology and Medicine*, 109: 218–225, 2019.

[178] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[179] Bas HM Van der Velden, Hugo J Kuijf, Kenneth GA Gilhuijs, and Max A Viergever. Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Medical Image Analysis*, 79:102470, 2022.

[180] Vanessa Buhrmester, David Münch, and Michael Arens. Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey. *Machine Learning and Knowledge Extraction*, 3(4):966–989, 2021.

[181] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards Automatic Concept-based Explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

[182] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-Free Concept Bottleneck Models. *arXiv preprint arXiv:2304.06129*, 2023.

[183] Desai Saurabh and Harish G Ramaswamy. Ablation-CAM: Visual Explanations for Deep Convolutional Network via Gradient-free Localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision WACV*, pages 983–991, 2020.