Open University of Catalonia

Doctoral Thesis

# Contributions to Explainable Deep Learning Models

*Author:*
Gereziher Adhane

*Supervisors:*
Prof. David Masip Rodo,
Dr. Mohammad Mahdi Dehshibi

*A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy*

AIWell Lab, Network and Information Technologies

April 30, 2024

# Declaration of Authorship

I, Gereziher Adhane, declare that this thesis titled, "Contributions to Explainable Deep Learning Models" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- 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.
- Where I have consulted the published work of others, this is always clearly attributed.
- 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.
- I have acknowledged all main sources of help.
- 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.

Signed:

Date:

"*It always seems impossible until it's done.*"

Nelson Mandela

Open University of Catalonia

Faculty of Computer Science, Multimedia and Telecommunications

Network and Information Technologies

Doctor of Philosophy

**Contributions to Explainable Deep Learning Models**

by Gereziher Adhane

***Abstract***

Deep neural networks (DNNs) have revolutionised computer vision and artificial intelligence applications. Nonetheless, their opaque nature, susceptibility to biases, and challenges in explaining decisions have remained pressing concerns in the academic community. This research addresses these concerns by advancing techniques that enrich both the performance and interpretability of convolutional neural networks (CNNs), a specific type of DNNs that have achieved remarkable results in various computer vision tasks, such as image classification, object detection, and face recognition. Specifically, the thesis proposes novel methods for informative sample selection, uncertainty quantification, visual explanation, and explaining knowledge distillation (KD).

Certain samples are informative, diverse, or representative, helping the model learn more efficiently. Given the computational and time constraints often associated with training DNNs, informative sample selection (ISS) becomes particularly pivotal. However, implementing ISS is a non-trivial task, as it requires a robust metric for sample informativeness, a mechanism for optimising the selection process, and a way to account for the dynamic importance of samples during the training cycle. Therefore, we propose two novel ISS methods that leverage different informativeness measures: reinforcement learning and uncertainty quantification. The first method integrates a meta-learning approach with reinforcement learning to create an agent that filters out samples that could contribute to model overfitting and bias, enhancing model accuracy, robustness, and fairness. The second method uses Monte Carlo dropout to estimate the uncertainty of samples and select the most informative ones for annotation, both at training and testing time; this method employs human-in-the-loop approaches to reduce the labelling cost, enhance credibility, and improve the performance of CNNs. We compare and contrast the proposed methods and demonstrate their effectiveness on various classification tasks.

While the proposed methods for ISS can improve the efficiency and performance of DNN models, they do not address the challenge of explainability, which is crucial for designing responsible AI. Explainability refers to the ability of a DNN model to provide transparent and interpretable reasons for its decisions, especially in response to critical questions such as how a trained model concludes. However, the opaque nature of DNN models makes explainability difficult, requiring further work to explain the decision-making process. To address this, we propose two novel explainability techniques that leverage different aspects of the feature maps: relevance and uniqueness. The first visual explainability method, ADVISE, visualises and quantifies the relevance of each unit of the feature map to provide better visual explanations. ADVISE uses adaptive bandwidth kernel density estimation to assign a relevance score to each unit of the feature map for the predicted class and generates more explainable maps. However, existing visual explanation methods do not adequately capture the nuanced differences between models or measure the extent of unique attributes on each model (such as during knowledge distillation). Therefore, we further develop a novel method, UniCAM, that provides precise and interpretable mechanisms to quantify and visualise unique features, which can help to make the KD process

explainable. UniCAM captures and visualises the unique attributes of Teacher and Student models during KD and measures the knowledge learned during KD. Understanding such unique attributes is essential for model comparison, enhancement, and knowledge transfer processes during knowledge distillation.

In summary, this research proposes a comprehensive set of methods, techniques, and metrics for improving, mitigating, and explaining DNN models. The thesis evaluates the proposed methods on various image classification tasks, showing their effectiveness in tackling the issues of bias and opacity. The research contributes to both the academic community and practical applications.

***Resumen***

Las redes neuronales profundas (DNN) han revolucionado la visión por computadora y las aplicaciones de inteligencia artificial, pero su naturaleza opaca, susceptibilidad a sesgos y desafíos para explicar sus decisiones siguen siendo preocupaciones apremiantes en la comunidad académica. Esta investigación aborda estas preocupaciones mediante el avance de técnicas que mejoran tanto el rendimiento como la interpretabilidad de las redes neuronales convolucionales (CNN), un tipo específico de DNN que ha logrado resultados notables en diversas tareas de visión por computadora, como la clasificación de imágenes, la detección de objetos y el reconocimiento facial. Específicamente, la tesis propone métodos novedosos para la selección informativa de muestras, la cuantificación de la incertidumbre, la explicabilidad visual y la explicabilidad en algoritmos de destilación del conocimiento (KD).

Ciertas muestras son particularmente informativas, diversas o representativas, lo que ayuda al modelo a aprender de manera más eficiente. Dadas las limitaciones computacionales y de tiempo asociadas a menudo con el entrenamiento de DNN, la selección de muestras informativas (ISS) se vuelve particularmente fundamental. Sin embargo, implementar ISS no es una tarea trivial, ya que requiere una métrica sólida para la informatividad de la muestra, un mecanismo para optimizar el proceso de selección y una forma de tener en cuenta la importancia dinámica de las muestras durante el proceso de entrenamiento. En esta tesis doctoral, proponemos dos métodos novedosos para ISS que aprovechan diferentes medidas de informatividad: aprendizaje por refuerzo y cuantificación de la incertidumbre. El primer método integra un enfoque de metaaprendizaje con aprendizaje por refuerzo para crear un agente que filtre muestras que podrían contribuir al sobreajuste y al sesgo del modelo, mejorando su precisión, solidez y equidad. El segundo método utiliza algoritmos de Monte Carlo aplicados a las capas de DropOut para estimar la incertidumbre de las muestras y seleccionar las más informativas para su anotación, tanto en el momento del entrenamiento como en el de la inferencia. Este método introduce al anotador humano en el proceso con el objetivo de reducir el costo de etiquetado, mejorar la credibilidad y mejorar el rendimiento de las CNN. Comparamos y contrastamos los métodos propuestos y demostramos su eficacia en diversas tareas de clasificación.

Si bien los métodos propuestos para ISS pueden mejorar la eficiencia y el rendimiento de los modelos DNN, no abordan el desafío de la explicabilidad, que es crucial para diseñar una IA responsable. La explicabilidad se refiere a la capacidad de un modelo DNN de proporcionar razones transparentes e interpretables para sus decisiones, especialmente en respuesta a preguntas críticas como cómo un modelo entrenado llega a una decisión. Sin embargo, la naturaleza opaca de los modelos DNN hace que la explicabilidad sea una tarea difícil y requiere más trabajo para explicar el proceso de toma de decisiones. Para abordar esto, proponemos dos técnicas novedosas de explicabilidad que aprovechan diferentes aspectos de los mapas de características: relevancia y singularidad. El primer método de explicabilidad visual, ADVISE, visualiza y cuantifica la relevancia de cada unidad del mapa de características para proporcionar mejores explicaciones visuales. ADVISE utiliza una estimación de la densidad de probabilidades mediante un kernel con ancho de banda adaptativo para

asignar una puntuación de relevancia a cada unidad del mapa de características para la clase predicha y genera mapas más explicables. Sin embargo, los métodos de explicación visual existentes no capturan adecuadamente las diferencias especificas entre los modelos ni miden el alcance de los atributos únicos en cada modelo (como durante la destilación del conocimiento). En esta tesis, desarrollamos un método novedoso, UniCAM, que proporciona mecanismos precisos e interpretables para cuantificar y visualizar características únicas, que pueden ayudar a que el proceso KD sea explicable. UniCAM captura y visualiza los atributos únicos de los modelos de Profesor y Estudiante durante KD y mide el conocimiento aprendido durante KD. Comprender estos atributos únicos es esencial para los procesos de comparación, mejora y transferencia de conocimientos de modelos durante la destilación del conocimiento.

Esta investigación propone un conjunto integral de métodos, técnicas y métricas para mejorar, mitigar y explicar los modelos DNN. La tesis evalúa los métodos propuestos en diversas tareas de clasificación de imágenes, mostrando su eficacia para abordar los problemas de sesgo y opacidad. La investigación contribuye tanto a la comunidad académica como a las aplicaciones prácticas.

***Resum***

Les xarxes neuronals profundes (DNN) han revolucionat la visió per computador i les aplicacions d'intel · ligència artificial, però la seva naturalesa opaca, la seva susceptibilitat als biaixos i els reptes a l'hora d'explicar les seves decisions han continuat sent preocupacions rellevants a la comunitat acadèmica. Aquesta investigació aborda aquestes preocupacions avançant en tècniques que enriqueixen tant el rendiment com la interpretabilitat de les xarxes neuronals convolucionals (CNN), un tipus específic de DNN que han aconseguit resultats notables en diverses tasques de visió per computador, com ara la classificació d'imatges, la detecció d'objectes i el reconeixement facial. Concretament, la tesi proposa mètodes nous per a la selecció de mostres informatives, quantificació de la incertesa, explicació visual i explicació de la destil · lació del coneixement (KD).

Algunes mostres són particularment informatives, diverses o representatives, fet que ajuda al model a aprendre de manera més eficient. Tenint en compte les limitacions computacionals i de temps associades sovint a l'entrenament de DNN, la selecció de mostres informatives (ISS) esdevé especialment fonamental. Tanmateix, implementar ISS és una tasca no trivial, ja que requereix una mètrica sòlida per a modelar la informació de la mostra, un mecanisme per optimitzar el procés de selecció i una manera de tenir en compte la importància dinàmica de les mostres durant el procés d'entrenament. Per tant, proposem dos mètodes nous per a l'ISS que aprofiten diferents mesures d'informació: aprenentatge de reforç i quantificació de la incertesa. El primer mètode integra un enfocament de metaaprenentatge amb l'aprenentatge de reforç per crear un agent que filtra mostres que podrien contribuir a l'ajustament excessiu i al biaix del model, millorant-ne la precisió, la robustesa i l'equitat. El segon mètode utilitza simulació de Monte Carlo en les capes de DropOut per estimar la incertesa de les mostres i seleccionar les més informatives per a l'anotació, tant en el moment de l'entrenament com de la inferència. Aquest mètode utilitza enfocaments humans-in-the-loop per reduir el cost de l'etiquetatge, millorar la credibilitat i millorar el rendiment de les CNN. Comparem i contrastem els mètodes proposats i demostrem la seva eficàcia en diferents tasques de classificació.

Tot i que els mètodes proposats per a ISS poden millorar l'eficiència i el rendiment dels models DNN, no aborden el repte de l'explicabilitat, que és crucial per dissenyar una IA responsable. L'explicabilitat es refereix a la capacitat d'un model DNN per proporcionar argumentaris transparents i interpretables per a les seves decisions, especialment en resposta a preguntes crítiques com ara com un model entrenat arriba a una decisió. Tanmateix, la naturalesa opaca dels models DNN fa que l'explicabilitat sigui una tasca difícil, que requereix més treball per explicar el procés de presa de decisions. Per solucionar-ho, proposem dues noves tècniques d'explicació que aprofiten diferents aspectes dels mapes de característiques: la rellevància i la singularitat. El primer mètode d'explicació visual, ADVISE, visualitza i quantifica la rellevància de cada unitat del mapa de característiques per oferir millors explicacions visuals. ADVISE utilitza l'estimació de la densitat de probabilitat mitjançant kernels amb amplada de banda adaptativa per assignar una puntuació de rellevància a cada unitat del mapa de característiques per a la classe prevista i genera mapes més explicables.

Tanmateix, els mètodes d'explicació visual existents no capturen adequadament les diferències matisades entre models ni mesuren l'extensió dels atributs únics de cada model (com els que es predueixen durant la destil · lació del coneixement). En aquesta tesi, desenvolupem un nou mètode, UniCAM, que proporciona mecanismes precisos i interpretables per quantificar i visualitzar característiques úniques, que poden ajudar a fer que el procés KD sigui explicable. UniCAM captura i visualitza els atributs únics dels models de professor i estudiant durant el KD i mesura el coneixement après durant el KD. Entendre aquests atributs únics és essencial per als processos de comparació, millora i transferència de coneixement de models durant la destil · lació del coneixement.

En resum, aquesta investigació proposa un conjunt complet de mètodes, tècniques i mètriques per millorar, mitigar i explicar els models DNN. La tesi avalua els mètodes proposats en diferents tasques de classificació d'imatges, mostrant la seva eficàcia per abordar els problemes de biaix i opacitat. La investigació contribueix tant a la comunitat acadèmica com a aplicacions pràctiques.

# *Acknowledgements*

I am deeply grateful to my esteemed advisers, Prof. David Masip and Dr. Mohammad Mahdi Dehshibi, for their unwavering support and invaluable guidance throughout my research. Their wisdom and insights have been the pillars upon which this work stands. I extend my heartfelt thanks to Dr. Gemma Roig of Goethe University, who generously hosted me for a highly productive research visit. The remarkable collaboration we established has been a great privilege. To my family, whose emotional support remained steadfast despite the barriers to our communication, I owe an enormous debt of gratitude. I especially thank my friends—Mohammad, Nina, Nasibeh, Marta, Golshan, and Alaleh—who have made this experience enriching and memorable. Your friendship has been a source of comfort and encouragement. I am deeply thankful to the people of Spain, who welcomed me and made me feel at home during a time when my own country was torn by conflict, and travelling to my country was a privilege I could not attempt. Finally, I am also grateful to the vibrant UOC community, whose support was instrumental in my daily academic endeavours. Your collective spirit has been both inspiring and motivating. My journey has been both challenging and rewarding, and it is with a full heart that I share this success with all of you. Thank you!

# Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| ADVISE | **AD**aptive **VIS**ual **E**xplanation |
| AI | Artificial Intelligence |
| AL | Active learning |
| B-LRP | Bayesian Layer-Wise Relevance Propagation |
| CNN | Convolutional Neural Network |
| DNN | Deep Neural Network |
| ISS | Informative Sample Selection |
| KD | Knowledge Distillation |
| KDE | Kernel Density Estimation |
| MC | Monte Carlo |
| MDP | Markov Decision Process |
| MSE | Mean Squared Error |
| SSIM | Structural Similarity Index |
| XAI | Explainable AI |

*This thesis is dedicated to the.*

# Chapter 1

## Introduction

Convolutional Neural Networks (CNNs) have emerged as a dominant architecture in deep learning, finding utility across various applications, including computer vision [1, 2] to natural language processing (NLP)[1] [3, 4]. Despite their unprecedented success in complex tasks, they present critical challenges related to model explainability. CNNs are often difficult to interpret, as their internal structure consists of many layers and nodes that perform nonlinear transformations on the input data. This makes it hard to trace the reasoning behind their outputs and verify their reliability and validity. This lack of transparency is a significant concern, especially when CNNs are deployed in sensitive domains such as healthcare, autonomous vehicles, and criminal justice, where the stakes are high, and accountability is crucial. Previous studies highlight that the opacity inherent in deep neural networks underlines the necessity for greater transparency [5, 6].

Building on the imperative for greater transparency and explainability, delving into the specific challenges that undermine these objectives becomes crucial. Some foundational challenges in transparency are data bias and decision uncertainty [7, 8]. The quality and volume of data influence the performance of CNNs and their generalisability across different scenarios. Erroneous, redundant, or biased data can skew the model's learning process, resulting in overfitting, underfitting and opaque decision-making [9]. For instance, biased data has been shown to produce discriminatory results in applications as varied as facial recognition to criminal sentencing [10]. Therefore, carefully selecting unbiased and informative training samples is pivotal in rendering CNNs more responsible and ethically aligned.

Expanding on the necessity for responsible data handling, transparency at the data level becomes an essential component for responsible Artificial Intelligence (AI) systems [8, 11]. Understanding the data utilised in the training of CNNs illuminates how data influences both model behaviour and performance. Recent advancements in interactive data explainability tools further facilitate more responsible and transparent AI [12, 13, 14]. Such tools provide critical insights into the intricate workings of AI models, thus helping to minimise bias and enhance decision-making capabilities.

While mitigating data bias is a crucial step in addressing transparency, the issue of uncertainty in CNNs presents an equally significant challenge requiring another attention [15]. In complex decision-making scenarios, CNNs may produce outputs with

[1] In the context of Natural Language Processing (NLP), Convolutional Neural Networks typically employ one-dimensional convolutional filters due to the one-dimensional nature of text data. In contrast, in the image domain, CNNs use two-dimensional or three-dimensional convolutional filters to effectively process the two-dimensional structure of images.

high levels of uncertainty. Such uncertainty not only questions the reliability of these models but also exposes them to ethical scrutiny. Strategies to quantify and manage this uncertainty are another area for minimising bias and the risks associated with ambiguous decisions. Therefore, addressing the uncertainty during decision-making complements the drive towards AI systems that align with ethical and responsible standards.

Beyond the focus on data-centric and uncertainty challenges, the focus transitions to another critical aspect of responsible AI: transparency and interpretability in decision-making [15]. While reducing bias and decision uncertainty lays the groundwork for ethical AI, it does not suffice to fully understand the complexities of CNN-based decisions. Transparency and interpretability refer to the ability to understand how and why CNNs make their predictions. This involves analysing the input data, the network architecture, and the output results. For example, one can examine which features, regions, or channels of the input data are most relevant for the prediction, how the network layers and filters process the data, and what kind of patterns or knowledge the network learns from the data. This level of detail fosters trust, facilitates error identification, and offers domain-specific insights. Visual explanation techniques such as gradient-based techniques [16], perturbation-based methods [17, 18], activation-based methods [19], and decomposition-based methods [20] have been popular and contributed towards model transparency and interpretability. Therefore, achieving high levels of transparency and interpretability remains essential for ensuring the responsible deployment and societal acceptance of CNN-based systems.

Visual explanations are important for understanding the decision-making process of CNNs, but they also face challenges, especially when applied to complex learning paradigms such as Knowledge Distillation (KD). Despite the benefits of KD, there is still a lack of understanding about how and why it improves performance and what the student learned during KD. Existing visual explanation techniques often fail to explain the nuanced interaction between Teacher and Student models during the knowledge transfer process. The issue becomes even more complex when attempting to understand how explicit features are learned during KD [21, 22].

This thesis aims to develop novel methodologies for enhancing deep neural networks' performance, reliability, and explainability in computer vision applications. We argue that the quality and trustworthiness of Deep Neural Networks (DNNs) can be amplified by selecting informative and unbiased samples, rejecting uncertain or ambiguous decisions, and proposing more accurate visual explanation techniques to enhance interpretability during decision-making. To this end, we introduced new techniques grounded in meta-learning for informative sample selection, active learning based on model uncertainty estimation, visual explanation based on adaptive Kernel Density Estimation (KDE), and novel visual explanation techniques and metrics to explain the KD process. We evaluate the proposed methods on various datasets for classification tasks in the computer vision domain and compare them with the existing baseline techniques. Finally, the thesis elaborates on the proposed methods' implications, limitations, and future research directions.

## 1.1 Problem Statement

CNNs have profoundly impacted the field of computer vision, enabling unprecedented performance on a range of complex tasks. However, as these models become integral components in critical applications like healthcare, autonomous vehicles, and public safety, their opaque nature raises substantial ethical and practical concerns. Despite their achievements, CNNs grapple with challenges that severely limit their transparency, reliability, and societal acceptance. Specifically, these challenges encompass:

- **Data Quality:** The quality and quantity of labelled datasets are crucial for the success of computer vision models. However, many challenges limit the availability and diversity of data for various applications. Data can be noisy, unbalanced, or unrepresentative, directly affecting the model's performance, fairness, and generalisation. These issues can also propagate or exacerbate existing human biases. Moreover, due to the limited expertise and cost required for reliable manual annotations, most vision datasets are relatively small in scale compared to common benchmarks like ImageNet. Many datasets also lack varied representations across different demographics, ethnicities, and imaging equipment, which can restrict real-world applicability. To overcome these challenges, alternative training methods such as transfer learning, zero/few-shot learning, and knowledge distillation offer potential solutions to mitigate the impact of limited dataset size. Additionally, active learning techniques can optimise the annotation process by selectively identifying the most useful and ambiguous samples for labelling, and incorporating uncertainty estimation into the sample selection process can further improve model performance on underrepresented classes.
- **Model Uncertainty:** CNNs often produce a singular output without quantifying the level of uncertainty or confidence associated with that decision. This omission is problematic, especially in critical applications with high stakes, such as medical diagnostics or autonomous navigation. The absence of uncertainty metrics can result in overconfident decisions that might not reflect the model's true reliability, complicating risk assessment and decision-making processes.
- **Model Explainability:** CNNs often lack the ability to provide intuitive and meaningful explanations for their outputs, reducing their transparency and accountability. Lack of explainability compromises accountability, making it difficult to measure the model's confidence in its decisions or to ascertain its intended uses and limitations.

These challenges pose significant barriers to adopting and accepting CNNs in various domains that impact end-users, such as healthcare, education, finance, or security. Therefore, there is a need for developing techniques that can improve and interpret model transparency at various levels of the CNNs. One technique that can address these challenges and enhance model transparency at various levels of the CNNs is explainable artificial intelligence (XAI). XAI can help developers identify and overcome the bottlenecks of DNN architectures, such as data quality, model complexity, or decision uncertainty. XAI can also provide insights into the input data, the network

architecture, and the output results, which can foster trust, facilitate error identification, and offer domain-specific knowledge.

## 1.2 Research Objectives

The main objective of this work is to address the challenges hampering the transparency, reliability, and efficacy of CNNs. Achieving these objectives will facilitate the responsible adoption of CNNs for end-users, domain experts, developers and policymakers. To this end, the research is committed to advancing techniques that improve both the performance and interpretability of CNNs. Specifically, the research objectives are as follows:

- To develop robust sample selection methodologies that enhance both the performance and generalisability of CNNs. This will entail devising algorithms capable of selecting informative samples pivotal for optimising the model's training and testing phases.
- To integrate uncertainty quantification approaches into CNNs decision making through adaptive sample selection. These techniques will focus on identifying and filtering out ambiguous or uncertain samples, thereby revealing the model's uncertainty and using experts' opinions to enhance its reliability.
- To introduce a novel method for explaining the decisions of convolutional neural networks (CNNs) in image classification tasks, which can generate visual explanations that highlight the most important features, regions, or channels of the input image for the prediction and assign a relevance score to each unit of the feature maps.
- To develop comparative visual explanation tools customised for KD paradigms that aim to compare and contrast the teacher and student models, providing detailed insights into the knowledge transfer process.
- To propose robust metrics for evaluating the efficacy of visual explanation techniques and the efficiency of KD processes within CNNs. These metrics aim to quantitatively assess the quality of visual explanations and the effectiveness of knowledge transfer between teacher and student models.

## 1.3 Thesis Contributions

This thesis aims to contribute to XAI in computer vision and machine learning, specifically focusing on CNNs. It addresses key challenges in data quality, model uncertainty, and explainability of CNNs and KD. The research objectives identify and tackle gaps in current scholarly literature and practical applications, offering novel

solutions for enhancing transparency, reliability, and interpretability of CNNs. Combining novel algorithms, evaluation metrics, and empirical studies sets the groundwork for new model performance and interpretability benchmarks.

- We propose a sample selection mechanism that employs meta-learning via reinforcement learning to optimise CNN performance. This approach filters out samples that could contribute to model overfitting and bias. Our empirical results demonstrate that this method enhances model performance in various image classification tasks, leading to increased model accuracy and robustness.
- We propose a sample selection technique that incorporates Monte Carlo dropout to the CNN training and filters samples with higher uncertainty during evaluation. We extend this technique to design an active learning framework that forwards the uncertain samples for expert labelling. We demonstrate the effectiveness of this technique on various classification tasks, showing that it can reduce the labelling cost, enhance credibility, and improve the model's performance.
- We introduce **AD**aptive **VIS**ual **E**xplanation (ADVISE), a novel explainability technique that employs adaptive bandwidth kernel density estimation to quantify the relevance of each unit in the feature map. This method is designed to offer more precise visual explanations by allocating a relevance score to each unit of the feature map based on the predicted class.
- We introduce Unique Class Activation Mapping (UniCAM), a novel visual explanation approach explicitly designed to explain and visualise the features distilled during the KD process. This technique seeks to provide an in-depth explanation of the foundational mechanisms in KD, delivering both visual and quantitative perspectives on the features learned during the KD process.
- We propose novel metrics to quantify visual explanations and to measure the similarities and relevance of features acquired during the KD process.

## 1.4 Thesis Outline

The principal objective of this thesis is to develop methodologies that contribute to the explainability of deep neural networks (DNNs) for computer vision applications. Specifically, the research concentrates on enhancing the performance of CNNs, quantifying the decision uncertainty of CNNs, and proposing more precise visual explanations that reveal the input-output relationship and highlight the most important features, regions, or channels of the input for the prediction. To evaluate the effectiveness of the proposed methodologies, the thesis also proposes evaluation protocols for explainability, which include quantitative metrics, qualitative analysis, and user studies.

**Chapter 1** provides a concise yet comprehensive introduction to the urgent need for responsible and transparent AI. It elaborates on the significance of designing

deep neural networks that are not only efficient but also ethical, accountable, and transparent. Key concepts such as responsible AI, ethical AI, and transparent AI are introduced to establish the framework for the subsequent discourse.

**Chapter 2** (see chapter 2 for details) provides an in-depth analysis of cutting-edge research focused on improving the transparency and interpretability of AI systems. This chapter reviews recent works on techniques for sample selection and uncertainty in decision-making, as well as techniques for Explainable AI (XAI).

**Chapter 3** (see chapter 3 for details) focuses on mitigating data bias by employing meta-learning techniques. This chapter proposes a novel methodology for selecting informative samples while avoiding those that may lead to model overfitting and bias, thereby contributing to the responsible use of AI.

**Chapter 4** (see chapter 4 for details) elaborates on methodologies that allow a CNN to refrain from making a decision when it is uncertain. Techniques for incorporating uncertainty into the decision-making process are discussed in detail. Explanations for the origins of the uncertainty, utilising existing XAI methods, are also provided.

**Chapter 5** (see chapter 5 for details) introduces and investigates novel visual explanation techniques for explaining the internal decision-making processes of CNNs. The focus is primarily on generating visual explanations that humans can easily interpret, thereby enhancing the model's transparency.

**Chapter 6** (see chapter 6 for details) proposes a novel visual explanation and metrics to interpret the process KD. The limitations of existing methods for explaining KD are discussed and solutions are proposed.

**Chapter 7** (see chapter 7 for details) discuss the methodologies proposed in previous chapters, offering a comparative analysis with existing solutions. It also outlines the implications of this work for future research, particularly the avenues it opens for developing more ethical and transparent AI systems. Ethical considerations and the impact of these methods on the broader domain of AI and computer vision are also discussed.

**Chapter 8** (see chapter 8 for details)provides an overview of the academic publications and scholarly contributions during this PhD study. It describes the methods used and the findings obtained.

# Chapter 2

## Literature Review

The field of computer vision has benefited greatly from the advancement of Convolutional Neural Networks (CNNs), which have enabled the accomplishment of tasks that were previously regarded as too challenging or impractical. However, as CNNs have become increasingly sophisticated, numerous challenges have emerged, notably in the areas of data scarcity and quality, model explainability and interpretability. This chapter provides a comprehensive overview of the related research studies categorising and examining these challenges, along with their approach to address the challenges. Furthermore, it aims to identify the relevant theories, methods, applications and gaps in the existing research, develop a theoretical framework and methodology for the research, and show how the research addresses a gap. This chapter is organised into three main sections: (1) informative sample selection (2) visual explainability methods, and (3) explaining knowledge distillation techniques.

## 2.1 Informative Sample Selection

The quality, diversity, and reliability of the data used to train a machine learning model, a Deep Neural Network, can significantly impact the model's performance, robustness, fairness, and generalisability. However, obtaining high-quality and correctly labelled data is costly and challenging. Moreover, some data samples may be more informative or representative than others, while some may be outliers, redundant, imbalanced, incomplete, or biased [23, 24, 25, 26].

Sample selection techniques generally rely on the principles of statistical learning, optimisation, and information theory. At the core, these methods aim to identify data instances that contribute the most to the performance of the model, which is often quantified using statistical measures like entropy, mutual information, or Bayesian uncertainty [27, 28]. In light of these theoretical foundations, researchers have developed various strategies to evaluate the relevance of samples using the commonly employed techniques such as uncertainty-based methods [26], active learning [29], and reinforcement learning [30].

Angelova et al. [31] proposed data pruning mechanism cleaning samples with noise. An initial classifier is trained using the complete training set $D$. Following this, the posterior probabilities $P$ for the labels of each sample are computed. The pruning process aims to filter out noisy or hard-to-classify samples and which was formally represented as:

$$D' = \{x \in D : I(x, P(x)) = \text{True}\} \tag{2.1}$$

Here, $I(x, P(x))$ serves as an inclusion condition, defined as:

$$I(x, P(x)) = \begin{cases} \text{True} & \text{if } P(x) \geq \text{threshold} \\ \text{False} & \text{otherwise} \end{cases} \tag{2.2}$$

This inclusion condition helps determine which samples are retained in the pruned dataset $D'$. While this method effectively removes samples that are presumably noisy, it may inadvertently discard samples that could be valuable for training the classifier. It may also introduce bias into the remaining dataset $D'$ and exhibits sensitivity to the choice of initial parameters, such as the model and the threshold employed for pruning.

Lapedriza et al. [24] investigated the significance of individual training examples by assessing their influence on a classifier's performance. They proposed a methodology to rank examples based on how much their absence affects the model's accuracy. This involved comparing the model's accuracy with the entire dataset against its accuracy after retraining without a specific example. To create a more efficient training subset that achieves superior accuracy, the proposed method prioritises examples whose removal results in a substantial decrease in accuracy. However, this approach has some challenges. Firstly, it necessitates repeated model training during selection, rendering it computationally demanding. Secondly, defining a universally applicable value function and determining the optimal parameters (subset size, initial classifier) could be challenging, as they can significantly influence the example selection process and the initial state of the classifier

In [32, 33], the authors propose new sample selection techniques leveraging classification uncertainty. Sensoy et al. [32] aim to enhance model robustness by highlighting samples near class boundaries. Song et al. [33] proposed a technique to adjust mini-batches based on recent prediction uncertainty. However, these approaches face challenges in creating or selecting auxiliary datasets to represent out-of-distribution samples, particularly in high-dimensional spaces like images.

While the aforementioned approaches offer valuable contributions to the field of sample selection, they also come with certain limitations. For instance, data pruning methods may introduce bias, overlook potentially informative samples or neglect samples that are informative yet challenging to classify. We proposed two novel methods to address these limitations and enhance the sample selection process: reinforcement learning (RL) using a meta-learning strategy and uncertainty quantification with active learning. Meta learning offers a dynamic framework for sample selection, enabling the model to evolve its selection strategy based on real-time feedback. In addition, uncertainty quantification and active learning could also serve as a powerful mechanism for sample selection.

### 2.1.1 Meta-learning

Meta-learning has found wide applicability in domains such as computer vision [34, 35] and natural language processing [36, 37], especially in applications that benefit from few-shot learning capabilities [38, 39]. The core idea is to train a model on many tasks, enabling it to quickly adapt to new, unseen tasks with minimal fine-tuning. The model, thus, not only learns the specifics of each task but also captures a higher level of abstraction that facilitates cross-task adaptability. This has particularly profound implications for informative training sample selection in DNNs, offering ways to dynamically adjust to the changing landscape of data distributions and complexities.

Building on the promising benefits of RL-based meta-learning approaches and DNNs, many studies have emerged, focused on improving both training efficiency and model performance [25, 26, 29, 30, 40, 41]. Advancements in bio-inspired algorithms and reinforcement learning have led to proposing variety of methods for dynamic sample selection, further solidifying the utility of meta-learning in the domain. Chen et al. [42], for instance, employed a RL-based transfer learning strategy to seamlessly adapt deep learning models to new tasks. Similarly, other approaches have utilised RL to automate curriculum design [43], improve adversarial learning for image production [44], and develop dynamic querying and labelling strategies [45].

Meta-learning typically employs a two-level training process. At the higher level, the meta-learner $M$ updates its parameters $\Theta$ based on the performance $P$ of the base learner $B$ across multiple tasks $T$:

$$\Theta_{new} = \Theta_{old} + \alpha \nabla P(B(T; \Phi); \Theta) \tag{2.3}$$

where $\Phi$ denotes the task-specific parameters for the base learner, $\alpha$ is the learning rate for meta-learning, and $\nabla$ represents the gradient. This gradient provides the direction in which the task-specific loss function $L_T$ increases most rapidly with respect to $\Phi$. At the task level, the base learner updates its parameters $\Phi$ based on the task-specific loss $L$:

$$\Phi_{new} = \Phi_{old} - \beta \nabla L(T; \Phi) \tag{2.4}$$

Meta-learning, which enables learning samples for multiple tasks, can be a valuable technique for informative sample selection. It can help a DNN model learn to distinguish between informative and non-informative samples, which can help overcome the challenges of informative sample selection. However, despite the advancements in RL-based meta-learning for optimising various aspects of DNNs, the specific challenge of training sample selection remains inadequately addressed. Current techniques of sample selection often rely on static metrics or thresholds, which can introduce bias, lead to overfitting, or miss out on informative samples crucial for model performance. This absence of a dynamic, adaptive approach to sample selection has left a noticeable gap in the literature, underscoring the need for a more robust strategy.

In this context, the emerging synergy between RL-based meta-learning and DNNs presents a timely opportunity to revolutionise various facets of machine learning, one

of which is sample selection. Meta-learning, with its intrinsic adaptability to diverse tasks and data types, offers the potential to dynamically and intelligently manage the sample selection process. When coupled with the real-time decision-making capabilities of RL, this approach holds a promising path to mitigate the common challenges like bias and overfitting. Thus, an RL-based meta-learning framework could pave the way for more adaptive and effective sample selection strategies in DNNs, setting the stage for more robust and reliable models

### 2.1.2 Uncertainty quantification

Deep neural networks are increasingly used to make decisions in various domains, some of which have high stakes for human lives. These domains require high accuracy and robustness from the models, as well as a clear indication of the confidence level of their predictions. For example, in medical diagnosis, self-driving cars, and natural disaster management, the consequences of wrong predictions can be catastrophic. Therefore, it is crucial to provide these systems with a way of estimating uncertainty in their predictions so that practitioners can make more reliable and safe decisions.

Uncertainty can be broadly categorised into two: epistemic (model uncertainty), and aleatoric (data uncertainty) [46, 47]. Epistemic uncertainty is caused by the imperfect model itself. It reflects the lack of knowledge about the data and can therefore be an important cue for identifying whether we see an already known concept, or not. In other words, epistemic uncertainty can be reduced as we acquire more training examples. In contrast, aleatoric uncertainty arises from the inherent randomness or noise within the data, such as measurement errors or natural variations.

Quantifying uncertainty, an important problem in many domains, has inspired various novel techniques such as Bayesian neural networks (BNNs), Monte Carlo dropout, and ensemble methods [48, 49, 50]. These techniques enhance the performance and reliability of machine learning models in various domains where data could be scarce or noisy. Moreover, they can provide useful information for tasks that require high reliability, such as medical diagnosis or autonomous navigation. However, implementing various uncertainty methods such as BNNs [48] are often impractical for real-time applications, as they require a large amount of memory and computation.

Epistemic uncertainty arises from our lack of knowledge about the model prediction. In the Bayesian approach, epistemic uncertainty is captured by the posterior distribution over the model parameters, which reflects our updated beliefs after observing the data. Let $D = \{X, Y\} = \{(x_i, y_i)\}_{i=1}^{N}$ represent the dataset, where $x_i \in \mathbb{R}^D$ are the inputs and $y_i \in \{1, ..., C\}$ are the corresponding classes, with $C$ denoting the total number of classes. The objective is to optimise the parameters, $\Theta$, of the function $y = f^{\Theta}(x)$ to produce the desired output.

The Bayesian approach defines a model likelihood as $p(y|x, \Theta)$, which is a softmax likelihood for classification tasks:

$$p(y = c|x, \Theta) = \frac{exp(f^c(x))}{\sum_{c'} exp(f^{c'}(x))}, \tag{2.5}$$

where $p(y = c|x, \Theta)$ is the probability of the output $y$ being class $c$ given the input $x$ and the model parameters $\Theta$, $f^c(x)$ is the the function to produce the desired output of the class $c$ given the input $x$, and $\sum_{c'} exp(f^{c'}(x))$ is the normalisation factor that ensures the probabilities sum up to one over all possible classes. Given the observed data, we can update our beliefs about the model parameters using the posterior distribution, which is obtained by applying Bayes' theorem:

$$p(\Theta|X, Y) = \frac{p(Y|X, \Theta)p(\Theta)}{p(Y|X)}. \tag{2.6}$$

where $\Theta$ is the set of model parameters, $X$ is the matrix of input features, and $Y$ is the vector of output labels.

However, computing the posterior predictive distribution $p(\Theta|X, Y)$ is often intractable for complex models and large datasets. Various approximation methods such as variational inference (VI), Markov chain Monte Carlo (MCMC), and Monte Carlo (MC) dropout have been proposed to estimate posterior predictive distribution.

Variational inference tries to find a tractable distribution $q(\Theta)$ approximating the true posterior $p(\Theta|X, Y)$ using the metric of Kullback-Leibler (KL) divergence [51]. The KL divergence measures how much information is lost using $q(\Theta)$ instead of the true posterior $p(\Theta|X, Y)$. The main objective is to optimise the variational parameters and minimise the KL divergence, which is equivalent to maximising the evidence lower bound (ELBO) [52], formulated as:

$$\text{ELBO}(q) = \mathbb{E}_q[\log p(X, Y, \Theta)] - \mathbb{E}_q[\log q(\Theta)], \tag{2.7}$$

The primary advantages of VI lie in its scalability and suitability for large datasets. However, VI often necessitates assumptions about the posterior's form and may underestimate the model's uncertainty.

MCMC is another method to approximate the posterior predictive distribution. It generates a random sample sequence that follows the posterior distribution $p(\Theta|X, Y)$. The samples can be used to calculate the mean, variance, or other posterior statistics. The main idea of MCMC is to build a Markov chain that converges to $p(\Theta|X, Y)$ as its stationary distribution. This means that the longer we run the chain, the closer the distribution of the states will be to $p(\Theta|X, Y)$.

The Markov chain is defined by a transition kernel $K(\omega, \omega')$, which gives the probability of moving from state $\omega$ to state $\omega'$ in one step. The transition kernel must satisfy this equation for any pair of states $\omega$ and $\omega'$:

$$p(\omega|X, Y)K(\omega, \omega') = p(\omega'|X, Y)K(\omega', \omega) \tag{2.8}$$

where $p(\omega|X,Y)$ is the probability of the state $\omega$ given the data $X$ and $Y$, and $K(\omega'|\omega)$ is the probability of moving from $\omega$ to $\omega'$. Note that each state $\omega$ is a possible value of the model parameters $\Theta$. After the Markov chain reaches convergence, we can collect samples $\{\omega^{(1)}, \omega^{(2)}, \ldots, \omega^{(N)}\}$ from the chain and use them to estimate quantities of interest from the posterior distribution, such as:

$$\text{Mean} \approx \frac{1}{N}\sum_{i=1}^{N}\omega^{(i)} \qquad \text{Variance} \approx \frac{1}{N-1}\sum_{i=1}^{N}(\omega^{(i)}-\bar{\omega})^2 \tag{2.9}$$

where $N$ is the number of samples, and $\bar{\omega}$ is the sample mean. MCMC is a powerful method for estimating complex and multimodal distributions, but it also has some drawbacks. It can be computationally expensive and time-consuming, especially for large datasets and high-dimensional problems. It also requires checking the convergence and mixing of the samples, and choosing appropriate proposal distributions or transition kernels.

To solve the challenges of MCMC in DNN, Monte Carlo dropout (MC dropout) was introduced, and it uses dropout as a regularisation term to compute the prediction uncertainty [53]. It applies dropout at inference time to compute the uncertainty of the predictions. Dropout is a method that randomly sets some weights of a neural network to zero during training to prevent overfitting. MC dropout uses dropout as a way of sampling from a distribution over the weights, which creates a distribution over the outputs. MC dropout is given by:

$$p(Y|X,D) \approx \frac{1}{T}\sum_{t=1}^{T}p(Y|X,\Theta_t) \tag{2.10}$$

where $p(Y|X,D)$ is the posterior predictive distribution of the output $Y$ given the input $X$ and the data $D$, $\Theta_t$ is the is the set of weights for the $t^{th}$ dropout sample, and $T$ is the number of iterations. MC dropout is easy to implement and scalable, but it requires choosing appropriate dropout rates and the number of samples.

While epistemic uncertainty reflects the lack of knowledge about the true model prediction, aleatoric uncertainty captures the inherent randomness or variability in the data. Aleatoric uncertainty can be estimated using probabilistic models that model the distribution of possible outcomes given the input variables. For instance, heteroscedastic regression models can account for the noise that depends on the input, thereby providing a more refined measure of uncertainty [54]. Addressing aleatoric uncertainty is important for assessing the risk and robustness of machine learning models and designing optimal decision strategies that balance expected reward and uncertainty. This is especially relevant in fields like robotics or clinical research, where sensor noise or patient variability can affect the model's performance and safety.

Uncertainty quantification is useful for many reasons. It can improve the performance, reliability, and safety of a DNN model. It can also help the decision-makers to know the possible outcomes and risks, prevent bias and overconfidence, and support better decision-making. In chapter 4, we will demonstrate how active learning

benefits from uncertainty estimation, as it enables the model to select informative samples for labelling and achieve faster and more effective training.

### 2.1.3 Active learning

Active learning (AL) emerges as a paradigm in machine learning that tackles the challenge of limited labelled data. Unlike traditional supervised learning methods which rely on large, pre-labelled datasets, AL focuses on achieving high-performance models with fewer labelled samples. It achieves this by interactively querying a user or an annotation source to label the most informative data points for learning.

AL approaches can be categorised through multiple lenses depending on the specific focus or application. Settles et al. [55] classify AL approaches based on the level of interaction between the learner and the oracle (the entity providing labels or feedback). They distinguish three primary strategies: membership query synthesis [56], stream-based selective sampling [57], and pool-based sampling [58]. Another commonly used criterion centres on query strategy, where methods are broadly grouped into uncertainty-based, diversity-based, and model-change-based categories [58, 59]. These two strategies are not mutually exclusive, but rather complementary. The level of interaction determines **how** the learner obtains the labels or feedback from the oracle, while the sample selection focuses **what** criterion the learner uses to select the most informative data points to query the oracle. Depending on the specific problem and setting, different combinations of these strategies can be used to achieve effective and efficient AL approaches.

In the membership query synthesis scenario [57, 60], the active learner generates its instances from the input space and queries the oracle for the true labels. This strategy allows the learner to explore the regions of the input space that are most relevant or informative for the learning task rather than relying on existing data. For example, if the task is to classify images of animals, the learner could create a synthetic image of a hybrid animal and ask the oracle what kind of animal it is. Membership query synthesis can be useful when the existing data is scarce, noisy, or unrepresentative of the true distribution. However, a major drawback of this scenario is that it can artificially generate instances that are impossible to reasonably label or do not reflect the real-world data distribution.

In stream-based selective sampling, the active learner receives a stream of instances from the input space and decides whether to query the oracle for the true labels [61]. This strategy allows the learner to filter out the instances that are easy or redundant for the learning task and only focus on the difficult or informative ones. For instance, in surveillance videos, if the task is to detect faces in videos, the learner could ignore the streams that clearly contain or do not contain faces and only query the oracle to label the ones that are occluded or blurred. Stream-based selective sampling can be useful when the data is generated continuously or online, and labelling all instances is not feasible. However, a drawback of this scenario is that it requires

a fast and reliable oracle response, which may not be available in some domains or applications.

The pool-based sampling strategy is a commonly used active learning strategy where the model can access a large pool of unlabelled instances from the input space and select a subset to query the oracle for their labels [62]. This strategy allows the learner to choose the most informative or diverse instances for the learning task rather than selecting them randomly or sequentially. For example, if the task is to segment objects in images, the learner could select images that contain different types or shapes of objects and avoid images that are empty or cluttered. Pool-based sampling can be useful when the data is available in advance, but labelling all instances is too costly or time-consuming. However, a drawback of this scenario is that it requires a large amount of unlabelled data, which may not be easy to obtain in some domains or applications.

The AL approaches based on query strategies mainly aim to optimise the trade-off between exploration and exploitation when querying the oracle. These techniques can often be combined with the aforementioned strategies to enhance the efficiency of the learning system. In uncertainty-based methods [63, 64], the learner queries the oracle for the most uncertain instances, thereby attempting to refine the model's decision boundaries. Diversity based [65, 66, 67] methods focus on selecting samples that cover a broad range of feature space, ensuring a representative understanding of the data distribution. Model change [68, 69] approaches prioritise samples that would cause a significant alteration in the current model, optimising for more rapid convergence to a robust model. These sample selection techniques serve as underlying methods that can be employed in conjunction with membership query synthesis, stream-based selective sampling, or pool-based sampling to tailor the active learning process to specific requirements and constraints.

Recently, uncertainty-based query strategies with significant contributions from Bayesian Neural Networks (BNNs) and Monte Carlo dropout have gained popularity [49, 53]. Many uncertainty-based query strategies are derived from pool-based AL techniques and use different methods to estimate uncertainty, such as:

- **Entropy** [70, 71]: This measures the amount of information contained in a probability distribution. A high entropy indicates a high level of uncertainty. For a discrete distribution $y$, the entropy is defined as

$$H(y) = -\sum_i p(y_i) \log p(y_i), \tag{2.11}$$

  where $p(y_i)$ denotes the model's predicted probability for each class $y_i$.
- **Margin** [72, 73]: This measures the difference between the the most likely class $y_1$ and the second most likely class $y_2$ predicted by the model. A low margin indicates a high level of uncertainty. Formally,

$$M(x) = p(y_1|x) - p(y_2|x), \tag{2.12}$$

where $p(y_1|x)$ and $p(y_2|x)$ refer to the model's predicted probabilities for the first and second most likely classes, respectively.

- **Least Confidence** [74]: This measures the confidence of the model in its most likely prediction $y_i$. A low confidence in the most likely prediction indicates a high level of uncertainty. The confidence is measured by $max_i\ p(y_i|x)$, and the uncertainty is defined as:

$$LC(x) = 1 - \max_i p(y_i|x), \tag{2.13}$$

where $\max_i\ p(y_i|x)$ refers to the maximum predicted probability across all classes.

- **Bayesian Active Learning by Disagreement (BALD)** [75]: This measures the mutual information between the model's predictions and its parameters. A high mutual information indicates a high level of uncertainty. For a probabilistic model with parameters $\Theta$ and a data point $x$, the BALD score is defined as

$$B(x) = H(\mathbb{E}_\Theta[p(y|x,\Theta)]) - \mathbb{E}_\Theta[H(p(y|x,\Theta))] \tag{2.14}$$

where $H$ is the entropy function and $\mathbb{E}$ is the expectation operator.

- **Monte Carlo dropout** [63]: This uses dropout at both training and testing stages to obtain stochastic predictions from the model. A high variance of the predictions indicates a high level of uncertainty. For a model with dropout rate $p$ and a data point $x$, the MC Dropout score is defined as

$$MCD(x) = \frac{1}{T}\sum_{t=1}^{T} f_t(x)^2 - \left(\frac{1}{T}\sum_{t=1}^{T} f_t(x)\right)^2 \tag{2.15}$$

where $f_t(x)$ is the prediction of the model with dropout mask $t$ and $T$ is the number of stochastic forward passes (iterations).

- **Generative Adversarial Active Learning** [76]: Here, a generative model $G$ is trained alongside the main predictive model $PM$. The generative model generates samples $x'$, and the predictive model's uncertainty on these samples is measured using

$$U(x') = 1 - \max_i p(y_i|x'; PM), \tag{2.16}$$

where $\max_i p(y_i|x'; PM)$ represents the maximum predicted probability when input $x'$ is fed into model $PM$. A high discrepancy between the model's predictions on real and synthetic data indicates a high level of uncertainty.

Uncertainty-based active learning offers robust mechanisms to measure the informativeness and diversity of samples, thereby allowing for efficient labelling efforts [63]. However, the computational cost of uncertainty estimation in deep models, particularly using Bayesian methods, poses challenges for scalability in large-scale applications [46]. Furthermore, the fidelity of uncertainty measures often depends on the approximation function, which may not always accurately capture the true model uncertainty. Despite these challenges, the potential for optimising data collection while ensuring high model performance makes uncertainty-based active learning an active

area of research. Ongoing work aims to address these issues by developing scalable algorithms, more accurate uncertainty measures, and novel techniques for balancing exploration and exploitation in the learning process.

Diversity-based query strategies aim to select data points that are diverse or representative of the unlabelled data distribution. The intuition is that by covering different regions of the feature space, the model can learn more generalise patterns and avoid overfitting to a specific subset of data. Some of the techniques used to estimate diversity are:

- **Cluster-based** [77]: This technique partitions the unlabelled data into clusters and selects one or more data points from each cluster. The clustering algorithm can be based on different criteria, such as distance, density, or graph structure.
- **Density-weighted** [78]: This technique assigns a weight to each data point based on its local density, often calculated via a kernel function, which is estimated by the number of neighbours within a certain feature space. The weight reflects the representatives of the data point, and the technique selects data points with high weights.

On the other hand, model-change-based query strategies aim to select data points that cause the most change in the model parameters or predictions. The intuition is that the model can learn more effectively and efficiently by choosing data points that have a large impact on the model. Some of the techniques used to estimate model change are:

- **Expected gradient length** [79]: This technique measures the expected length of the gradient vector of the loss function with respect to the model parameters, given a data point and its possible labels. The technique selects data points with large expected gradient lengths, assuming they will induce a large update on the model parameters.
- **Expected error reduction** [80]: This approach measures the expected reduction in the generalisation error of the model, given a data point and its possible labels. The technique selects data points that have large expected error reductions, assuming that they will improve the model performance on unseen data.
- **Expected information gain** [81, 82]: This measures the expected information gain about the model parameters, given a data point and its possible labels. The information gain between the posterior and prior distributions of the parameters is quantified using Kullback-Leibler divergence. The technique selects data points that have large expected information gains, assuming that they will reduce the uncertainty about the model parameters.
- **Model disagreement** [83]: In a multi-model context, this technique measures the disagreement among multiple models trained on different subsets of labelled

data, given a data point and its possible labels. Different metrics, such as vote entropy, average Kullback-Leibler divergence, or variance, can measure the disagreement. The technique selects data points that have high disagreement scores, assuming that they will resolve the conflicts among the models.

In addition to the query strategies and the levels of interaction, another aspect of active learning is using Query-by-Committee (QBC), a general framework where the learner consists of a committee of models that vote on the labels of the instances. The learner queries the oracle for the instances with the highest disagreement among the committee members, as they are likely to be informative and reduce the learner's uncertainty. The QBC can be applied to any of the sample selection criteria and the levels of interaction, depending on how the committee of models is formed and how their disagreement is measured. For example, QBC can be used in stream-based selective sampling by receiving a stream of instances and querying the oracle for the ones with the highest disagreement among the committee members [61]. QBC is a powerful and flexible framework that can capture different aspects of informativeness and diversity in active learning.

Uncertainty sampling is a powerful tool for selecting the most informative samples to label and train a deep neural network in active learning and improve the model's performance with a small labelled training set. It can also be used to design a DNN model that will say "I do not know" when it faces uncertain or out-of-distribution samples and reduces the risk of biased decisions. We demonstrate how uncertainty-based active learning can help enhance the performance of DNN models in real-world applications that face challenges such as data scarcity and lack of expert annotators. Furthermore, we employ existing visual explanation tools to inspect the sources of model uncertainty and suggest mechanisms to reduce such uncertainties, thereby contributing to more responsible and explainable AI practices.

## 2.2 Visual Explainability

While mitigating data bias through adaptive sample selection and estimating uncertainty are instrumental in creating responsible AI models, it does not fully unveil the opaque nature of the models. Explainability in Artificial Intelligence is growing rapidly as a critical dimension to making the decision-making process of Deep Neural Networks transparent, ethical, and accountable. Various methodologies, ranging from local interpretable model-agnostic explanations to natural language justifications, have been proposed to decode the complex operations of these networks. However, our focus within this vast terrain of options is *Visual Explainability* in CNN architectures. This approach leverages visual cues and representations to intuitively explain the model's decision-making process, thereby simplifying complex mechanisms into simple and easier to understand. This notion aligns well with our research objective of optimising the model's performance and making its decisions more interpretable to a broader audience.

Visual explainability techniques play a crucial role in interpreting the internal decision-making processes of CNNs. This approach encompasses pinpointing relevant regions in input images, generating descriptive natural language captions, synthesising exemplary images, or visualising internal model states. Visual explainability can be systematically classified into four primary categories [84]: feature visualisation, class activation mapping, perturbation analysis, and integrated gradients. Each category represents a distinct technique that aims to explain the model's decision-making process.

### 2.2.1 Feature visualisation

Feature visualisation serves as a main technique of visual explainability, primarily focused on decoding the learned features or patterns within a CNN [84, 85, 86, 87]. Feature visualisation involves generating synthetic images that maximally activate specific neurons or layers in the network. The method generally employs gradient ascent to modify an input image in such a way that it maximally activates a chosen neuron or layer in the network. Feature visualisation methods are formulated as an optimisation problem as follows:

$$\max_{I} A_{l,n}(I) + \lambda R(I) \tag{2.17}$$

where $A_{l,n}(I)$ represents the activation of the $n$-th neuron in layer $l$ for the image $I$, and $R(I)$ is a regularisation term controlled by $\lambda$ to enforce image smoothness or other desired properties. The optimisation iteratively updates the image $I$ using gradient ascent to increase $A_{l,n}(I)$.

The advantages of feature visualisation lie in its power to offer rich insights into the internal representations that CNNs acquire during their training. This method can unveil the level of abstraction and the types of features that a model perceives as significant. However, there are notable drawbacks as well. Among them is the difficulty in interpreting the synthetic images generated, especially when dealing with complex CNN architectures with multiple layers and neurons. The images may be rich in information but overwhelming or ambiguous to the human, making it challenging to drive meaningful and understandable insights.

### 2.2.2 Integrated gradients

Integrated Gradients (IGs) are gradient-based methods for visual explainability that aims to explain the model's prediction to each input feature [19]. They achieve this by gradually transforming the baseline input into the actual image. Throughout this transformation, IGs track how much each part of the image influences the model's prediction. This allows IGs to assign importance scores to different image regions, highlighting the features that most significantly contribute to the final decision.

IGs have several advantages over other gradient-based methods. First, IGs satisfy two desirable properties for attribution methods: sensitivity and implementation invariance [19]. Sensitivity means that if a feature does not affect the model's output, its attribution should be zero. Implementation invariance means that if two models have functionally equivalent outputs, their attributions should be identical. Second, IGs can be applied to any differentiable model without modifying the model's architecture or training process. Third, IGs can quantify the contribution of each feature in the input to the model's prediction, which can help identify relevant and irrelevant features.

However, IGs also have some drawbacks that limit their applicability and interpretability. First, IGs are computationally expensive, requiring multiple gradients for each pixel value (or feature) in the image. Second, IGs suffer from gradient saturation, meaning features with high activation values may have low gradients and thus low attributions. This can lead to misleading or incomplete explanations, especially for models with nonlinear activations like ReLU. Third, IGs depend on the choice of the baseline input, which can affect the quality and consistency of the attributions [88]. There is no clear guidance on selecting an appropriate baseline input for a given model or task.

### 2.2.3 Class activation mapping (CAM)

Class Activation Mapping (CAM) is commonly used technique in visual explainability that seeks to identify the discriminative regions within an image, contributing significantly to the prediction of a particular class [89, 90, 91, 92, 93, 94]. In formal terms, the core operation in CAM involves computing the class-specific importance for each spatial location in an image and is often represented as:

$$\text{CAM} = \sum_k w_k \times A_k \tag{2.18}$$

where $A_k$ represents the activation map at a particular location and $w_k$ is the corresponding weight. The weights are then obtained through various attention mechanisms, and a ReLU operation is typically applied to filter out negative activations.

Multiple advancements in CAM techniques have been developed over the years. Grad-CAM [90] refines CAM by incorporating class-specific gradients. Grad-CAM++ [91] and Score-CAM [92] extend this by employing a weighted combination of positive partial derivatives and eliminating gradient dependence, respectively. Layer-CAM [93] and Zoom-CAM [94] introduce nuanced methods to improve granularity and small-object discernibility. The main difference between these methods is how they compute the weights $w_k$.

Grad-CAM [90] is a generalisation of CAM [89] that can be applied to any CNN, regardless of its architecture or the presence of a global average pooling layer. Grad-CAM computes the weights $w_k$ as the global average of the gradients of the class

score with respect to the activation maps:

$$w_k = \frac{1}{Z} \sum_{i,j} \frac{\partial y_c}{\partial A_{k,i,j}} \tag{2.19}$$

where $y_c$ is the score for class $c$, $A_{k,i,j}$ is the activation of unit $k$ at spatial location $(i, j)$, and $Z$ is a normalising constant. Grad-CAM then applies a ReLU operation to filter out negative activations.

Grad-CAM++ [91] is an extension of Grad-CAM that incorporates higher-order partial derivatives and weights each pixel contribution based on its relative importance. Grad-CAM++ computes the weights $w_k$ as:

$$w_k = \sum_{i,j} \alpha_{k,i,j} \frac{\partial y_c}{\partial A_{k,i,j}} \tag{2.20}$$

where $\alpha_{k,i,j}$ is a weight coefficient that depends on the first and second derivatives of $y_c$ with respect to $A_{k,i,j}$. Grad-CAM++ also applies a ReLU operation to filter out negative activations.

Score-CAM [92] is another extension of CAM that eliminates the dependence on gradients and uses the activation map values directly. Score-CAM computes the weights $w_k$ as:

$$w_k = \frac{\text{Score}(A_k)}{\sum_{k'} \text{Score}(A_{k'})} \tag{2.21}$$

where $A_k$ represents the activation map for channel $k$ in the final convolutional layer, $A_k^{masked}$ refers to a mask derived from the original activation map ($A_k$), $Score(A_k^{masked})$ denotes the class score obtained after applying the mask ($A_k^{masked}$) to the network and calculating the output for the target class. The summation in the denominator considers the class scores obtained using masks derived from all activation map channels ($k'$).

Layer-CAM [93] and Zoom-CAM [94] are two recent methods that improve the quality and granularity of CAM, especially for small objects. They both use the same formula as Grad-CAM to compute a class activation map, but they introduce novel techniques to select or enhance the activation maps. Layer-CAM defines a layer selection criterion based on the class-specific activation entropy, which measures the uncertainty of the activation distribution and then it selects the layer with the lowest entropy as the optimal one for CAM. Zoom-CAM introduces a zooming operation that magnifies the activation maps by a factor of $s$ before applying CAM as follows:

$$\text{Zoom-CAM} = \text{ReLU}(\sum_k w_k \times \text{Zoom}AF_k, s)) \tag{2.22}$$

where $\text{Zoom}(A_k, s)$ is the zoomed activation map obtained by bilinear interpolation. The main difference between Layer-CAM and Zoom-CAM is that Layer-CAM focuses on selecting the best activation layer for CAM, while Zoom-CAM focuses on enhancing the activation maps by zooming. Both techniques can improve the quality and

granularity of CAM compared to previous methods.

Despite their effectiveness in revealing how a CNN makes class-based predictions, CAM methods including Grad-CAM and its variants have limitations. These approaches primarily focus on individual activations, potentially neglecting the complex interactions between features. To address these shortcomings, we propose a novel explainability method, ADVISE [95] presented in chapter 5, which leverages activation map values directly and incorporates feature relevance, aiming to provide a more comprehensive understanding of the factors influencing CNN predictions.

### 2.2.4 Perturbation analysis

Perturbation analysis takes a different approach to visual explainability, focusing on manipulating the input and examining the subsequent changes in model predictions. Unlike CAMs techniques that offer insights based on the internal workings of a neural network, perturbation analysis is more concerned with external validation, aiming to explain the model's decision-making process through alterations in input data. Within this framework of external validation, several techniques have been proposed to create perturbations and measure their impact.

There are several ways to create these changes (perturbations) in the image. One commonly employed technique is called occlusion [85, 87, 96]. Perturbation works by covering parts of the image with a mask, like blurring a section. We then observe how much the model's prediction changes when that area is hidden. This helps us understand which parts of the image are most important for the model's decision.Other techniques include LIME [97], meaningful perturbation [17], and real-time saliency [98, 99, 100]. These methods use different ways to change the image and measure the impact, giving us various perspectives on what's important in the image for the model.

LIME simplifies the complex model locally by creating a basic model around each image. It then compares the original model's prediction on a masked image with the prediction from this simpler model on the masked image. Meaningful Perturbation instead focuses on finding the best mask (hiding parts of the image) that maximally changes the model's prediction, considering limitations like mask size and smoothness. Real-time saliency methods take a different approach by training another neural network to predict the best mask that minimises the difference between the original and masked image predictions.

Perturbation analysis is a powerful tool to understand how a model focuses on specific areas or features in an image. It provides more detailed explanations compared to CAM family visual explanation techniques, which offer a more general perspective. However, this method can be time-consuming because it requires creating and testing many altered versions of the image. In addition, its effectiveness depends on the chosen method for modifying the image, which can impact its reliability across various scenarios.

While all the above explanation techniques for CNNs provide valuable insights, they have several limitations. A key limitation lies in their inability to quantify the importance of each unit within the activation maps. This hinders understanding how individual units contribute to the final model decision. To address this limitation, we propose ADVISE [95] (chapter 5), a method that quantifies the contribution of each unit, offering better model explainability. Furthermore, existing methods struggle to explain the Knowledge Distillation (KD) process, failing to explain which specific features the student model learns during KD.

## 2.3 Explaining Knowledge Distillation

Knowledge distillation (KD) is a technique that trains a student neural network by transferring knowledge from a trained teacher network. It has attracted considerable interest as a model compression technique, enabling the development of smaller and more efficient models while preserving performance [101, 102, 103, 104, 105, 106]. KD trains the student with the training data and the knowledge from the teacher model, which improves student performance [105, 107, 108, 109]. KD has shown its effectiveness in various domains, such as language models [110], image captioning [111], semantic segmentation [112], and object detection [113].

While there has been significant research on KD, few efforts have been made to explain the KD process. Raed et al. [114] proposed a framework to improve neural network interpretability through KD. Lee et al. [115] introduced an interpretable knowledge transfer method using principal component analysis and graph neural networks. Seunghyun et al. [116] developed a graph-based KD technique with a multi-head attention network. Cheng et al. [21] applied information theory to explain the effectiveness of KD. Xue et al. [22] proposed KDExplainer, a task-oriented model, to explain KD. However, these techniques don't explain the distilled features explicitly and don't quantify their contribution for the prediction. To address these limitations and enhance the transparency of KD, we introduce *UniCAM* (Chapter 6), a method specifically designed to explain the features learned through distillation.

# Chapter 3

## Improving Image Classification with Meta-learning and Sample Selection using Reinforcement Learning

Deep Neural Networks (DNNs) are powerful models that can learn from extensive datasets, but not every sample in these datasets are equally useful for effective training. Some samples may be redundant, out of distribution, or noisy, thus undermining the model's performance and reliability. On the other hand, some samples are more informative, diverse, or representative, thus helping the model to learn more efficiently. Informative Sample Selection (ISS) is a strategic approach that enhances learning by prioritising the most impactfull samples, especially when computational and time constraints are involved in training DNNs. The ISS is not just about filtering data; it is an indispensable part the DNN training process to improves the performance by prioritising the most informative samples.

The ISS has many advantages for DNN training, such as reducing the training time and cost, improving generalisation and robustness. However, ISS also poses many challenges, such as how to measure the informativeness of samples, how to balance the exploration and exploitation of data, and adapt the selection criteria as the model learns. Existing ISS methods often rely on heuristic rules or predefined metrics that may not capture the dynamic and complex nature of informativeness [26, 30, 32, 33].

Implementing ISS within deep learning frameworks presents a multifaceted challenges. It requires a robust method to assess the value of each sample and a flexible strategy to continuously refine the selection process as the model's training progresses. A sample that may seem peripheral at the beginning of the training may become essential as the model training progress. To cope with this complexity, we need an approach that can dynamically learn informativeness of samples rather than just applying a static filtering algorithm. Such an approach should be able to adapt the learning process itself based on the feedback and experience from each iteration of the same task or different tasks. This ability to adapt the selection process to the feedbacks and performance allows us to formulate the challenges as meta-learning [117], a technique that teaches models how to learn more effectively across various tasks by adjusting the learning parameters and strategies.

Meta-learning is a machine learning technique that learns how to learn from data and has achieved remarkable improvements in model performance across a range of challenging applications: from the complexities of image classification [29, 118], hyperspectral image classification [119] to the dynamic challenges of activity recognition [120]. These advances underscore the potential of meta-learning to address the

ISS challenges. Incorporating feedback and learned experiences to the ISS will help the model to prioritise the data that will effectively optimise their learning performance.

This chapter introduces a novel meta-learning based ISS algorithm that integrates RL into DNN training processes. Our goal is to create an ISS agent that learns from its experience and the DNN's feedback during each training iteration. This agent strategically selects samples to enhance DNN performance and prevent overfitting and bias. The "ISS agent" refers to this sample selection algorithm, formulated as a sequential decision-making problem. We particularly focus on Convolutional Neural Networks (CNNs), a powerful type of DNN for image classification tasks. Through interaction with the CNN and the dataset, the ISS agent learns to select the most informative samples for each iteration based on its accumulated experience. By incorporating RL, the ISS agent gains flexibility and adaptability as the CNN classifier learns and the informativeness of samples changes through time. The agent can leverage knowledge gained from previous experiences to quickly and efficiently adapt its sampling strategy for each sample over time. This results in improved generalisation and robustness compared to traditional statistical ISS methods.

We evaluate the proposed method on three image classification benchmark datasets: CIFAR-10 [121], MNIST [122], and Fashion-MNIST [123]. We compare it with an equivalent CNN model trained using classical methods (entire training dataset, stochastic gradient descent or variants). Classical training doesn't leverage the ISS agent to select informative samples, which can enhance the model's learning effectiveness and generalisability. Importantly, while tested on image classification, the ISS algorithm is versatile and applicable to various tasks with small modifications.

## 3.1 Proposed Method

We consider a meta-learning setting where we have access to a classification task with a set of classes $C$ and a dataset $D = \{x_j, y_j\}_{j=1}^{N}$, where $x_j$ is the input feature vector and $y_j \in C$ is the corresponding label. We assume the samples are drawn from a task-specific distribution. Our goal is to train a an ISS agent that can select informative samples from the pool of candidates for the classification task and use informative samples to fine-tune or train a CNN and achieve better performance on the training ($D^{train}$) and test ($D^{test}$) sets. The ISS agent utilises Q-learning to determine the value of actions within specific states throughout the model's training process. This allows it to continuously learn and adjust its sample selection strategy, prioritising informative samples that benefit both the current iteration and future ones based on current outcomes and past observations.

The proposed method (see Figure 3.1) consists of four main components: the **CNN**, the **ISS agent (RL)**, the **original dataset**, and the **new dataset**.

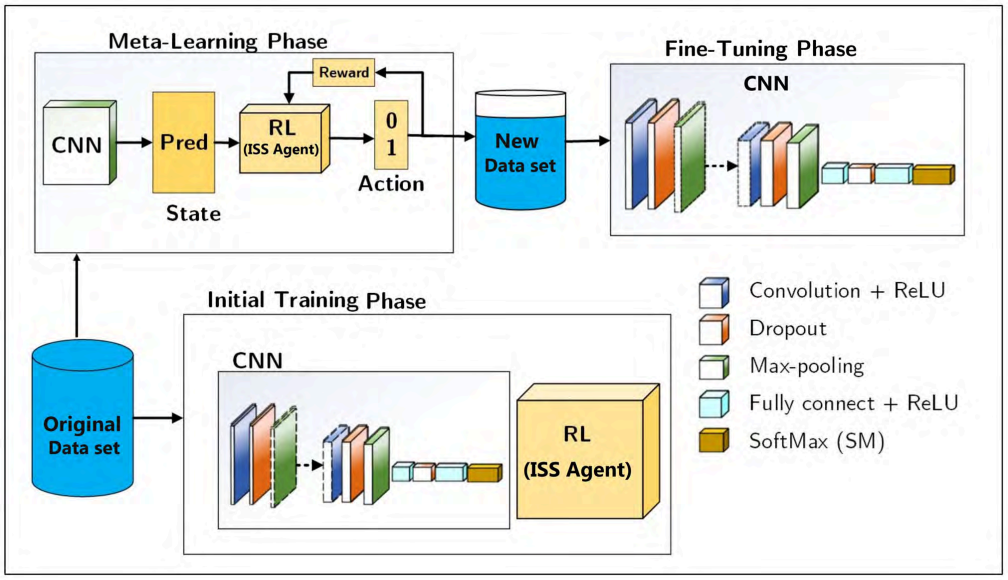



**Figure 3.1:** Architecture of the proposed method: CNN model, the ISS agent using Q-learning, an original dataset, and informative dataset.

- **The CNN**: A DNN that takes an input image and gives a probability distribution over the class prediction. In our setting, initially the CNN is trained on the entire dataset for few epochs. The CNN contains standard layers such as convolutional layers with ReLU activation, dropout for regularisation, max-pooling for feature dimension reduction, fully connected layers with ReLU, and a SoftMax layer for classification.
- **The ISS agent**: This module employs Q-learning, a model-free reinforcement learning algorithm, to learn the informativeness of data samples. The ISS agent receives predictions from the CNN as its input state and decides on actions that aim to retain or discard samples based on the reward feedback. The reward measures the contribution of each sample to the model's performance.
- **The original dataset**: The original dataset $D$ is a dataset which contains informative and non-informative samples from which the model begins its training.
- **The new dataset**: This is a new dataset $\hat{D}$ created by the ISS agent and selected from $D$. This dataset is expected to be more efficient for training the CNN, leading to improved performance and generalisation capabilities for fine-tuning or training the model.

Furthermore, the proposed method has three phases: **initial training**, **meta-learning**, and **fine-tuning**. These phases form a complete learning cycle that drives the model to achieve better performance in image classification tasks.

### 3.1.1 Initial training

The initial training phase is a preliminary stage where both the CNN model and the ISS agent are trained on the entire dataset for a few epochs to establish baseline performance and learn the fundamental features and characteristics of the classes. Simultaneously, the ISS agent, which is responsible for the Informative Sample Selection (ISS) algorithm, also begins to initialise its parameters. However, at this point, the ISS agent does not engage in sample selection, allowing the CNN model to develop a broad understanding of the data. This initial training helps mitigate the risk of the model being too influenced by the initial decisions of the ISS agent, which may not be fully informed.

To train the CNN model, we use cross-entropy loss as the learning objective and stochastic gradient descent as the optimisation method. To train the ISS agent, we use Q-learning as the learning algorithm and a random reward function as the learning objective. This pre-training equips both the CNN model and the ISS agent with the essential knowledge about the dataset and class distributions, preparing them for the next epochs.

### 3.1.2 Meta-learning

At the core of this work, the concept of meta-learning is to equip the CNN model with the ability to adapt rapidly with the informativeness for the target task and to design a model that can generalise well. To quickly adapt to the task and the dataset, we first train the CNN model $f_\phi$ and the ISS agent $\pi_\psi$ on $D$ for $k$ epochs, where $\phi$ and $\psi$ are the parameters of the CNN model and the ISS agent, respectively.

For the rest of the $n - k$ epochs, we perform the meta-learning, where we set the CNN model to evaluation mode and evaluate the pool of candidates $D$. The results of the evaluation prediction are used as the state by the ISS agent. The ISS agent uses Markov Decision Process (MDP) to update its parameters and learn to select or reject each sample based on improving the CNN model's performance and the quality of selecting or rejecting each sample.

Q-learning learns the value of an action in a particular state, and selects the action that maximises the expected future reward. The state of the ISS agent is a vector of predictions for each sample in $D$, such as $TP$, $FP$, $TN$, and $FN$. The action is a binary vector $A = \{1, 0\}$ where 1 indicates the sample is selected and 0 indicates rejected. The reward is the improvement of the CNN model's performance after fine-tuning or training on the selected samples.

### ISS agent

The ISS agent is responsible for learning how to select the informative samples from the pool of candidates $D$ during each iteration. It learns informativeness from the prediction output of the CNN model on $D$ and learns a policy to maximise its reward depending on the actions taken on each state.

The ISS agent is trained using MDP [124] and updates the Q-table based on the expected reward. The MDP is represented by a tuple $M = (S, A, R, P, \gamma)$, where $S$ is the set of states, $A$ is the set of actions, $R$ is the reward function, $P(s'|s, a)$ is the probability of transitioning from state $s$ to state $s'$ using action $a \in A$, and $\gamma \in [0, 1)$ is a discount factor. The discount factor $\gamma$ is a hyper-parameter that helps adjust the significance of long-term or immediate rewards, with high values favouring long-term rewards and low values favouring immediate rewards.

The ISS agent learns to select and reject samples and stores the Q-values for each state-action pair, and updates the Q-table according to the MDP. During each iteration, the ISS agent learns a policy by comparing the quality of past actions and immediate rewards using the Bellman equation as follows:

$$Q_t(s, a) \leftarrow Q_{t-1}(s, a) + \alpha[r + \gamma \max_{a' \in A} Q(s', a') - Q_{t-1}(s, a)] \tag{3.1}$$

where $\alpha$ is the learning rate, $r$ is the immediate reward, and $Q_{t-1}(s, a)$ and $Q(s', a')$ represent the Q-values for the action taken to the previous and new states, respectively. The Q-learning algorithm iterates over all possible state-action pairs and is formulated as follows:

- **State:** The state is represented by a vector of pairs of the true label and the predicted label for each sample in $D$, as $(y_j, f_\phi(\mathbf{x}_j))$. The state is represented as follows:
$$s = [(y_1, f_\phi(\mathbf{x}_1)), (y_2, f_\phi(\mathbf{x}_2)), ..., (y_M, f_\phi(\mathbf{x}_M))] \tag{3.2}$$
- **Action:** The action is represented by a binary vector that indicates which samples are selected or rejected. The action is sampled as follows:
$$a = \pi_\psi(a|s) \tag{3.3}$$
where $\pi_\psi(a|s) \in \{1, 0\}$ is the ISS agent that determines which actions to take given a state.
- **Reward:** The reward is defined as a combination of the improvement of the CNN model's performance after fine-tuning or training on the selected samples, and the quality of selecting or rejecting each sample, based on the true label and the predicted label. The reward is computed as follows:
$$r = \Delta L(\phi) + \lambda Q(a, y, \hat{y}) \tag{3.4}$$
where $\Delta L(\phi)$ is the difference between the loss of the CNN model before and after fine-tuning or training on the selected samples, $\lambda$ is a hyper-parameter that

controls the trade-off between performance improvement and selection quality, and $Q(a, y, \hat{y})$ is a function that measures the quality of selecting or rejecting each sample based on the following:

$$Q(a, y, \hat{y}) = \begin{cases} +1 & \text{if } a = 1 \text{ and } y = \hat{y} \\ -1 & \text{if } a = 1 \text{ and } y \neq \hat{y} \\ +1 & \text{if } a = 0 \text{ and } y \neq \hat{y} \\ -2 & \text{if } a = 0 \text{ and } y = \hat{y} \end{cases} \tag{3.5}$$

where $a$ is the action, $y$ is the true label, and $\hat{y}$ is the predicted label for the sample. The positive-value reward indicates that the selected sample was successfully classified, or that the rejected sample was classified incorrectly. The negative-value reward indicates that the selected sample was classified incorrectly, or that the rejected sample was classified correctly. We slightly increase the negative reward when the agent rejects a correctly classified sample to avoid discarding informative samples. The reward values can be adjusted based on the associated risk to $FP$ and $FN$ so as to preserve more informative samples.

- **Exploration rate:** The exploration rate is a parameter that controls the trade-off between exploration and exploitation in the Q-learning algorithm. Exploration means selecting a random action, while exploitation means selecting the action with the highest Q-value. The ISS agent selects a random action with probability $\epsilon$, and selects the action with the highest Q-value with probability $1 - \epsilon$. The value of $\epsilon$ belongs to $(0, 1]$, where a value close to 1 indicates that the ISS agent will rely more on exploitation than exploration.
- **State transition:** Once the informative samples have been selected and added to the temporary training data, the CNN model is fine-tuned or trained using the new dataset. The trained or fine-tuned CNN is then set in evaluation mode in the next iteration of the meta-learning phase.
- **Episode:** The episode $e$ for a given iteration ends when all the training samples in the training set have been fed to the agent. At the end of the episode, the Q-table and the policy network are updated using the Q-learning algorithm and the reward function. This completes one iteration of the learning cycle.

The ISS agent can be seen as a meta-learner that learns to select informative samples for any given iteration. It can adapt to different learning dynamics, and select samples that are not only informative for the current iteration, but also for future iterations.

### 3.1.3 Fine-Tuning

During the fine-tuning phase, the CNN model is fine-tuned or trained on the new dataset. The fine-tuning phase follows the same steps as the meta-learning phase, except that the CNN model is switched to training mode and trained on the new dataset, and no Q-table update is performed. Then the fine-tuned CNN model is then evaluated on the test set, and it is switched to evaluation mode to enter meta-learning phase for the next iteration.

The CNN model is a Convolutional Neural Network (CNN) that takes an input feature vector and outputs a probability distribution over the classes. The parameters of the CNN model are initialised randomly, and updated by using gradient descent methods, which take a gradient step in the direction that minimises the loss function. We use cross-entropy loss, which measures the difference between the true label distribution and the predicted label distribution, and penalises incorrect predictions as follows:

$$L(\phi) = -\sum_{j=1}^{M} y_j \log p(y_j|x_j, \phi)$$

where $y_j$ is the true label and $p(y_j|x_j, \phi)$ is the predicted probability of the label for the sample $x_j$, and $M$ is the number of samples.

## 3.2 Experiments

In this section, we explain the results from various experimental setups designed to evaluate the proposed meta-learning framework for ISS, offering a balance between computational efficiency and generalisation ability across various datasets.

### 3.2.1 Experimental setup

We used PyTorch [125] to implement the CNN model and trained the pipeline on an NVIDIA GeForce RTX 3060 GPU. We employed a cross-entropy loss and stochastic gradient descent optimiser for the training process of the CNN model. The initial learning rates and momentum were set to 0.001 and 0.9, respectively. The learning rate was decayed by a factor of 0.1 every five epochs using the StepLR scheduler. Training was conducted with a mini-batch size of $B = 32$ for 25 epochs. We evaluated the performance of the model using $Precision$, $Recall$, $F1 - score$, and $Error - rate$ (see Eq. 3.6).

$$\begin{aligned} Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}, \\ F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \\ Error - rate = \frac{FP + FN}{TP + TN + FP + FN}. \end{aligned} \tag{3.6}$$

### 3.2.2 Dataset

We conducted experiments on three publicly available image classification datasets: CIFAR-10, MNIST, and Fashion-MNIST. These datasets are widely used and serve as standard benchmarks for evaluating the performance of various machine learning algorithms in computer vision.

- **CIFAR-10:** This dataset consists of 60,000 colour images of size 32 × 32 pixels, divided into ten classes, with 6,000 images per class. The dataset is split into a training set of 50,000 images and a test set of 10,000 images.
- **MNIST:** This dataset is a collection of 70,000 grayscale images of handwritten digits from 0 to 9, each size 28 × 28 pixels. The dataset is split into a training set of 60,000 images and a test set of 10,000 images.
- **Fashion-MNIST:** The Fashion MNIST dataset is a large, freely available fashion image database commonly used for training and testing various classification models. It comprises 70,000 grayscale images of size 28 × 28 pixels, divided into ten classes, with 7,000 images per class. The dataset is split into a training set of 60,000 images and a test set of 10,000 images.

We trained our model on the training sets and evaluated its performance on the test sets regarding classification accuracy on unseen data.

We chose these three diverse datasets to assess the proposed approach's robustness and generalisation capabilities across different image domains and complexity levels. The CIFAR-10 dataset enabled us to evaluate the model's ability to handle colour images, while MNIST and Fashion-MNIST provided insights into the model's performance on grayscale images and its generalisation beyond handwritten digits.

### 3.2.3 CNN architecture

The CNN model consists of convolution, dropout, pooling, activation, and fully connected layers. The CNN pipeline is trained to extract features from the input images and classify them into different classes. The details of the CNN model such as the layer type, kernel size, stride, padding, and output size for each layer is shown in Table 3.1. The purpose of this meta-learning experiment is to optimise the CNN parameters and enhance its performance. The resulting trained CNN model would then be ready for further evaluation and testing in subsequent experiments.

**Table 3.1:** The details of CNN architecture.

| | **Layer (type)** | **Output Shape** | **Filter Size** | **Kernel Size** | **Stride** | **Activation** |
|---|---|---|---|---|---|---|
| Input | Image | $[w, h, c]^{\dagger}$ | - | - | - | - |
| 1 | Convolutional | $[B, 10, 28, 28]$ | 10 | $5 \times 5$ | 1 | ReLU |
| | Dropout | $[B, 10, 28, 28]$ | - | - | - | |
| | Max-pooling | $[B, 10, 14, 14]$ | 10 | $3 \times 3$ | 2 | |
| 2 | Convolutional | $[B, 20, 10, 10]$ | 20 | $5 \times 5$ | 1 | ReLU |
| | Dropout | $[B, 20, 10, 10]$ | - | - | - | |
| | Max-pooling | $[B, 20, 5, 5]$ | 20 | $3 \times 3$ | 2 | |
| 3 | FC | $[B, 50]$ | - | - | - | ReLU |
| | Dropout | $[B, 50]$ | - | - | - | |
| 4 | FC | $[B, 10]$ | - | - | - | Softmax |

† For the CIFAR-10 dataset, $[w, h, c] = [32, 32, 3]$.
The $[w, h, c]$ values in the other datasets must be adjusted based on the input image size.

### 3.2.4 Meta-Learner (ISS agent)

In the context of our meta-learning, the meta-learner is composed of the CNN model set to evaluation mode and the ISS agent. The ISS agent is specifically tasked with learning informative samples. The state space for the ISS agent is determined by the output from the CNN as defined in Eq. 3.2 and its action space is expressed in Eq. 3.3.

We adopt a Q-learning algorithm optimised using Markov Decision Process (MDP)s for policy learning in Eq. 3.1. The ISS agent's reward function is configured based on the informativeness of each sample and the performance enhancement on the CNN (see Eq. 3.4). This setup aims to guide the optimisation of ISS parameters in a way that maximises the learning of informative samples. In the interaction between the ISS agent and the CNN, the ISS agent uses the output from the CNN model as its state, and the performance improvement of the CNN model as its reward.

### 3.2.5 Results

In this section, we present the results of the proposed ISS method. We discuss the main findings, and shortcomings of proposed method. We compared the performance of the CNN model trained with the ISS agent (**RL-CNN**) and the CNN model trained without the ISS agent (**Classical-CNN**). We used train and test accuracy, and generalisation to unseen classes to evaluate the proposed method. The ISS agent aims to improve the CNN model's ability to learn from the most informative samples, which leads to enhanced classification performance compared to the classical model trained without the ISS agent. We reported the learning curves, mean average accuracy, precision, recall, F1-score, and error rate as performance metrics to assess the impact of the meta-learning agent on the CNN training process.

Figure 3.2 and 3.3 show the learning curves for the MNIST, Fashion-MNIST, and CIFAR-10 datasets for training and validation, respectively. The performance gap between RL-CNN and Classical-CNN is significant and consistent across all three datasets during training and validation. RL-CNN converges faster and achieves higher accuracy than Classical-CNN, indicating that it can learn more effectively from the data. Moreover, RL-CNN avoids overfitting and maintains a stable performance throughout the training process. When the ISS agent selects informative samples, it can reduce the redundant and irrelevant data, improving the efficiency and robustness of the model. Furthermore, as non-informative samples are excluded, the ISS agent can reduce the computational cost and memory usage, making it more scalable and practical for large-scale image classification tasks.

Similarly, Table 3.2 shows the training and validation accuracy on the MNIST, Fashion-MNIST, and CIFAR-10 datasets. The CNN model trained with the ISS agent achieves higher accuracy than the model trained without the ISS agent on all three datasets. The accuracy gains are 2% to 4% on the MNIST and Fashion-MNIST datasets, respectively, and 10% on the CIFAR-10 dataset. The substantial

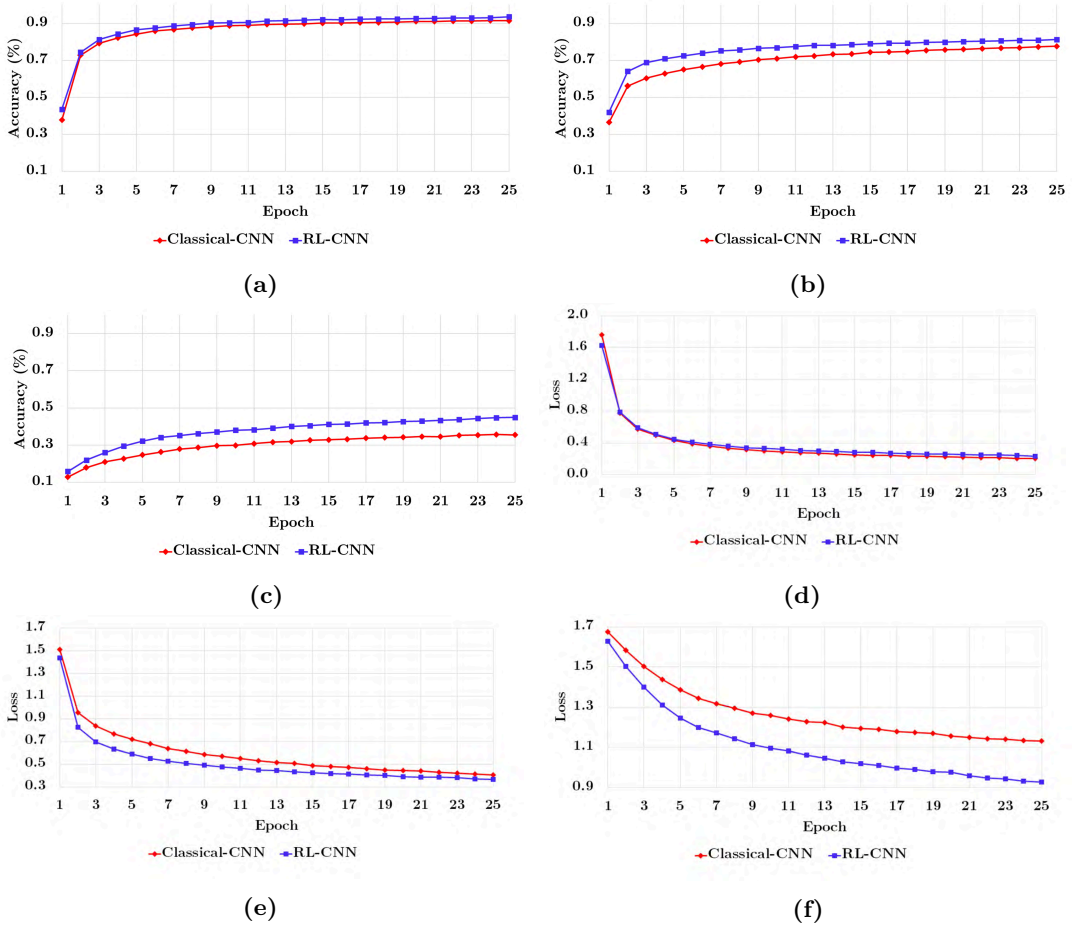



**Figure 3.2:** Learning curves of RL-CNN and Classical-CNN on the MNIST, Fashion-MNIST, and CIFAR-10 datasets. (a-c) Training accuracy, (d-f) Training loss. RL-CNN converges faster than Classical-CNN on all three datasets.

performance improvement on the CIFAR-10 dataset indicates that ISS agent helps the classifier to learn more discriminative features and avoid confusion between classes when the dataset has higher inter-class similarity. The CIFAR-10 dataset contains colour images of various objects, such as animals, vehicles, and plants, which are more complex and diverse than the grayscale images of digits or fashion items in the MNIST and Fashion-MNIST datasets. The ISS agent helps the classifier to focus on the most relevant samples from each class and ignore the irrelevant ones, resulting in better classification performance.

The evaluation of the test sets (see Table 3.3) also shows similar improvement trends. The CNN model trained with the ISS agent (RL-CNN) achieves higher *accuracy*, *precision*, *recall*, *F1-score*, and lower error rate than the CNN model trained without the ISS agent (Classical-CNN). The experimental results demonstrate that the ISS agent-based meta-learning can effectively select informative samples and avoid samples that could potentially cause bias and overfitting.

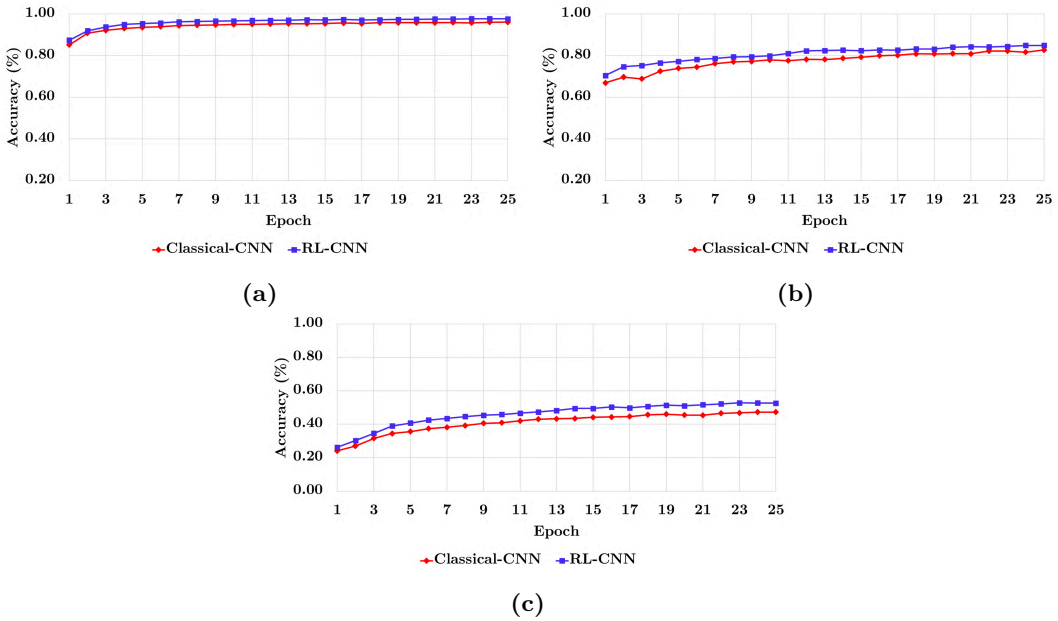



**Figure 3.3:** Validation progress curves of RL-CNN and Classical-CNN on the MNIST, Fashion-MNIST, and CIFAR-10 datasets.

**Table 3.2:** Comparison of the average classification accuracy (%) of the proposed strategy (RL-CNN) with the Classical-CNN training approach on MNIST, Fashion-MNIST, and CIFAR-10 datasets.

| Method | MNIST | Fashion-MNIST | CIFAR-10 |
|---|---|---|---|
| | Training accuracy(%) | | |
| Classical-CNN | 91.43 | 77.43 | 33.49 |
| RL-CNN | **93.53** | **81.23** | **44.99** |
| | Validation accuracy(%) | | |
| Classical-CNN | 96.10 | 80.73 | 47.28 |
| RL-CNN | **98.01** | **84.91** | **52.62** |

To illustrate the samples excluded from the training set, we present random images from MNIST and CIFAR-10 in Figure 3.4. The MNIST images excluded from the training set are difficult to identify, even for humans. For instance, images of 7 that resemble 1 are predicted as 1 and excluded from the training. Moreover, for the CIFAR-10 dataset, the model often struggles to distinguish between images with similar visual features (e.g., Cat vs. Dog and Deer vs. Horse), especially when noisy or containing multiple objects from different classes. The excluded images are either ambiguous, have multiple objects in them or difficult to distinguish one from the other class.

**Table 3.3:** Comparison of Classical-CNN and RL-CNN on MNIST, Fashion-MNIST and CIFAR-10 test sets.

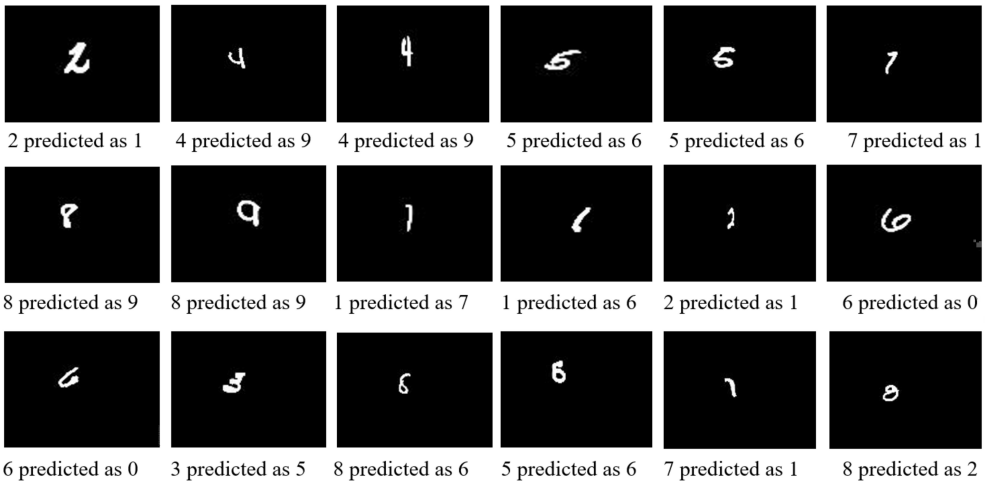
| Method | *Accuracy*(%) | *Precision* | *Recall* | *F1-Score* | *Error-rate*(%) |
|---|---|---|---|---|---|
| | | MNIST | | | |
| Classical-CNN | 97.2 | 0.97 | 0.97 | 0.97 | 0.484 |
| RL-CNN | **98.3** | **0.98** | **0.98** | **0.98** | **0.352** |
| | | Fashion-MNIST | | | |
| Classical-CNN | 83.0 | **0.83** | 0.83 | 0.82 | 3.41 |
| RL-CNN | **84.2** | **0.83** | **0.84** | **0.83** | **3.06** |
| | | CIFAR-10 | | | |
| Classical-CNN | 46.93 | 0.46 | 0.47 | 0.45 | 10.61 |
| RL-CNN | **52.60** | **0.52** | **0.53** | **0.51** | **9.47** |

### 3.2.6 Analysing time and space complexity

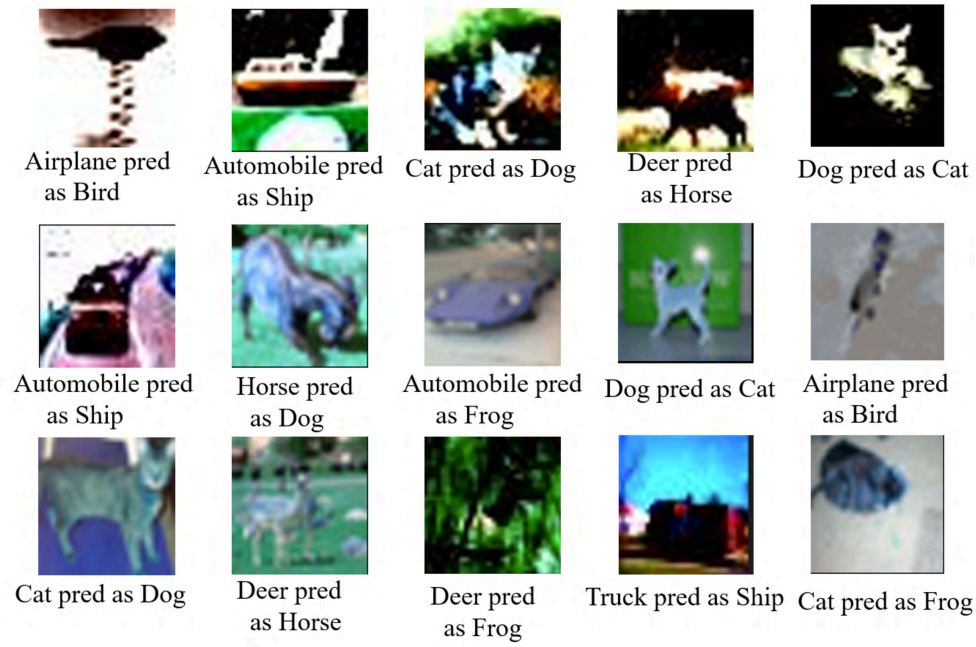
While our method brings notable benefits, examining the potential overhead introduced by the ISS agent for sample selection is crucial. This will help to analyse the computational costs and trade-offs associated with incorporating the ISS agent for sample selection, providing valuable insights into our approach's overall efficiency and effectiveness.

Let $k$ be the number of hidden layers in the CNN model, $n$ be the number of training samples, $d$ be the number of features, and $k_i$ be the number of nodes in each layer. Let $E$ be the number of epochs for training and $T$ be the number of iterations for Q-learning. For Classical-CNN training, the time complexity is dominated by the forward and backward passes of the gradient descent algorithm. The forward pass involves matrix multiplications between the input and weight matrices of each layer, which takes $O(dk_1 + \sum_{i=1}^{k-1} k_i k_{i+1})$ time per sample. The backward pass involves matrix multiplications between the error and weight matrices of each layer, which takes $O(k_1 + \sum_{i=1}^{k-1} k_i k_{i+1})$ time per sample. Therefore, the total time complexity for normal CNN training is $O(En(dk_1 + \sum_{i=1}^{k-1} k_i k_{i+1}))$. The space complexity for normal CNN training is dominated by the storage of the weight matrices of each layer, which takes $O(dk_1 + \sum_{i=1}^{k-1} k_i k_{i+1})$ space.

For RL-CNN, the time complexity is the sum of the meta-learning and fine-tuning phases. The meta-learning phase involves evaluating the CNN model on the original training dataset and updating the Q-table based on the rewards. The evaluation of the CNN model takes $O(n(dk_1 + \sum_{i=1}^{k-1} k_i k_{i+1}))$ time per epoch. The update of the Q-table takes $O(T)$ time per epoch. Therefore, the total time complexity for the meta-evaluation phase is $O(En(dk_1 + \sum_{i=1}^{k-1} k_i k_{i+1}) + ET)$. The fine-tuning phase uses gradient descent to fine-tune the CNN model on the informative training dataset. The size of the informative training dataset is at most $n$, so the time complexity for this phase is also $O(En(dk_1 + \sum_{i=1}^{k-1} k_i k_{i+1}))$. Therefore, the total time complexity for the proposed method is $O(En(dk_1 + \sum_{i=1}^{k-1} k_i k_{i+1})) + ET$.




**(a)** MNIST




**(b)** CIFAR-10

**Figure 3.4:** Examples of non-informative samples excluded from the training set by the RL method for MNIST and CIFAR-10 datasets.

The space complexity of the proposed method is the sum of the storage of the weight matrices of each layer, the Q-table, and the informative training dataset. The weight matrices take $O(dk_1 + \sum_{i=1}^{k-1} k_i k_{i+1})$ space. The Q-table takes $O(n)$ space. The informative training dataset takes at most $O(nd)$ space. Therefore, the total space complexity for the proposed method is $O(nd + dk_1 + \sum_{i=1}^{k-1} k_i k_{i+1})$.

The proposed method introduces a slight overhead in time complexity due to the ISS agent and the CNN model in the meta-learning phase. However, this overhead

is negligible compared to the significant improvement in performance and reliability achieved by the proposed method. Furthermore, the proposed method has a comparable space complexity to normal CNN training, as it only stores a subset of informative samples rather than all.

## 3.3 Discussion

One of the implications of incorporating reinforcement learning for ISS is that it can be seen as a form of curriculum learning, where the CNN model learns from easy to hard samples in a self-paced manner. Curriculum learning improves generalisation ability by presenting the samples in a meaningful order, from simple to complex or familiar to novel. However, curriculum learning also faces some challenges, such as defining the difficulty of each sample, avoiding discarding informative samples, and obtaining feedback or guidance from an expert or a teacher. In the proposed method, we tackled these challenges by using ISS agent to learn how informative each sample was based on how much it improved the CNN model's performance and how good each selection or rejection is. In this way, the ISS agent acts as a curriculum generator that selects the most informative or relevant samples for each iteration and excludes uninformative samples. Unlike the existing methods for curriculum learning that rely on predefined criteria or heuristics to order the samples, our method uses ISS to learn a policy that adapts to the learning dynamics and the data distribution.

The proposed method has some limitations that can be addressed in future research and can be extended in multiple directions. The proposed method does not involve any expert opinion or uncertainty measure for the sample selection. Moreover, the model lacks explainability for its selection decisions, which may reduce its trustworthiness and transparency. Therefore, a possible improvement could be to design a model that accepts expert opinion and uses uncertainty instead of point estimation for sample selection.

The reasons for choosing a simple CNN architecture for this work was primarily guided by two factors. First, the work presented here is conference and subject to the space constraints of conference publications. Second, future extensions of this research intend to employ more complex CNN architectures. Nevertheless, the selection of the dataset is also crucial. Selecting challenging datasets with numerous misleading examples, such as fine-grained classification datasets, are ideal. Using robust CNN architectures for simple datasets can result in high performance, thereby marginalising the impact of the ISS agent. Therefore, the future direction is to apply this method to either simple CNN architectures that have performance issues, or complex architectures designed for more challenging tasks and datasets.

Another limitation of the proposed method is that our method uses Q-learning to design the ISS agent, which may not be the most efficient or effective algorithm for the sample selection problem. Q-learning requires a large state-action space, which may be impractical or intractable for large or complex datasets. Some possible directions

for future research are to extend the proposed method to handle multiple tasks and datasets and to use more efficient ISS algorithms, such as policy gradient methods or deep Q-networks.

## 3.4 Conclusion

In this chapter, we proposed a novel ISS agent based meta-learning method for image classification task. The ISS agent learns a policy to select or reject samples from the pool of candidates based on the predictions of the CNN model and the reward function. The CNN model is fine-tuned on the selected samples and evaluated on the test set. We conducted experiments on three public datasets (i.e., MNIST, Fashion-MNIST, and CIFAR-10) and compared the proposed method with the CNN model trained without the ISS agent. The results show that the proposed method outperforms the CNN model trained without the ISS agent on all datasets, achieving higher accuracy, faster convergence, better robustness, and better generalisation.

This work has some implications and directions for future research. First, it shows that not all samples are informative and that some samples may cause the model to overfit or perform poorly. Therefore, excluding them might minimise bias and overfitting and to improve the quality and efficiency of the training data. Second, it opens up new possibilities for enhancing model performance and avoid bias.

# Chapter 4

## Uncertainty-Guided Learning with Monte Carlo Dropout

Deep Neural Networks (DNNs) have revolutionised many fields, but their real-world application can be limited by many challenges. One of the main challenges is the need for large amounts of labelled data to train the DNNs, which is often costly and time-consuming to obtain, especially for specific domains that require expert annotation. Furthermore, DNNs often struggle to quantify their own uncertainty, leading to overconfident and erroneous predictions, especially when dealing with noisy data, out-of-distribution samples, or the inherent limitations of the model itself. To address this challenge and the need for large amounts of labelled data, we propose using Monte Carlo (MC) dropout, a robust uncertainty estimation technique, that optimises the utility of manual annotation, especially for specific domains requiring expert annotation.

We estimate uncertainty using MC dropout and propose two human-in-the-loop approaches that use the uncertainty measures to select the most informative samples for annotation. Our first approach employs an active learning framework incorporating human annotators into the training loop, using MC dropout to assess the uncertainty of unlabelled data. The idea is to use the available unlabelled data to query the annotator for the labels of the most uncertain samples, which are expected to provide the most information gain for the DNN. This strategy aims to enhance both training accuracy and reduce annotation efforts. The second approach extends a human-in-the-loop testing for classifiers equipped with a rejection option, based on the uncertainty estimate. This enables the DNN to request feedback from the human annotator or abstain from making a decision during inference. Specifically, when the model encounters either out-of-distribution or noisy samples, it can choose to say "No, I do not know," thus avoiding potential missclassifications. The model consults the human annotator to verify the samples with large prediction uncertainty, likely to be misclassified. This strategy helps to improve accuracy while minimising the number of samples directed to the expert annotator.

We evaluate the applicability of the proposed method to a real-world problem within the Mosquito Alert project [126], where selecting informative samples can reduce labelling cost and error in identifying mosquito species. Mosquito Alert is a citizen science initiative that monitors and controls the spread of invasive mosquito species that transmit diseases such as dengue, Zika, or chikungunya. This project relies on citizens taking pictures of mosquitoes and uploading them to a central dataset where experts manually label the mosquito species. However, this process is time-consuming and costly, as the experts cannot handle all the labelling tasks.

To overcome these challenges, we design a DNN that leverages model uncertainty

to select informative and representative samples to enhance the performance and reliability of a DNN model. The model uses MC dropout to measure the uncertainty of its predictions and integrate it into an active learning framework. During inference, the model ranks the citizen-submitted samples based on their uncertainty score and labels the samples that it is confident. It then forwards the ambiguous or out-of-distribution samples to the experts for further inspection. This way, the experts only deal with a smaller subset of samples requiring their attention, while the DNN labels most samples. This approach speeds up the labelling process and ensures a more cost-effective utilisation of expert resources.

## 4.1 Proposed Methods

Over the past few years, MC dropout has been used to approximate Bayesian inference in DNN and estimate uncertainty in to the model prediction [49, 53, 127]. MC dropout is used as a practical way to estimate uncertainty in model predictions, acting as an efficient alternative to more computationally demanding Bayesian methods. Specifically, MC dropout enables the design of DNNs that utilise variational inference as an efficient approximation strategy for Bayesian inference during training and testing, where we get multiple predictions for each input by running multiple forward passes. Each forward pass produces a prediction that is a sample from the probability distribution over the possible classes given by the softmax output of the network. The final probability score for a given input image is obtained by averaging the scores from multiple forward passes, and the variance is used to measure model uncertainty. This model uncertainty reflects how confident the DNN is about its predictions.

Suppose that we have a dataset of $n$ samples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, with $x_i \in \mathbb{R}^{d_x}$ and $y_i \in \mathbb{R}^{d_y}$, our goal is to train a neural network $\mathcal{H}(x) = \mathbb{E}[\mathbf{Y}|\mathbf{X} = x]$ that classify images by minimising the cross-entropy between the class labels and the softmax output as in Eq. 4.1.

$$p(y_i|x; w, b) = \frac{\exp(x^T w_i + b_i)}{\sum_{j \in d_y} \exp(x^T w_j + b_j)} \tag{4.1}$$

where $w$ and $b$ are the weight and bias parameters. This softmax function takes the exponential of each score and divides it by the sum of all the exponentials to convert the raw scores from the network into probabilities that sum up to one.

We integrated the concept of MC dropout into the fine-tuned version of VGG-16 [128] to obtain a calibrated model with uncertainty estimation, as motivated by [63, 129]. After each convolution and fully-connected layer, we added a dropout layer with a probability of $\alpha$ and $\beta$, respectively, and kept these layers active during the evaluation phase to define a variational posterior distribution for each weight matrix, as shown in Eq. 4.2.

$$\begin{aligned} z_i &\sim Bernoulli(p_i) \\ W_i &= M_i \cdot diag(z_i), \end{aligned} \tag{4.2}$$

where $z_i$ denotes the random inactivation coefficients, $M_i$ denotes the weights matrix prior to dropout, and $p_i$ denotes the activation probability for the $i^{th}$ layer, which can be learned or manually set. This means that some of the neurons will have their outputs multiplied by zero, which effectively removes them from the network to prevent overfitting and improve generalisation. The dropout layer can be seen as a way of sampling from a distribution over the possible network configurations, where each configuration has a different set of active neurons. The activation probability $p_i$ is not a value of the activation function, but rather a hyperparameter that controls how likely a neuron is to be kept in the network. For example, if $p_i = 0.5$, then each neuron has a 50% chance of being dropped out.

More formally, Eq. 4.3 represents the loss function which incorporates two regularisation techniques, L2 regularisation and dropout applied to the cross-entropy loss function, to improve generalisation performance on our relatively small dataset. This loss function is analogous to a standard objective function that incorporates dropout for regularisation and an additional weight decay term. The dropout applied to the cross-entropy quantifies the model's accuracy in predicting the true class labels based on the input data, and it is minimised when the network assigns high probabilities to the correct classes. The weight decay regularisation, controlled by the hyperparameter $\lambda$, penalises large weights in the network, helping to mitigate overfitting and enhance the model's generalisation. This regularisation term is significant as it imposes an L2 norm penalty on the weight parameters, which is particularly important for smaller datasets. Dropout, applied to the cross-entropy loss function, improves generalisation performance on our relatively small dataset by forcing the network to rely on different subsets of neurons during each training epoch, which helps prevent overfitting.

$$\ell_{droupout} = -\sum_{i=1}^{n} \log \frac{\exp(x_i^{transp} w_i + b_i)}{\sum_{j \in d_y} \exp(x_i^{transp} w_j + b_j)} + \lambda \sum_{i=1}^{n} w_i^2. \tag{4.3}$$

During evaluation, we performed $T$ stochastic forward passes through the trained model to estimate the prediction uncertainty. Each stochastic forward pass ($t \in \{1, 2, \cdots, T\}$) produces a new softmax prediction ($\tilde{y}^t$). The posterior predictive distribution is obtained by averaging the softmax outputs from multiple stochastic forward passes through the network with dropout layers active. The class mean ($\mu$) of each distribution represents the final prediction for a given input, and it is calculated by averaging a distribution per each class. The variance ($\sigma$) of this distribution represents the model uncertainty, and it is calculated by measuring how much the samples deviate from the mean.

$$\mu = \frac{1}{T} \sum_{i=1}^{T} \tilde{y}^i, \quad \sigma = \frac{1}{T-1} \sqrt{\sum_{i=1}^{T} (\tilde{y}^i - \mu)^2}. \tag{4.4}$$

### 4.1.1 MC dropout as an acquisition function in active learning

Active learning (AL) is a paradigm that aims to reduce the labelling cost and improve the model performance by selecting the most informative samples from a pool of unlabelled data for expert annotation. The selection criterion is usually based on an acquisition function, which measures the expected value of observing the label of a sample. A common acquisition function for AL is least confidence, which selects samples with low prediction confidence, indicating that the model is uncertain about its output. However, the least confidence only uses the direct output of the softmax layer, which may not reflect the true probability distribution over the possible classes. In fact, DNN models have demonstrated that deeper architectures typically lead to higher confidence scores, which tend to overestimate their accuracy. This may lead to selecting samples that are not very informative or representative of the data distribution, and thus slowing down the training. Moreover, least confidence can be sensitive to noisy or outlier data points, as these often present low confidence scores, diverting valuable annotation resources towards less meaningful samples.

In our active learning framework, the acquisition function, $Q(x_i)$ is formulated using the variance of the model's prediction, obtained through Monte Carlo (MC) dropout:

$$Q(x_i) = \sigma^2(x_i) \tag{4.5}$$

where $\sigma$ represents the variance of the predictions for sample $x_i$ over multiple stochastic forward passes with dropout enabled. The model selects samples $\{x_{i*}\}$ for which $Q(x_{i*})$ exhibit the lowest prediction variance, indicating where the model's certainty is high. This process ensures that the selection of samples is data-driven and focuses on refining the model where it is most needed, optimising the allocation of annotation resources.

To select samples from a pool of $m$ unlabelled samples, $\mathcal{U} = \{(x_i, l_i)\}_{i=1}^{m}$, where $l_i$ represents an unknown label, we calculate the variance of the posterior predictive distribution for each sample, which measures the model (epistemic) uncertainty. We rank the samples based on their acquisition scores and select the top $k$ samples with the highest scores to form a batch $\mathcal{B} = \{(x_i, l_i)\}_{i=1}^{k}$, where $k < m$. Then, we request the expert to annotate the samples in $\mathcal{B}$ and add them to a labelled dataset $\mathcal{L} = \{(x_i, y_i)\}_{i=1}^{n}$, where $y_i$ is the true label. Finally, we update $\mathcal{U}$ by removing the samples in $\mathcal{B}$ from it. This iterative process is continued until we achieve a targeted model accuracy or exhaust a labelling budget.

### 4.1.2 Classification with rejection

One of the main challenges in DNN is handling inputs that are out of the model's training distribution or contain ambiguities or noise. Such inputs can cause DNNs to produce uncertain predictions, which means the model is not confident about its decision. However, DNNs often do not express their uncertainty explicitly and produce

predictions for any given input, even if they are incorrect or meaningless. For example, a DNN model trained to classify cats and dogs will say either cat or dog when it gets an input of a human, even if with low confidence. This behaviour can lead to erroneous decisions, which can have serious consequences in safety-critical applications, such as medical diagnosis, autonomous driving, or fraud detection. Therefore, it is important to develop methods that can estimate and communicate the uncertainty of DNNs and enable them to reject inputs for which they cannot make confident predictions.

A possible way to achieve this goal is to use classification with rejection, a testing approach that allows the model to say "No, I do not know" when faced with ambiguous or out of distribution inputs. This approach is related to active learning, which is a training approach that allows the model to select the most informative samples from a pool of samples and request their labels from an expert but differs from it in two aspects: first, classification with rejection works under the open set assumption, where the model may encounter inputs that belong to unknown classes that are not seen during training; second, classification with rejection works during the inference phase, where the model queries the expert for labels only when it is uncertain about its predictions, and it does not update the model parameters. Therefore, classification with rejection is a useful approach that allows the model to express its uncertainty and query the expert for labels during inference. However, a crucial research question remains: how can uncertainty be measured and effectively utilised, particularly with respect to optimising resource allocation between trained models and human experts?

To address this, we implement MC dropout during inference and equip the classifier with rejection capabilities. This classifier is designed with rejection criteria based on uncertainty scores or percentiles, which dictate whether a sample is to be accepted or flagged for expert review. Rejected samples, characterised by high uncertainty, are automatically marked for annotation. Such a system aims to decrease the volume of inaccurate or irrelevant predictions that necessitate expert review, thereby minimising the experts' time and ensuring their expertise is reserved for the most ambiguous cases.

We consider a scenario where we have a trained model $\mathcal{H}$ and a pool of samples $\mathcal{U}$ to make predictions. We also have a budget $\mathcal{B}$ that determines the maximum number of samples that can be queried to the expert for labelling. Using MC dropout, the model estimates the uncertainty for each sample in $\mathcal{U}$ and ranks them based on their uncertainty score. Depending on the rejection rules used, the model accepts or rejects the samples and accepted samples are labelled by the model, while the rejected ones are forwarded to the expert.

We define the first rejection rule as the uncertainty threshold, where the model employs a threshold value $\tau$ to accept or reject samples. The model makes predictions only on the samples that have an uncertainty score $\leq \tau$, and forwards the rest of the samples to the expert for annotation. Thus, out of $n$ total samples in $\mathcal{U}$, $N = n - R$ samples are accepted by the model and $R$ samples are rejected, as long as $R \leq \mathcal{B}$, where, $N$ and $R$ represents the non-rejected and rejected samples, respectively. This

is formulated as follows:

$$\tilde{y} = \begin{cases} \mathcal{H}(x) & \sigma \leq \tau \\ \text{query} & \sigma > \tau \text{ and } R \leq \mathcal{B} \\ \text{reject} & \sigma > \tau \text{ and } R > \mathcal{B} \end{cases} \tag{4.6}$$

where $\tilde{y}$ is the predicted label for a sample $x$, $\sigma$ is the uncertainty score, $\mathcal{B}$ is the budget and $R$ is rejected samples.

The second rejection rule is based on the percentile (number of samples) that the expert can annotate. The expert can specify the number of most uncertain samples to be annotated by the expert ($R$), and the model will automatically make predictions on the rest of the samples ($A$). Given $n$ number of samples in $\mathcal{U}$, the model ranks the samples based on their uncertainty and predicts only the top $A$ most certain samples. It then forwards the bottom $R$ most uncertain samples to the expert for annotation. We formulate this scenario as follows:

$$\tilde{y} = \begin{cases} \mathcal{H}(x) & \rho \leq A \\ \text{query} & \rho > A \text{ and } R \leq \mathcal{B} \\ \text{reject} & \rho > A \text{ and } R > \mathcal{B} \end{cases} \tag{4.7}$$

where $\rho$ is the rank of the sample based on its uncertainty score. In our setting, we assume that $R$ is always less than or equal to $\mathcal{B}$, which means that all samples will be either labelled by the model or the expert.

### 4.1.3 Visual explainability

Visual expandability involves generating visualisations or explanations that highlight how a model processes input data and arrives at its conclusions, making it easier to interpret, trust, and improve the model's decisions.

In this chapter, we employ two saliency-based visual explainability methods, namely Grad-CAM [90] and Bayesian Layer-Wise Relevance Propagation (B-LRP) [130], to assist users and experts understand how the model makes decision and learns new patterns to identify mosquito species. Grad-CAM uses the gradients of the final convolutional layer for the predicted class, to produce a coarse localisation map highlighting the important regions in the image for predicting the class. B-LRP assigns relevance scores to each input pixel based on how much they contribute to the output prediction, taking into account the uncertainty of the model. B-LRP applies a local normalisation step to each layer to improve the stability and interpretability of the relevance scores. We use Grad-CAM to generate saliency maps and explain how the model makes decisions and identify the parts of the mosquito species. We use B-LRP to explain what causes the uncertainty, and the sources of the uncertainty during prediction.

## 4.2 Experiments

### 4.2.1 Mosquito alert dataset

We conducted experiments on the open-source dataset from the Mosquito Alert project[1] [126]. This project was launched in 2014 by the Centre for Research and Ecological Applications (CREAF) and the Centre for Advanced Studies of Blanes (CEAB-CSIC) near Barcelona (Spain) to monitor and control disease-carrying mosquitoes. The data collection process for this platform is based on images of mosquitoes and mosquito breeding sites submitted by citizens. The uploaded images are in RGB format. Since this platform has no photo size restrictions, image sizes range from $200 \times 200 \times 3$ to $460 \times 460 \times 3$, with an average size of $420 \times 368 \times 3$. A team of three entomologists inspects, validates, and classifies the images that are submitted. In the event of a disagreement, the final label is assigned by a senior-expert who holds the final decision.

During the period of our study, the Mosquito Alert dataset comprised images categorised as*Aedes albopictus*, *Aedes aegypti*, other species, unclassified or unknown, with each image being further classified as confirmed or probable by expert analysis. At the time of our analysis, the dataset included a total of 3364 confirmed *Aedes albopictus* images. Figure 4.1 shows samples from the Mosquito Alert dataset, which are *Aedes albopictus* and other *Non-tiger* samples.

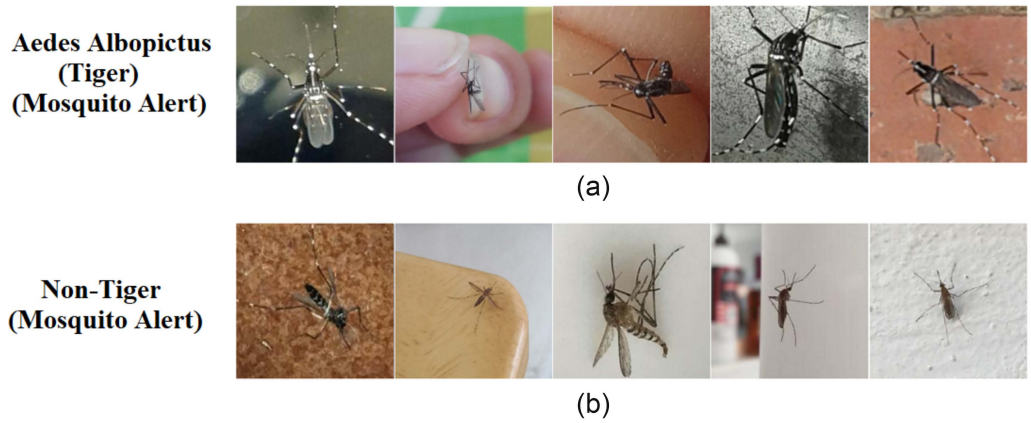



**Figure 4.1:** Sample of (a) *Aedes albopictus* (tiger) and (b) *non-tiger* mosquitoes from the Mosquito Alert dataset.

It is important to note that the Mosquito Alert dataset is continually updated and expanded, potentially leading to an increase in the number and variety of mosquito species beyond those available during our experiment. For the purposes of our study, we used 3364 images labelled as confirmed *Aedes albopictus* cases to represent positive samples and classified images of other species as negative samples. This approach allowed us to train our architecture on a dataset consisting of 6378 images, representing

[1]The database is available on this link.

both tiger and non-tiger mosquito categories. We acknowledge that subsequent access to the dataset may reveal larger quantities and additional varieties of mosquitoes, reflecting ongoing contributions to and evolution of the Mosquito Alert project.

### 4.2.2 Architecture details

To leverage the power of DNNs for uncertainty estimation, we chose the VGG-16 [131] pre-trained model as our base architecture. However, the standard VGG-16 architecture lacks dropout layers, essential for quantifying uncertainty through MC dropout. Therefore, we modified the model's capabilities to quantify uncertainty by incorporating dropout layers after each convolutional and fully connected layer.

We modified the pre-trained VGG-16 architecture by applying dropout with probability $\alpha$ after each convolution layer and setting the dropout rate of the fully connected layers to $\beta$. However, since no definitive values for $\alpha$ and $\beta$ universally apply across datasets, we conducted an extensive experiment to determine the optimal combination for our specific dataset. We trained the modified architecture with different values of $(\alpha, \beta)$ using five-fold cross-validation and evaluate the mean accuracy to find the best combination. As shown in Figure 4.2, we achieved the highest accuracy of 97.6% at $\alpha = 0.5$ and $\beta = 0.4$, and we used these dropout rates for the rest of the experiments. The proposed end-to-end architecture was implemented using PyTorch [132] and trained using Tesla K80 GPU. We compared our model with the previous studies on mosquito alert [128, 133] and demonstrated that the proposed model achieved better performance.

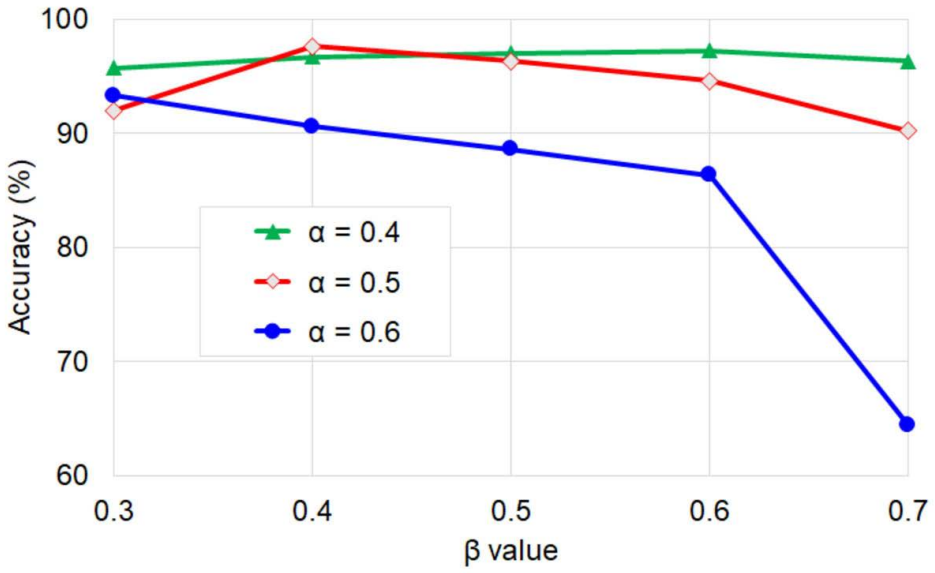



**Figure 4.2:** Accuracy of the proposed architecture as a function of $\alpha$ and $\beta$, where $\alpha$ and $\beta$ are the dropout rates for the hidden and fully-connected layers, respectively.

We used cross-entropy as a loss function and stochastic gradient descent with initial learning rates and momentum of 0.001 and 0.9, respectively, during the training process. The learning rate decayed by a factor of 0.1 every 5 epochs by using the StepLR scheduler. We fed the network with a mini-batch size of 64, and the optimisation process was terminated after 25 epochs. We did not use data augmentation to avoid unrealistic changes in micro-morphological patterns of mosquitoes that could have skewed the final results. We kept the dropout active during the test phase to measure the uncertainty outcomes from the MC dropout by conducting $T = 100$ stochastic forward passes through the network. As a result, rather than a single point estimate for a given input, we have a per-class output distribution of the softmax confidence in which the distribution variance serves as the model uncertainty.

### 4.2.3 Evaluation metrics

To evaluate the performance of the proposed classifier with rejection, we need to consider not only the accuracy of the accepted samples but also the quality of the rejected samples. A naive way to achieve high accuracy is to reject more uncertain or ambiguous samples, but this would reduce the coverage and usefulness of the classifier. On the other hand, accepting more samples that are likely to be misclassified would lower the accuracy and reliability of the classifier. Therefore, we need to balance accuracy and coverage and measure how well the classifier can reject truly difficult or out-of-distribution samples.

To this end, we adopted the performance metrics proposed by Filipe et al. [134], which include non-rejection accuracy ($NRA$), classification quality ($CQ$), and rejection quality ($RQ$). The non-rejection accuracy is the accuracy of the model on the accepted samples, i.e., the ratio of correctly classified samples to the total number of accepted samples. The classification quality is the ratio of correctly classified samples to the total number of samples, i.e., the model's overall accuracy. The rejection quality is the ratio of incorrectly classified samples to the total number of rejected samples, i.e., how well the model rejects uncertain or ambiguous samples. The metrics are formulated as follows:

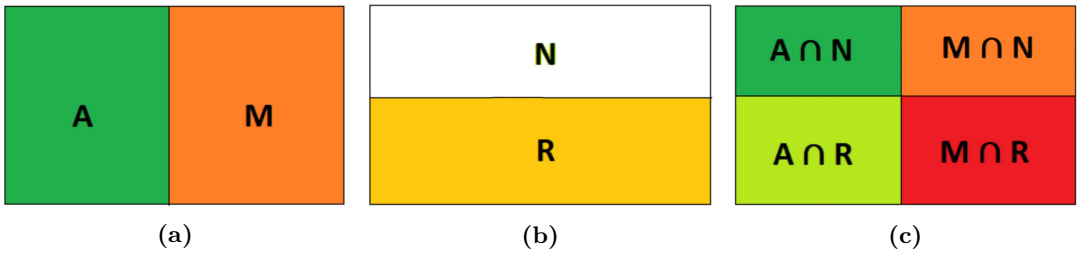



**Figure 4.3:** Rejection performance metrics proposed in [134]. (a) Classification partition space (b) rejection partition space, (c) classification with rejection. Correctly classified samples, misclassified samples, non-rejected samples, and rejected samples are represented by the letters $A$, $M$, $N$, and $R$, respectively.

$$
\begin{aligned}
NRA &= \frac{|A \cap N|}{|N|} \\
CQ &= \frac{|A \cap N| + |M \cap R|}{|N| + |R|} \\
RQ &= \frac{|M \cap R|\,|A|}{|A \cap R|\,|M|}.
\end{aligned}
\tag{4.8}
$$

where $A$, $M$, $N$, and $R$ represent correctly classified samples, misclassified samples, non-rejected samples, and rejected samples, respectively. $NRA$ measures the classifier's ability to accurately classify non-rejected samples, $CQ$ measures the classifier's ability to accurately classify non-rejected samples and reject misclassified samples, and $RQ$ measures the classifier's ability to concentrate all misclassified samples into the rejected partition of samples. We also reported precision, recall, and F1-score using Equaiton 4.9 to provide further insight into the predictive model performance and to compare it to competing methods.

$$
\begin{aligned}
\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \\
\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.
\end{aligned}
\tag{4.9}
$$

where $TP$, $FP$ and $FN$ stand for true positive, false positive and false negative, respectively.

### 4.2.4 Results

In this section, we present the experimental results of the proposed method in various scenarios. We first use the proposed method as an acquisition function to design an AL framework and accelerate the learning process of a DNN model. Next, we tested the trained model in a classifier with rejection settings. The DNN model can reject uncertain prediction and refer them to the expert for annotation. We show that using model uncertainty as a proxy to select samples improves the model's performance and robustness. Finally, we apply existing explainability methods and approaches to interpret the model predictions and the sources of uncertainty.

#### Active learning with uncertainty based sampling

We apply the proposed uncertainty estimation in AL to select informative samples and accelerate the model training. We compare the proposed method with two baseline methods: least-confidence and random sampling. The least-confidence acquisition function selects the most informative samples based on the model's confidence in its prediction. This method assumes that the model is more uncertain about samples with lower confidence scores. The MC dropout acquisition function selects the most

informative samples based on the variance of the posterior predictive distribution for each sample, which measures the epistemic uncertainty of the model. This method assumes that the model is more uncertain about the samples with higher variance scores. It can also consider the informativeness and diversity of the samples by combining the variance and the mean of the posterior predictive distribution. We measure the test accuracy, the number of queries, and the learning curves of the methods. The results show that MC dropout as an acquisition function performs better with fewer samples.

Figure 4.4 show the results of the experiment and the proposed method reaches higher test accuracy with fewer labelled samples compared to the other methods. Although uncertainty based sampling converges slower at the beginning of the training, it improves gradually by starting with a small number of labelled samples and adding more informative samples. For example, uncertainty based sampling method achieved a test accuracy of 90% using only 25% of the labelled training samples, while the least-confidence and random methods need 34% and 58% of the data, respectively.

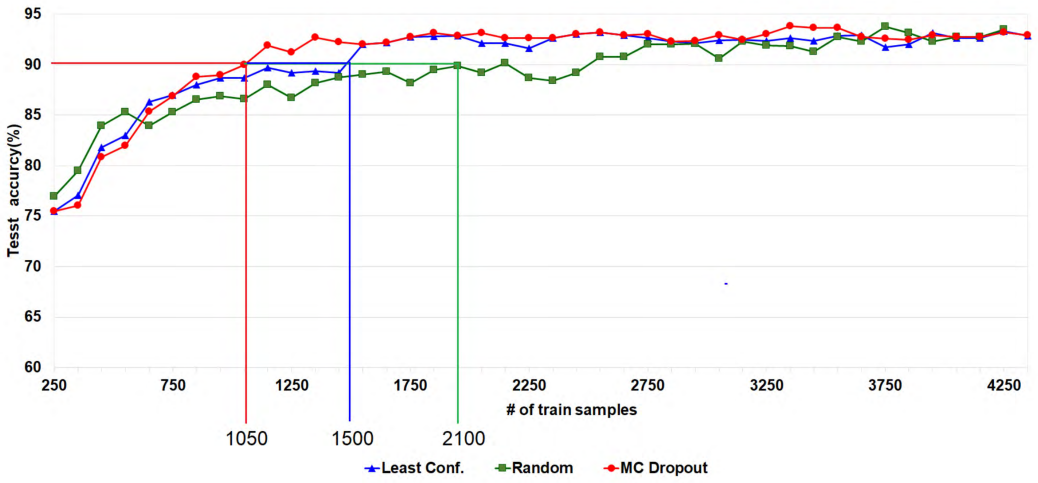



**Figure 4.4:** A comparison of three query strategies based on accuracy and the number of images acquired from the pool.

The results demonstrate that MC dropout as an acquisition function improves the model's performance with fewer samples. In the following section, we extend this approach during the inference stage, where the experts are queried to annotate the most uncertain samples during prediction.

### Classification with rejection

Classification with rejection is a technique that allows the model to say "No, I do not know" for some samples that are too ambiguous or difficult to classify and forward them to the expert. We apply the proposed uncertainty sampling method to design a classifier with rejection during the evaluation phase. This classifier uses model

uncertainty estimated using MC dropout. It improves the model's performance and robustness as it dynamically queries the expert for samples that the model is uncertain about. We explore various conditions with different uncertainty thresholds and DNN architectures to evaluate the classification with rejection.

To evaluate the classifier with rejection, we consider two rejection policies. The first policy involves setting an uncertainty threshold, $\tau$, selected from a range of values $\{0.08, 0.1, 0.2, 0.3\}$. These values were determined based on the observed minimum and maximum estimated uncertainty (variance) within the Mosquito Alert dataset, which range between 0.08 and 0.3. A lower threshold (e.g., 0.08 or 0.1) allows the model to accept only samples with minimam model uncertainty and high confidence, referring the rest to human experts for annotation. While this approach boosts evaluation accuracy, it also increases annotation costs due to the higher number of samples requiring expert review. On the other hand, a higher threshold (e.g., 0.3) permits the model to accept a wider range of samples, even those with higher model uncertainty. Although this reduces annotation costs, it compromises performance and increase the likelihood of accepting more uncertain predictions.

**Table 4.1:** Performance of different DNN architectures referring uncertain samples with varying uncertainty thresholds $\tau$. The uncertainty threshold determines when a DNN model says "No, I don't know" and refers a sample to an expert.

| Model | Un. Threshold | Rejected samples # | Retained samples (Samples #) | TP | TN | FP | FN | Precision | Recall | F1 score |
|---|---|---|---|---|---|---|---|---|---|---|
| **Proposed architecture** | $\tau \leq 0.08$ | 467 | 808 | 502 | 290 | 3 | 13 | 0.99 | 0.98 | 0.98 |
| | $\tau \leq 0.1$ | 390 | 885 | 535 | 329 | 6 | 15 | 0.99 | 0.97 | 0.98 |
| | $\tau \leq 0.2$ | 125 | 1140 | 636 | 447 | 15 | 52 | 0.97 | 0.93 | 0.95 |
| | $\tau \leq 0.3$ | 3 | 1272 | 665 | 495 | 30 | 82 | 0.96 | 0.89 | 0.92 |
| | No $\tau$ | 0 | 1275 | 664 | 492 | 31 | 88 | 0.95 | 0.88 | 0.92 |
| **Adhane et al.** [128] | $\tau \leq 0.08$ | 195 | 1080 | 572 | 470 | 20 | 18 | 0.97 | 0.97 | 0.97 |
| | $\tau \leq 0.1$ | 155 | 1120 | 596 | 477 | 24 | 23 | 0.96 | 0.96 | 0.96 |
| | $\tau \leq 0.2$ | 2 | 1273 | 651 | 535 | 42 | 45 | 0.94 | 0.94 | 0.94 |
| | $\tau \leq 0.3$ | 0 | 1275 | 653 | 543 | 42 | 46 | 0.94 | 0.93 | 0.94 |
| | No $\tau$ | 0 | 1275 | 652 | 533 | 43 | 47 | 0.94 | 0.93 | 0.94 |
| **AlexNet** [135] | $\tau \leq 0.08$ | 254 | 1021 | 593 | 377 | 15 | 36 | 0.97 | 0.94 | 0.95 |
| | $\tau \leq 0.1$ | 220 | 1055 | 606 | 390 | 16 | 43 | 0.97 | 0.93 | 0.95 |
| | $\tau \leq 0.2$ | 37 | 1238 | 652 | 473 | 34 | 79 | 0.95 | 0.89 | 0.92 |
| | $\tau \leq 0.3$ | 0 | 1275 | 658 | 487 | 37 | 93 | 0.95 | 0.87 | 0.91 |
| | No $\tau$ | 0 | 1275 | 658 | 497 | 37 | 88 | 0.95 | 0.88 | 0.91 |
| **Modified AlexNet** | $\tau \leq 0.08$ | 705 | 570 | 303 | 251 | 6 | 10 | 0.98 | 0.96 | 0.97 |
| | $\tau \leq 0.1$ | 615 | 660 | 365 | 272 | 9 | 14 | 0.97 | 0.96 | 0.97 |
| | $\tau \leq 0.2$ | 202 | 1073 | 574 | 402 | 23 | 74 | 0.96 | 0.88 | 0.92 |
| | $\tau \leq 0.3$ | 17 | 1258 | 636 | 467 | 48 | 107 | 0.92 | 0.85 | 0.89 |
| | No $\tau$ | 0 | 1275 | 641 | 471 | 54 | 109 | 0.92 | 0.85 | 0.88 |

Table 4.1 summarises the performance of classification with rejection for various DNN models with various uncertainty threshold values. The uncertainty threshold determines when a DNN model has to say "No, I do not know" and reject the sample to an expert. A lower uncertainty threshold means more rejections and a higher threshold means more samples accepted. We compared the proposed architecture with [128, 135] and a modified AlexNet. We modified AlexNet [135] in the same way we modified VGG-16 [128], i.e., adding a dropout layer after each convolution and FC layers to the original AlexNet architecture.

In the second policy, we used percentiles to select samples that will be labelled by

the model and refer the rest to the expert. We compare two approaches of selecting the samples to refer to an expert: informed referral and random referral. Informed referral ranks the samples based on their estimated uncertainty and refers the most uncertain ones to the expert. Random referral selects the samples randomly and refers them to the expert. We measure the performance of the model on the quality of the samples retained and referred. The quality is measured by comparing the trade-off between the accuracy and the sample coverage using the metrics proposed to evaluate classification with rejection [134]. The metrics capture how well the model can classify the samples it accepts and how well it refers the uncertain samples to the expert.

Figure 4.5 shows the percentage of correct and incorrect predictions that are accepted or referred by the model. The model becomes more conservative and defers more predictions to the expert as the referral rate increases, which indicates the model prefers to rely on the expert's feedback for more samples, even for predictions that are correct. Informed referral accepted more correct and forwards more incorrect predictions compared to random referral. This is because informed referral leverages the model's uncertainty to assess its confidence and refer predictions that are more likely incorrect.

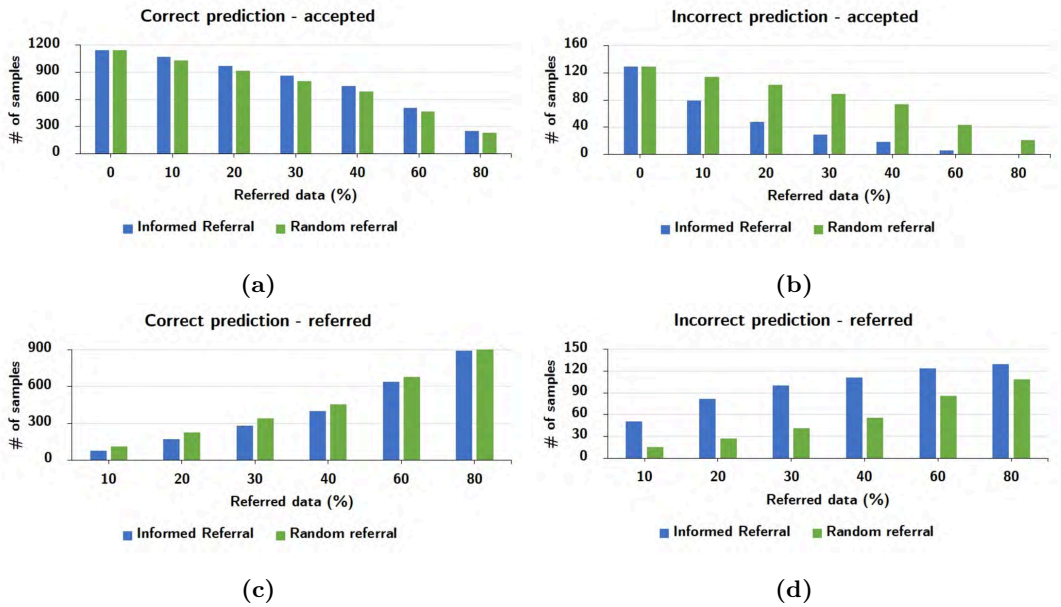



**Figure 4.5:** Number of samples that the model accepts or says "No I don't know" under different percentiles of referral for informed and random referral.

To systematically evaluate the implications of rejecting varying number of samples for expert review, we employ three key metrics: $NRA$, $CQ$, and $RQ$ formulated in Eq. 4.8. These metrics serve to quantify both the quality and robustness of the model's decisions pertaining to accepted and referred samples. We vary the referral percentile incrementally, ranging from 0% to 90%, and observe its impact on model performance. Our analyses reveal that an optimal balance between predictive accuracy and model

reliability is struck when 10-20% of the samples are referred for expert assessment, achieving in a performance enhancement of up to 4% (see Figure 4.6(b)).

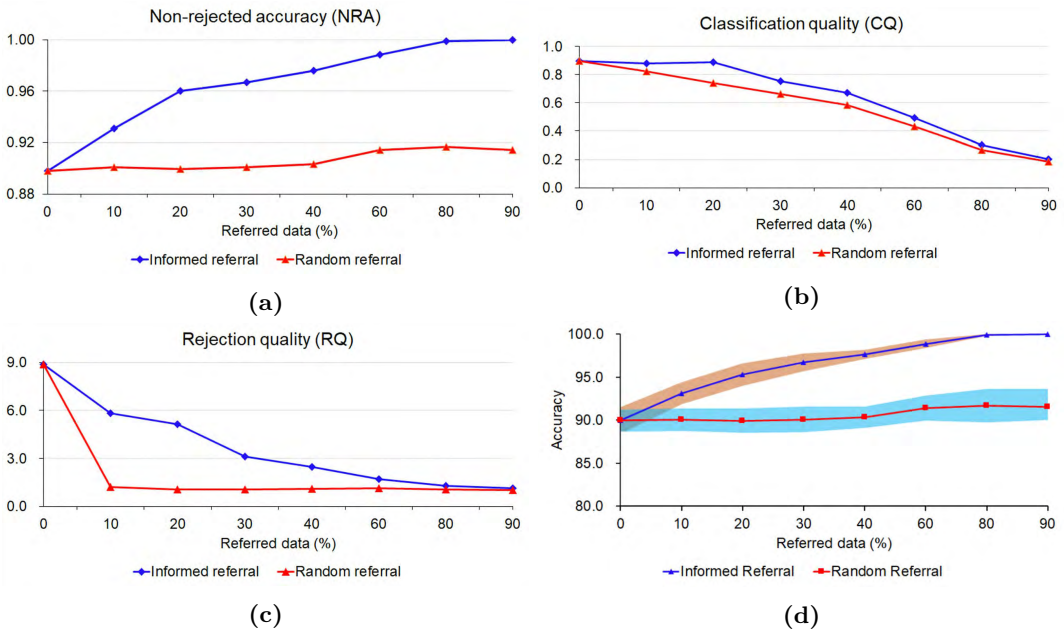



**Figure 4.6:** Performance measure of informed vs random referral as a function of the rejected fraction. (a) Non-rejection accuracy ($NRA$), where the Y-axis shows the accuracy of the non-rejected prediction. (b) classification quality ($CQ$), where the Y-axis shows the number of points that are correctly/wrongly classified. (c) rejection quality ($RQ$), where the Y-axis shows the number of points that are correctly/incorrectly classified and not-rejected/rejected over the total samples. (d) the prediction accuracy over $N\%$ of retained data as a function of informed vs random referral. The standard deviation from the five-fold cross-validation is shown by the shaded region around the curves.

Figure 4.6 presents a comparative analysis of informed and random referral policies across different values of $N$ (non-rejected samples)—the proportion of samples not referred to experts. The results indicate that the informed referral strategy demonstrates superior performance when $N$ is relatively small. Selecting samples based on their uncertainty score effectively identifies those most prone to misclassification or that lie outside the model's training distribution, thereby enhancing the model's overall accuracy.

Figure 4.6(a) explores the relationship between $N$ and $NRA$. In scenarios where 20% of the most uncertain samples are referred to an expert using the informed referral policy, the model is then evaluated on the remaining 80% of samples, characterised by lower uncertainty ($\sigma$). This results a higher $NRA$, approaching 100% accuracy. In contrast, a random referral strategy, referring 20% of samples, leaves behind a less predictable mix, leading to a decreased $NRA$ owing to increased misclassifications and rejections.

Figures 4.6(b) and (c) show how $CQ$ and $RQ$ vary as a function of $N$. The

*CQ* measures the model's capacity for correctly rejecting misclassified or out-of-distribution samples, while *RQ* quantifies the trade-off between model accuracy and the extent of sample coverage. For instance, referring 80% of all samples to experts results in minimum acceptance rates and larger rejection rates, negatively impacting both *CQ* and *RQ*. Thus, these metrics serve as valuable tools for identifying the optimal value of rejection point that maximises performance and efficiency. Accordingly, for our experiment, referring 20% of the samples to the expert emerges as the optimal strategy.

Since the optimal referral point is established at 20%, we further examine its impact on individual classes. To this end, we compute class-specific metrics such as recall, precision, and F1-score when $N$ is set at 20%, meaning the model opts to refrain from making predictions for the most uncertain 20% of samples. Table 4.2 reveals that the informed referral policy surpasses the random referral policy in terms of recall, precision, and F1-score across most classes.

**Table 4.2:** Comparison of informed and random referral for uncertain samples using class-wise recall, precision, and F1-score.

| | **Number of samples** | **TP** | **TN** | **FP** | **FN** | **Precision** | **Recall** | **F1-score** | **Accuracy** |
|---|---|---|---|---|---|---|---|---|---|
| No referral | 1275 | 614 | 553 | 58 | 70 | 0.91 | 0.89 | 0.90 | 0.91 |
| Random referral (20%) | 1021 | 486 | 432 | 49 | 54 | 0.90 | 0.90 | 0.90 | 0.90 |
| Informed referral (20%) | 1021 | 506 | 487 | 20 | 28 | 0.96 | 0.94 | 0.95 | 0.97 |

In addition, we examined the predictive posterior distribution under various uncertainty thresholds and displayed the histogram of the predictive posterior distribution for some randomly selected inputs. The histogram plotted from the multiple forward passes conducted during the evaluation phase, shows the probability distribution of the output labels for a given input sample and the prediction output. It also reflects the model's confidence and uncertainty in its prediction. A high probability for a certain class implies a confident prediction, while a low or uniform probability for all classes implies an uncertain prediction. Based on the predictive posterior distribution and the true label, we classified the outcome into four possible states.

1. *Correct certain*: This refers to the state of the outcome when the classifier predicts the correct class with a high level of certainty. In other words, the classifier confidently assigns the correct label to the input, and there is minimal or no uncertainty associated with this prediction.

2. *Correct uncertain*: This term describes the state of the outcome when the classifier predicts the correct class, but with a certain degree of uncertainty. In this case, the classifier assigns the correct label to the input, but it also indicates some level of doubt or ambiguity in its prediction. This uncertainty could arise due to various factors, such as the input being on the boundary between two classes or having similar characteristics to multiple classes.

3. *Incorrect uncertain*: It refers to the state of the outcome when the classifier predicts an incorrect class, but with a certain degree of uncertainty. In this scenario, the classifier assigns an incorrect label to the input, but it also expresses

some level of uncertainty or lack of confidence in its prediction. The uncertainty may arise due to the input having characteristics that make it difficult to classify accurately or being similar to multiple classes.

4. *Incorrect certain*: It is the state of the outcome when the classifier predicts an incorrect class with a high level of certainty. In this case, the classifier confidently assigns an incorrect label to the input, and there is minimam or no uncertainty associated with this prediction. The classifier's high confidence in an incorrect prediction suggests that it is likely making a systematic error or misinterpreting certain patterns in the data.

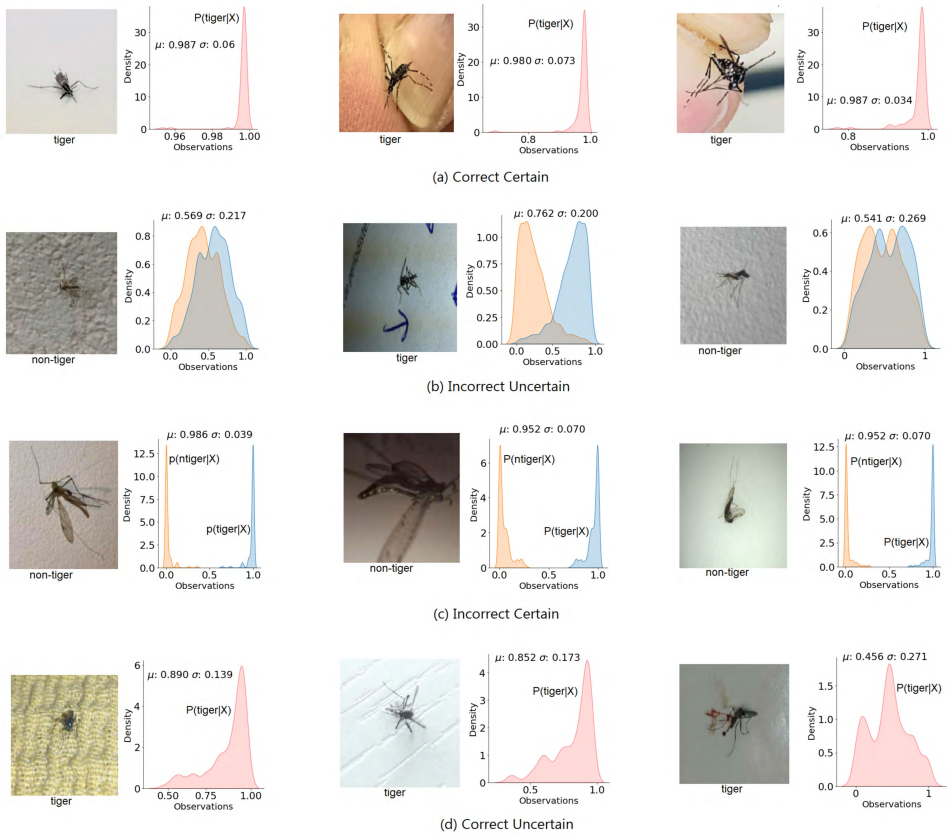



**Figure 4.7:** Visualisation of the posterior distributions and associated uncertainty estimates for 12 samples at $\sigma \leq 0.1$. The distribution of correct predictions is shown in red, while the distribution of incorrectly classified samples is shown in blue.

Based on the possible states described above, plotted the probability density function of the prediction's level of confidence. A single histogram with a narrow tail (i.e., *correct certain*) implies a confident and correct prediction, which the classifier accepts. A single plot with a broad tail (i.e., *correct uncertain*) or two overlapped plots (i.e., *incorrect uncertain*) imply an unconfident prediction, which is forwarded to the expert. Two non-overlapped plots with narrow tails (i.e., *incorrect certain*) imply a confident but incorrect prediction, which the classifier accepts. The model encounters some difficulties when processing images acquired in-the-wild, such as (1) various objects in

the image that are more salient than the mosquito and (2) mosquitoes with damaged body parts (see Figure 4.7 (b) and (d)). These factors can increase the uncertainty in the outcomes.

In summary, we used MC dropout to measure model uncertainty, identify the most likely misclassified or uncertain samples, and forward them to an expert for feedback. Model uncertainty was used as an acquisition function in active learning, and trained the model with more informative data, improving the model's accuracy and generalisation capability with fewer samples. Furthermore, model uncertainty helped to design a classifier with rejection, allowing the classifier to express its predictive doubts during inference and forward samples to the expert for annotation. Incorporating uncertainty sampling into an active learning framework significantly improved the model's efficiency, accuracy, and adaptability, making it a feasible approach in various domains.

### Visual explainability

To understand the model's decision-making process and the differences among mosquito species, we apply Grad-CAM [90] to visualise the regions of interest for the model. We observe that the model's decisions are consistent with the entomological knowledge but also that the model has learned some additional features that the entomologists have not used as classification criteria. This finding helps entomologists discover new characteristics to identify mosquito species. Moreover, we use B-LRP [130] to examine how the model estimates its uncertainty and what factors influence its confidence level. This method allows us to identify the input features that contribute to the uncertainty by breaking down the predictive distribution into individual feature contributions. We then compare the contributions of different features and see how they affect the predicted probabilities of different classes.

The use of visual explanations has shown that the key components of tiger mosquito specimens that support network convergence are the white band on the legs, abdomen patches, head, and thorax, see Figure 4.8 for the anatomy of *Aedes albopictus*. Entomologists mainly distinguish *Aedes albopictus* from other Aedes species by its morphological features such as the unique white banding patterns on the legs, clear markings on the dorsal surface of the abdomen, and noticeable stripes on the head and thorax [136, 137]. These morphological features are consistent with the regions of interest identified by Grad-CAM. Details of the findings are presented in the following section.

Figure 4.9 shows the Grad-CAM visualisation of randomly selected images predicted as a tiger and non-tiger mosquitoes. The heatmap reveals that for the tiger mosquito species, the model focuses mainly on the thorax and head of the mosquito, which are unique features of tiger mosquito species. On the other hand, the images predicted as non-tiger mosquito shows that the model pays attention to the legs and abdomen, which differ from those of the tiger mosquito, and the areas around the wings, which may vary among species.

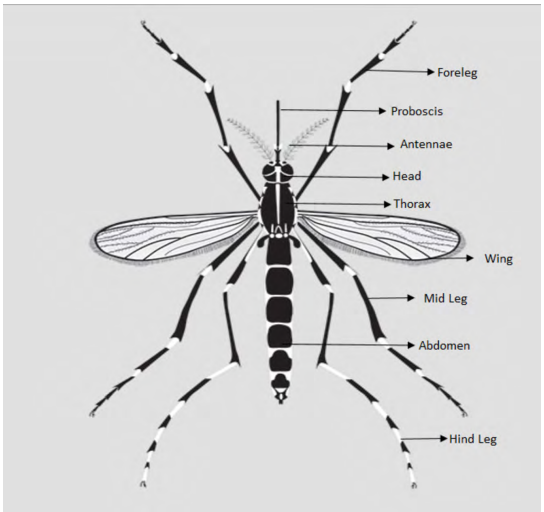
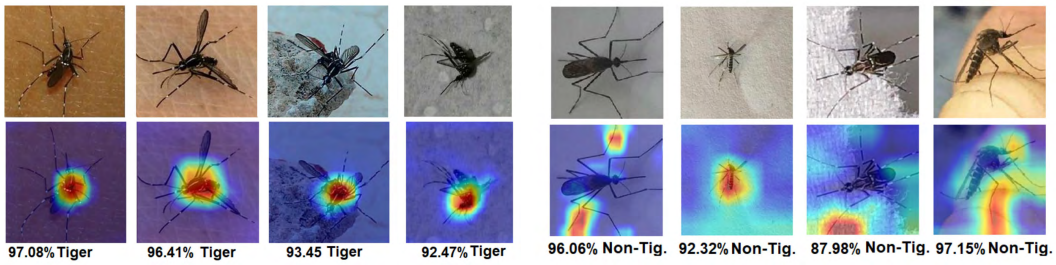



**Figure 4.8:** Anatomy of *Aedes albopictus* mosquito. Image source - Biogents USA.



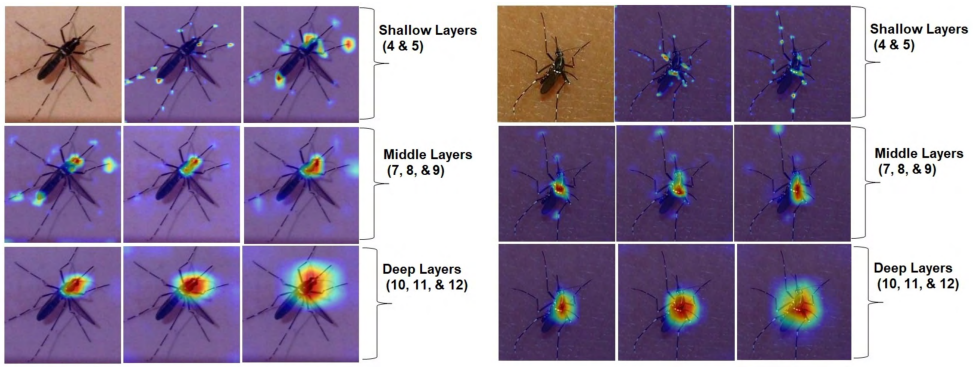

**(a)** Tiger mosquito **(b)** Non-tiger mosquito

**Figure 4.9:** Examples of Grad-CAMs for the (a) tiger and (b) non-tiger mosquitoes species generated from the last convolutional layer.

Figure 4.10 shows an example of a tiger mosquito image and the corresponding heatmaps at the shallow, middle, and deep layers. We observe that the shallow layers emphasised the legs, antennae, and proboscis. The middle layers highlight the head, thorax, and abdomen, which are important features for species identification. The deep layer highlights the area around the thorax and the white stripes around the abdomen. The heatmaps show that the model learns to focus on similar features as the entomologists do. The thorax are the most reliable feature for identifying the tiger mosquito, while the other two features are used to confirm or reject the classification.

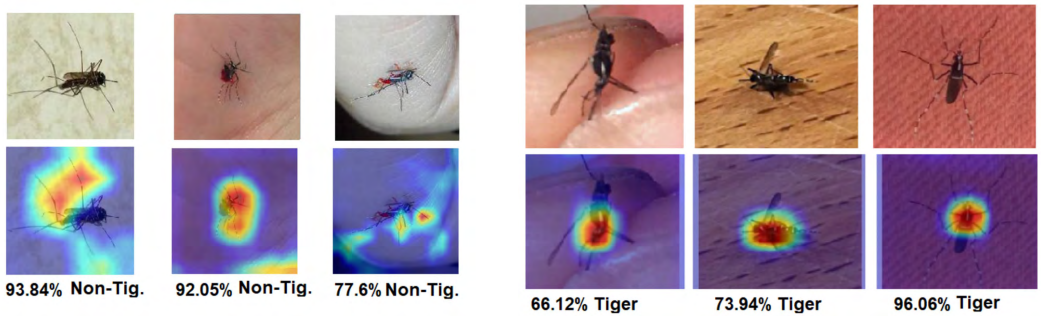
We then investigate the cases where the model makes incorrect predictions and analyse the reasons for the errors. Figure 4.11 shows some examples of misclassified images along with their Grad-CAMs and prediction scores. The model misses the tiger mosquito features when they are damaged or occluded, especially the legs and thorax (Figure 4.12). It also confuses some non-tiger mosquito features, such as striped legs and abdominal patches, with those of the tiger mosquito (Figure 4.13).

While Grad-CAM offers valuable insights into the regions of an image that the



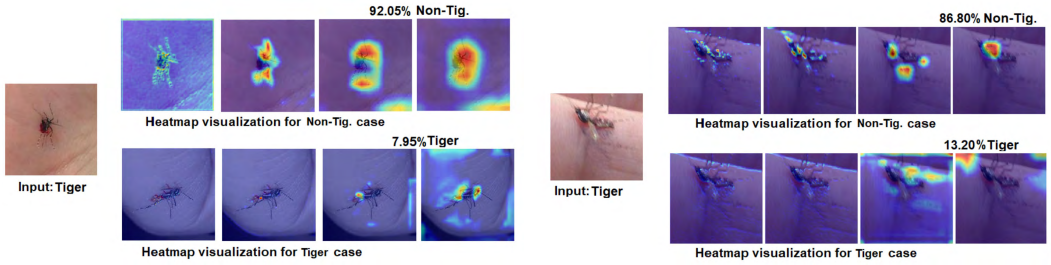

(a) (b)

**Figure 4.10:** Visualisation of discriminative regions of images predicted as tiger mosquitoes; the white stripes in the legs and antennae are slightly highlighted in the shallow layers, whereas the thorax is strongly highlighted in the middle and deeper layers.




(a) (b)

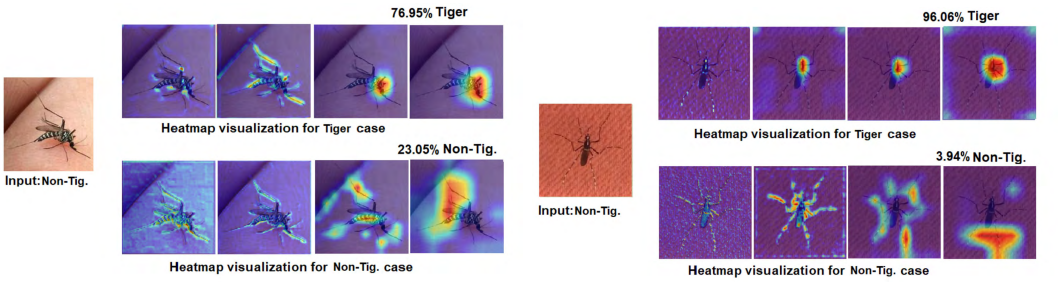
**Figure 4.11:** (a) Examples of tiger input images predicted as non-tiger. (b) Examples of non-tiger input images predicted as tiger. The Grad-CAMs were generated from the last convolution layer.




(a) (b)

**Figure 4.12:** Discriminative regions for tiger images predicted as non-tiger in the shallow, middle and deeper layers.

model focuses on, it does not identify regions contributing to model uncertainty. To address this limitation, we employ another explainability technique known as B-LRP [130]. B-LRP provides a pixel-wise relevance score, highlighting both the regions crucial for the model's prediction as well as those that contribute to the model




**(a)** **(b)**

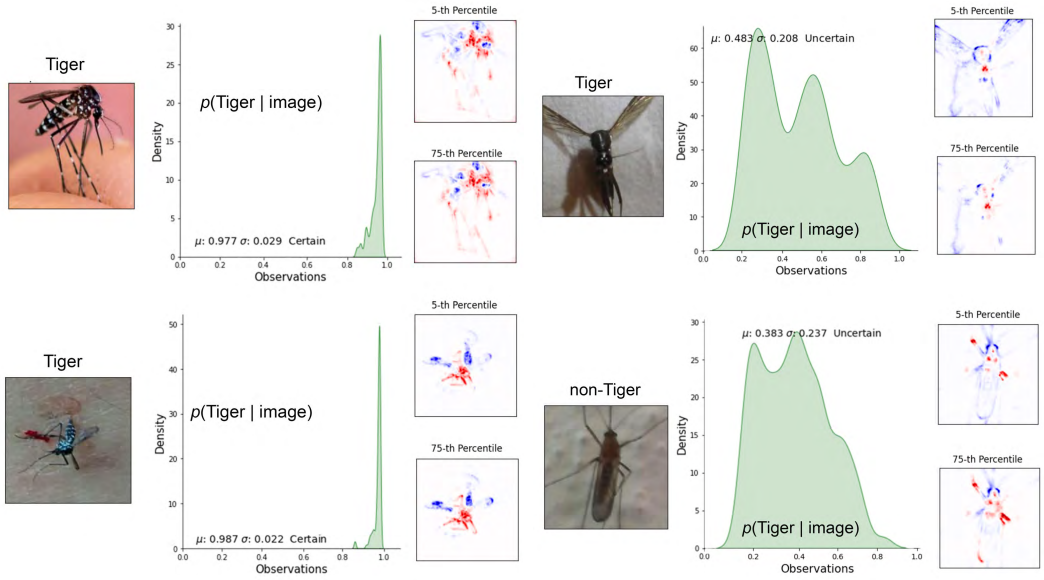
**Figure 4.13:** Visualisation of discriminative regions for non-tiger images predicted as tiger in the shallow, middle and deeper layers and their respective accuracy.

uncertainty. Using this method, we aim to investigate the factors that contribute to uncertainty and errors in the model's decision-making process, presenting specific examples for clarity.

In the B-LRP framework, each pixel's relevance score measures its contribution to the model's prediction. Higher scores mean more important pixels, while lower scores mean uncertain pixels. We can filter the pixels by a percentile or threshold to show only those above it. For example, the $5^{th}$ percentile shows the top 5% of critical pixels, while the $75^{th}$ percentile shows the top 75% of pixels, some of which may be uncertain or less important. The percentile choice affects the explanation risk, with lower percentiles being more conservative and reliable, and higher percentiles being more inclusive but riskier.

Figure 4.14 displays some examples of B-LRP visualisations for different images. It illustrates that when the model is confident about its prediction, for example in Figure 4.14(a), the $75^{th}$ percentile has mostly positive relevance scores for the pixels that are important for identifying the tiger mosquito, like the thorax and abdomen. The pixels that are irrelevant or can cause confusion, like the background or reflections, have negative relevance scores. However, when the model is uncertain about its prediction, for example in Figure 4.14(b), the $75^{th}$ percentile has more mixed relevance scores for the pixels that are not very distinctive or informative for identifying the mosquito species, like the wings or legs. The $5^{th}$ percentile also has some pixels with low or negative relevance scores, indicating that the model is not very certain about them.

We used Grad-CAM and Bayesian Layer-Wise Relevance Propagation (B-LRP) to explain the model's decisions in mosquito species classification. Both methods proved that when the model is certain about its decision, it mainly focuses on the parts of the mosquito features similar to those used by entomologists. B-LRP quantified pixel-wise relevance and captured the model's uncertainty, revealing the factors that influenced the model's confidence and errors. Grad-CAM generated heatmaps that highlighted the model's regions of interest, without requiring a Bayesian framework or a pixel-ranking mechanism. These two explainability tools helped us visualise the relevant anatomical features for identifying mosquito species.




**(a)** **(b)**

**Figure 4.14:** Examples of B-LRP [130] explanation. (a) Two samples of *Aedes albopictus* (Tiger) mosquitoes where the model is certain (confident) in its prediction. The B-LRP highlights the thorax and legs as key discriminant features in the $5^{th}$ percentile and it highlights the regions that can be referred to as the second discriminant features in the $75^{th}$ percentile. (b) Samples of *Aedes albopictus* and non-*Aedes albopictus* (non-Tiger) mosquitoes, where the model is uncertain in its prediction. Here, the B-LRP in the $5^{th}$ and $75^{th}$ is unable to highlight discriminative regions. Red pixels indicate positive relevance, and blue pixels represent negative relevance.

## 4.3 Discussion and Future Work

In this chapter, we presented a framework that uses MC dropout as a technique to estimate model uncertainty and uses it as an acquisition function to design an active learning framework and classification with rejection. The proposed method was applied to automated mosquito species classification, which has significant implications for entomology and public health. The approach allows domain experts to focus only on labelling the most uncertain samples, while the DNN model takes responsibility for making decisions on more certain predictions. This configuration improves the overall reliability and trustworthiness of the classifier.

The proposed method achieved promising performance in terms of accuracy, reliability, and efficiency. Model uncertainty estimated using MC dropout allowed the model to train and achieve baseline performance with fewer training samples. Using the uncertainty during inference time and allowing the trained model to abstain from making predictions on uncertain or ambiguous samples enhanced its performance and reduced the risk of errors or bias. The ability to accurately quantify and utilise uncertainty is crucial for DNN models to avoid making inaccurate predictions, particularly

when faced with inputs that are out of distribution, noisy, or adversarial. Moreover, we also provide visual explainability to support the domain experts and users in understanding the basis of the model's decisions and the sources of uncertainty.

However, the proposed methodology does come with limitations that require future attention. One of these limitations is that the uncertainty estimation using MC dropout does not provide a principled way to select the dropout probability, which may influence the quality and reliability of the uncertainty estimates. A possible solution to this issue could be using Bayesian optimisation to determine the most suitable dropout rates for each model layer. Another limitation is that providing uncertainty estimates is insufficient to explain how the model makes its decisions. Therefore, a technique that explains the internal model and decision-making process is required. In the next chapters, we will address this challenge by providing a more comprehensive explainability method capable of explaining the internal decision-making of a DNN model.

## 4.4 Conclusion

This chapter presents a framework that uses uncertainty estimation based on MC dropout for active learning and classification with rejection. We applied the proposed method to the task of automated mosquito species classification—a complex and crucial task in the fields of entomology and public health. Through empirical evaluation, we have demonstrated that our approach improves the model's accuracy, reliability, and efficiency and reduces the costs associated with labelling specimens collected from citizens. The capability to reject uncertain or ambiguous images offers significant progress to improve AI decision making. The contribution of this chapter is twofold: firstly, it advances the state-of-the-art in classification with rejection and active learning by introducing a novel framework that leverages uncertainty estimation through Monte Carlo (MC) dropout. This advancement enables more precise and confident decision-making capabilities in models. Secondly, it augments the transparency and safety of DNNs by providing a mechanism to acknowledge their limitations in the form of rejection based on uncertainty. This acknowledgement allows for more responsible deployment of artificial intelligence in critical domains such as public health and entomology. The proposed approach has potential applications for entomologists and public health practitioners who require accurate and efficient classification of mosquito species.

# Chapter 5

## ADVISE: A Novel Approach to Quantify and Visualise Feature-Relevance

Convolutional Neural Networks (CNNs) have gained significant prominence with the potential to outperform expectations in various computer vision tasks such as image classification [30, 128, 138, 139, 140], pattern detection [141, 142], semantic segmentation [143, 144], image captioning [145], and human behaviour analysis [146]. However, this sub-symbolism (also known as the opaque or black-box model) is vulnerable to the underlying barrier of *explainability* in response to critical questions like how a particular trained model arrives at a decision, how certain it is about its decision, if and when it can be trusted, why it makes certain mistakes, and in which part of the learning algorithm or parametric space correction should take place [5, 6, 29]. Majority of explainability techniques in CNNs are linked to post-hoc explainability [147] and, as proposed by Arrieta et al. [6], relies on model simplification [97, 148, 149], feature-relevance estimation [20, 150, 151, 152], visualisation [85, 90, 93, 153, 154, 155], and architectural modification [156, 157, 158] to convert a non-interpretable model into an explainable one.

While model simplification and architectural modification techniques have been used to make CNNs interpretable, their associated complexity grows as the number of layers and parameters increases. Furthermore, several studies [6, 159, 160] have shown that altering CNNs may result in the spontaneous appearance of a disentangled representation [161, 162], which is not only unrelated to the model's initial intention but also challenging to interpret. As a result, the emphasis in explaining CNNs has shifted toward feature-relevance and visualisation methods.

Feature visualisation has received much attention because human cognitive skills favour the understanding of visual data. However, feature visualisation methods do not necessarily provide a comprehensive level of explainability and interpretability. Even for high-performance image classification models like VGG16 [163] and Xception [164], surpassing human-level accuracy, their feature visualisations may differ significantly despite generating similar predictions for a given input image.

Therefore, several studies [20, 150, 165, 166] focused on feature-relevance approaches, which provide an importance score to each feature for specific input. These approaches have both advantages and disadvantages. On the positive side, they provide direct information about the relevance of input features for the model's decision, which can help build trust and interpretability in the model. These approaches are also generally model-agnostic, meaning they can be applied to various machine-learning models. However, there are also some downsides to these approaches. They

can be computationally expensive and may not provide a complete understanding of how the model makes its decision. In addition, they can be sensitive to feature correlations and may not account for complex interactions between features. Overall, feature-relevance approaches should be used in conjunction with other methods to gain a more comprehensive understanding of the model's behaviour.

In this chapter, we propose a method for quantifying the feature-relevance and visualising the latent representations in CNNs. We revisit the relationships between feature maps[1] and their associated gradients by introducing **AD**aptive **VIS**ual **E**xplanation (ADVISE). ADVISE estimates the kernel density of gradients with an adaptive bandwidth for each unit in the feature map (see Figures 5.1) to assign an importance score ($\phi_k(A)$) to each unit. Then, we calculate the cumulative gradient of units with the same importance score for the class of interest to visualise the feature map. In this way, we simultaneously quantify the relevance of each unit and highlight how much the cumulative gradient of units influences the model's decision using the generated saliency map(s). We use the proposed method to demonstrate that individual units are significantly more interpretable than cumulative linear combinations of gradient's units. Furthermore, we propose a new evaluation metrics that measures the quality and effectiveness of visual explanations.

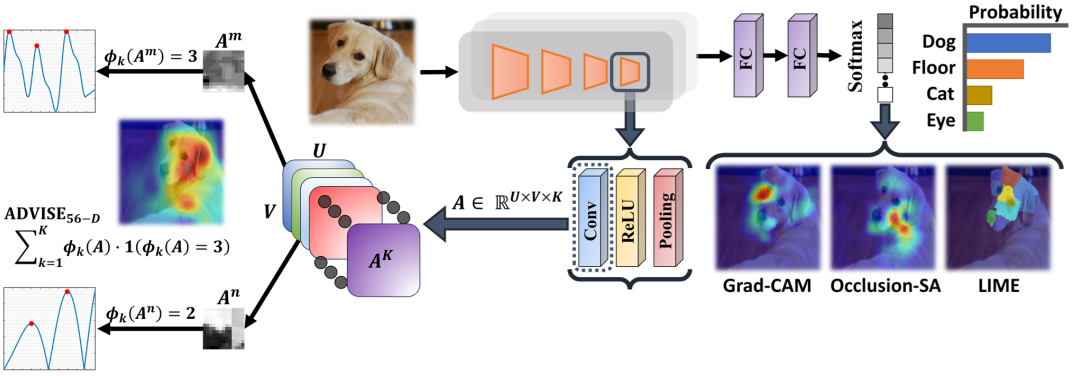



**Figure 5.1:** A schematic of the proposed scoring method. The proposed method assigns an importance score $\phi_k(A)$ to the $k^{\text{th}}$ unit in the feature map $A$. This score represents the contribution of each feature map to the network decision. Using the computed scores (weights), we then calculate the linear *weighted* sum of the feature maps in $A$.

The experiment is centred on the image classification task since it allows us to visualise adaptive cumulative gradient attributions and compare ADVISE with attention approaches that focus on global information. We use AlexNet [135], VGG16 [163], ResNet50 [167], and Xception [164], which were trained on the ImageNet [168] in order to decide to which of 1000 classes each image belongs. We should note that estimating the kernel density of gradients with the adaptive bandwidth can be applied to a wide range of deep learning models without requiring architectural changes or retraining.

[1]The terms *feature map* and *activation map* are used interchangeable here since the former refers to a mapping of where a specific type of feature can be found in an image, and the latter is a mapping that relates to the activation of different areas of the image.

## 5.1 Proposed Methods

We present a novel approach to overcome the drawbacks of existing methods for visual explanation of CNNs. One of the challenges with gradient-based visual explainability methods stem from the oversimplified assumption of a fixed rate for accumulating gradients across feature maps. Our approach, ADVISE, uses Kernel Density Estimation (KDE) [169] and Adaptive Mean Integrated Squared Error (AMISE) [170] criterion to perform a more refined analysis of feature map relevance. KDE is a non-parametric method to estimate the probability density function of a random variable, and AMISE is a criterion to optimise the bandwidth parameter of KDE. Using these approaches, we estimate the probabilistic density of each unit and their contributions, thereby providing a more nuanced and precise quantification of their impact on the model's decisions. In addition, we propose an evaluation protocol to quantitatively assess visual explainability generated by various explanation techniques. The proposed metrics measure the correlation between the importance scores and the saliency maps generated by various explanation techniques.

### 5.1.1 KDE and adaptive bandwidth selection

Kernel Density Estimation (KDE) [169] is a non-parametric technique for estimating the probability density function of a random variable. It involves smoothing the observed data points with a kernel function, usually a symmetric and unimodal function that integrates into one. The estimated density's quality depends on the kernel function's choice and bandwidth, which determines the degree of smoothing. A common approach is to use a fixed bandwidth for all data points, which may result in under-smoothing or over-smoothing in some regions of the data space. However, this approach may not capture the local density variations in the data, which can affect the accuracy and reliability of the density estimation. Therefore, a more flexible and accurate approach is to use adaptive KDE methods, which allow the bandwidth to vary according to the local density of the data. This way, the bandwidth can be larger in regions with low density and smaller in regions with high density. Adaptive KDE methods have been applied to various fields, such as image processing, clustering, anomaly detection, and edge detection.

Let $x_1, x_2, \cdots, x_n$ be a set of independent and identically distributed gradient values from an unknown density function $f(x)$. The kernel density estimate $\hat{f}(x)$ of $f(x)$ is given by:

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K_{h_x}(x - x_i), \tag{5.1}$$

where $K_{h_x}$ is a kernel function with a bandwidth $h_x$ that determines the degree of smoothing. A common choice for the kernel function is the Gaussian kernel, which is defined as:

$$K_{h_x}(s) = \frac{1}{\sqrt{2\pi h_x}} \exp\left(-\frac{s^2}{2h_x^2}\right), \tag{5.2}$$

where $s$ represents the distance between the sample point $x$ and a given point $x_i$ from the dataset, and the term $\sqrt{2\pi h_x}$ normalises the Gaussian function, ensuring that its integral over its domain is 1. The bandwidth $h_x$ can be either fixed or adaptive.

The adaptive mean integrated squared error (AMISE) is a sophisticated method for bandwidth selection in kernel density estimation. It allows adjusting the bandwidth according to the local density variations in the data. The method optimises the bandwidth parameter within local regions, thereby mitigating the expected $L_2$ loss between the estimated and true density functions. Adjusting bandwidth is particularly advantageous when dealing with datasets that exhibit heterogeneous density characteristics, as it permits the bandwidth to expand in sparser regions and contract in denser regions for enhanced accuracy. The AMISE is formulated as follows:

$$\text{AMISE} = \int \mathbb{E}\left(\hat{f}(x) - f(x)\right)^2 \rho_W^{u-x} \, \mathrm{d}x, \tag{5.3}$$

where $\mathbb{E}$ denotes the expected value, $\hat{f}(x)$ the estimated density function, $f(x)$ the true density function, and $\rho_W^{u-x}$ a weight function that regulates the bandwidth based on the local density within an interval $W$ centred at $x$. The quality and reliability of the density estimation are improved by optimising the bandwidth in a localised manner with the AMISE criterion.

### 5.1.2 ADVISE: ADaptive VISual Explanation

We consider a convolutional neural network that performs image classification, which is defined as $f(I;\theta) = \mathbb{E}[y^c|I;\theta]$, where $\theta$ denotes the network's parameters, $I \in \mathbb{R}^{H\times W\times 3}$ is the input image with height $H$ and width $W$, and $y^c$ is the score for the predicted class $c$. The activation map of $f$ is represented by $A \in \mathbb{R}^{U\times V\times K}$, where $A^k$ is the $k^{\text{th}}$ feature map in $A$, and $U$, $V$, and $K$ are the height, width, and the number of units, respectively. The gradient of the predicted score $y^c$ with respect to the spatial location $(i, j)$ in the feature map $A$ is given by $\frac{\partial y^c}{\partial A_{i,j}}$.

Most visualisation methods use cumulative gradients, which is a linear weighted summation of all feature maps in $A$. However, these methods lack sensitivity because they assume a stationary rate variation in the gradients. To address this issue, we propose a new method that computes $\phi_k(A)$, an importance score assigned to the $k^{\text{th}}$ unit in the feature map $A$. This score represents the contribution of each feature map to the network decision. We then compute the linear *weighted* sum of the feature maps in $A$ using the same importance score. In this way, our visualisation method preserves both implementation invariance and sensitivity axioms [19]. These sensitivity axioms are two essential properties that any attribution method should adhere to:

- **Sensitivity**: If the function implemented by a deep network does not mathematically depend on a certain variable, then the attribution to that variable should always be zero. This axiom captures the desired insensitivity of attributions when the function does not depend on a specific variable.

- **Implementation invariance**: This axiom states that the attribution method should be invariant to the implementation details of the network, focusing solely on the function computed by the network rather than how it is implemented. This ensures that the attribution method is robust and not affected by changes in the network's implementation.

Our approach satisfies these sensitivity axioms by assigning and using non-zero importance scores to the feature maps that contain the differing or perturbed features in the linear sum. These axioms ensure that the attribution method is sensitive to changes in the input features that affect the output prediction.

We introduce a method that leverages KDE [169] as a key component in a pipeline to assess the importance score of each feature map. Specifically, KDE estimates the density of gradient values, which indicate how sensitive the output is to variations in the input.These gradients reveal the contribution of each feature map to the final model prediction. However, the shape of the estimated density may vary due to the intrinsic difference of each unit in the feature map, and conventional KDE may not capture this variation. Therefore, we use the AMISE criterion to optimise the estimated density locally in an interval length. This criterion helps determine the goodness-of-fit and regulate the estimated density's shape.

The $k^{\text{th}}$ unit of the activation map $A$ is made up of a collection of independent gradients that have been flattened into a one-dimensional array. The gradients values, $(a_1, a_2, \cdots, a_n)$, change with respect to the input image $I$. In order to determine the raw density of the gradient values, $x_a$, we must calculate the average of the Dirac delta function $\delta(a)$ for each gradient value $a_i$. This raw density is represented by Eq. 5.4.

$$x_a = \frac{1}{n} \sum_{i=1}^{n} \delta(a - a_i), \tag{5.4}$$

where $n = U \times V$. To estimate the kernel density $\widehat{\lambda}_a$, we convolve the raw density $x_a$ with a kernel $\mathcal{H}_{\omega_a}$ that has a variable bandwidth $\omega_a$, optimising over a local interval. This operation is represented in Eq. 5.5.

$$\widehat{\lambda}_a = \int_{-\infty}^{\infty} x_{a-s} \mathcal{H}_{\omega_a}(s)\, \mathrm{d}s. \tag{5.5}$$

We evaluate the estimated density $\widehat{\lambda}_a$ for goodness-of-fit by comparing it to the unknown underlying density $\lambda_a$ using the mean integrated squared error (MISE) [171]. To regulate the shape of the function $\lambda_a$, determine the goodness-of-fit, and select an interval length for local optimisation at gradient $a$, we introduce the adaptive MISE (AMISE) criterion based on [170]. The AMISE criterion is expressed in Eq. 5.6.

$$\text{AMISE} = \int \mathbb{E}\left(\widehat{\lambda}_u - \lambda_u\right)^2 \rho_W^{u-a}\, \mathrm{d}u, \tag{5.6}$$

where $\mathbb{E}$ represents the expected $L_2$ loss function, $\widehat{\lambda}_u$ is the estimated density with a

fixed bandwidth $\omega$, $\rho_W^{u-a}$ is a weight function that determines the integration of the squared error in a particular interval $W$ centred at $a$.

Our objective is to minimise AMISE by introducing an adaptive cost function with respect to $a$. This adaptive cost function eliminates the irrelevant term for the bandwidth $\omega$ choice. This adaptive cost function is expressed in Eq. 5.7.

$$C_n^a(\omega, W) = \text{AMISE} - \int \lambda_u^2 \rho_W^{u-a} \, \mathrm{d}u. \tag{5.7}$$

Then, the optimal fixed bandwidth, denoted by $\omega^*$, is determined by minimising the estimated cost function in Eq. 5.8.

$$\hat{C}_n^a(\omega, W) = \frac{1}{n^2} \sum_{i,j} \psi_{\omega,W}^a(a_i, a_j) - \frac{2}{n^2} \sum_{i \neq j}^{n} \mathcal{H}_\omega(a_i - a_j) \rho_W^{a_i - a}, \tag{5.8}$$

This cost function has two parts. The first is the average of the kernel $\psi_{\omega,W}^a(a_i, a_j)$ across all observation pairs $i$ and $j$. The second is a correction term that considers overlapping regions of the kernel. Eq. 5.9 defines the kernel $\psi_{\omega,W}^a$, which includes a convolution of the kernel function $\mathcal{H}_\omega(u-a)$ with the density function $\rho_W^{u-a}$.

$$\psi_{\omega,W}^a(a_i, a_j) = \int \mathcal{H}_\omega(u - a_i) \mathcal{H}_\omega(u - a_j) \rho_W^{u-a} \, \mathrm{d}u. \tag{5.9}$$

In our experiments, we use an interval length of $\frac{\omega^*}{\gamma}$ to control the degree of fluctuation in the variable bandwidth since the optimal bandwidth $\omega^*$ varies according to the length of the interval $W$. The smoothing parameter $\gamma$ determines the extent of the fluctuations, with smaller values, $\gamma << 1$, leading to minor fluctuations and larger values, $\gamma \sim 1$, resulting in more significant fluctuations. The Nadaraya-Watson kernel regression [172] is used to obtain a variable bandwidth $\omega_a^\gamma$ using Eq. 5.10 for the interval $[0, 1]$.

$$\omega_a^\gamma = \int \rho_{W_s^\gamma}^{a-s} \bar{\omega}_s^\gamma \, \mathrm{d}s \Big/ \int \rho_{W_s^\gamma}^{a-s} \, \mathrm{d}s \, . \tag{5.10}$$

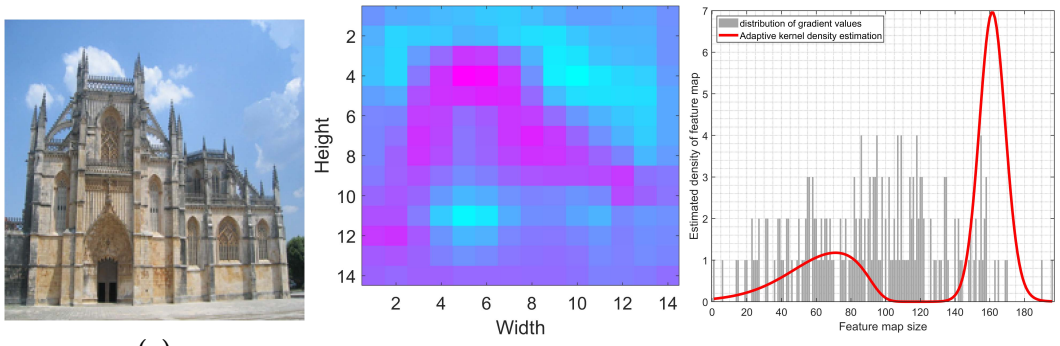
Here, $W_a^\gamma$ and $\bar{\omega}_a^\gamma$ represent the interval length and fixed bandwidth at $a$, respectively. The variable bandwidth $\omega_a^\gamma$ is determined from the same data, but different $\gamma$ values result in varying levels of smoothness. The cost function for the variable bandwidth selected with $\gamma$ is obtained using Eq. 5.11.

$$\hat{C}_n(\gamma) = \int_0^1 \hat{\lambda}_a^2 \, \mathrm{d}a - \frac{2}{n^2} \sum_{i \neq j} \mathcal{H}_{\omega_{a_i}^\gamma}(a_i - a_j), \tag{5.11}$$

The estimated rate $\hat{\lambda}_a$ with the variable bandwidth $\omega_a^\gamma$ is calculated numerically with the stiffness constant $\gamma^* = \frac{\sqrt{5}+1}{2}$ that minimises Eq. 5.11. In this study, we use the Gauss density function expressed in Eq. 5.12.

$$\mathcal{H}_{\omega^\gamma}(s) = \frac{1}{\sqrt{2\pi\omega^\gamma}} \exp\left(-\frac{s^2}{2(\omega^\gamma)^2}\right), \tag{5.12}$$

Figure 5.2b depicts one of the activation map units in the final convolution layer of the VGG16 model. Additionally, Figure 5.2c displays the estimated density of gradient values (shown as solid red line) and the underlying gradient value distribution (depicted as a grey area) at that specific unit. The density of gradient values is estimated using the proposed variable bandwidth kernel density estimation method.




**(a)** **(b)** **(c)**

**Figure 5.2:** (a) Input image. (b) The 265$^{\text{th}}$ unit of the activation map in the last convolution layer of the VGG16 model. The gradient values are represented by colours in the 'cool' colour map to aid visualisation. (c) The estimated kernel density with a variable bandwidth (solid red line) using Eq. 5.11. The grey area displays the underlying distribution of the gradient values in the 265$^{\text{th}}$ unit of the activation map.

The proposed scoring method that assigns an importance score to the $k^{\text{th}}$ unit in the feature map, as well as the visualisation approach (ADVISE), are summarised in Algorithm 1.

The results from using ADVISE to generate the saliency map and score function $\phi_k(A)$ to quantify feature-relevance can be summarised into three key observations.

1. Not all feature map units contribute equally to the model's prediction, and some units may even mislead the model. For example, in Figure 5.3a, saliency maps generated by ADVISE for units with 1, 6, and 7 peaks, which make up 119 out of 256 AlexNet units, highlight uninformative regions of the image with respect to the predicted class (i.e., 'Bernese mountain dog'). On the other hand, the generated saliency map with only 2 units with 2 peaks performs better than Grad-CAM, which needs the incorporation of all 256 AlexNet units.

2. As Bau et al. [159] pointed out, CNNs trained for a specific purpose may encounter disentangled representations unrelated to the model's initial intention. For example, Grad-CAM generated saliency maps of the VGG16 network in Figure 5.3b show highlights in regions unrelated to the predicted 'monastery' class, while 4 units with 6 peaks in ADVISE highlight more portions of the building.

3. When different visual explainability methods show less divergence, using ADVISE can assist developers in determining which layers contribute the most to

**Algorithm 1** ADaptive VISual Explanation

**Require:** $A^{U\times V\times K}$ – Feature map, also known as *activation map* in CNNs.
$y^c$ – predicted class.
[row, col] – size of input image.

**Ensure:** $\phi_k(A)$ – Importance score for units in $A$.
ADVISE – Feature saliency map(s).

**for** $k = 1$ to $K$ **do**
  $\{a_i\}_{i=1}^{n} \leftarrow$ **flatten**$(A)$ ▷ $n = U \times V$
  $\phi_k(A) =$ **findPeaks** $\left(\int_0^1 \hat{\lambda}_a^2 \, \mathrm{d}a - \frac{2}{n^2}\sum_{i\neq j} \mathcal{H}_{\omega_{a_i}^{\gamma}}(a_i - a_j)\right)$

$g = \frac{\partial y^c}{\partial A}$
**for** $i = \min(\phi_k(A))$ to $\max(\phi_k(A))$ **do**
  $\mathrm{idx} \leftarrow$ **find** $(\phi_k(A) == i)$
  $\tilde{A}_i = A(:, :, \mathrm{idx})$
  $\tilde{w}_i^c = \frac{1}{n}\sum_U \sum_V g(:, :, \mathrm{idx})$
  $\mathrm{map}_i = \mathrm{ReLU}\left(\sum_{j=1}^{|\mathrm{idx}|}\left(\tilde{w}_{i,j}^c \cdot \tilde{A}_{i,j}\right)\right)$ ▷ $|\bullet|$ is the cardinality of $\bullet$
  $\mathrm{ADVISE}_i =$ **resize** $(\mathrm{map}_i, [\mathrm{row}, \mathrm{col}], \mathrm{bc})$ ▷ 'bc' is bicubic interpolation

**return** $\phi_k(A)$, ADVISE

the final model's prediction. In Figures 5.3c and 5.3d, ADVISE shows that ResNet50 and Xception require 177 units with 1 peak and 8 units with 3 peaks, respectively, for generating relatively accurate saliency maps. However, the Grad-CAM method requires incorporating all 2048 units to generate comparable or even less salient maps.

### 5.1.3 Evaluation metrics

Evaluating visual explainability is not a trivial task, as it involves both objective and subjective criteria that may vary depending on the context and the user. Moreover, different visual explanation methods may have different strengths and weaknesses and highlight different aspects of the model's decision process. Therefore, we need a comprehensive and consistent evaluation framework that can measure the quality and effectiveness of visual explanations from different perspectives. We use existing and new metrics that measure different aspects of visual explanation quality, such as class sensitivity, hit rate, and confidence drop.

**(1) Class Sensitivity (CS):** it measures the similarity of saliency maps generated with respect to the top two class scores predicted by the model. It uses Pearson's Correlation Coefficient to measure CS as in Eq. 5.13.

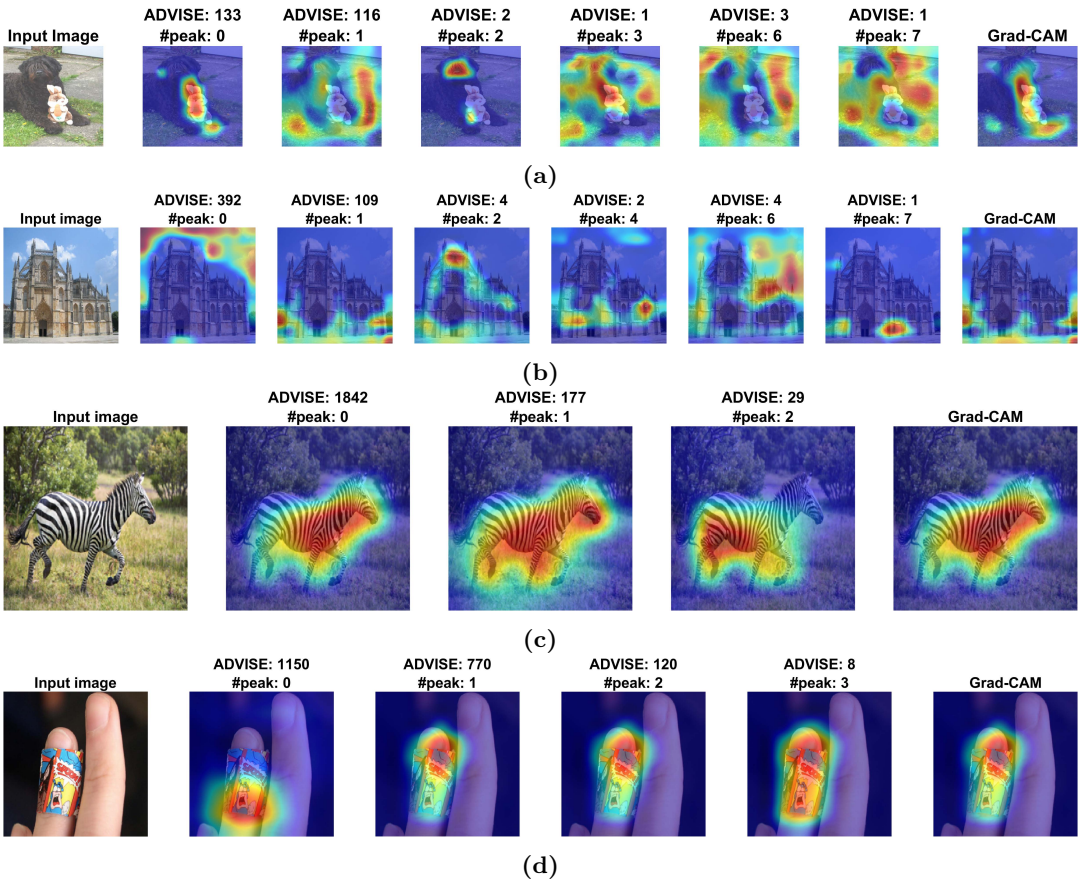



**Figure 5.3:** The outputs of ADVISE and Grad-CAM [90] are compared for four images fed into the pretrained AlexNet [135], VGG16 [163], ResNet50 [167], and Xception [164] models on ImageNet [168]. The use of $\phi_k(A)$ on the estimated kernel density and ADVISE show that in the explainability of (a) AlexNet prediction ('Bernese mountain dog'), two units with two peaks work better than Grad-CAM that requires 256 units, (b) VGG16 prediction ('monastery'), four units with six peaks contribute more than Grad-CAM that requires 512 units, (c) ResNet50 prediction ('Zebra'), 177 units with one peak outperform Grad-CAM, which requires 2048 units, and (d) Xception prediction ('band aid'), eight units with three peaks perform better than Grad-CAM that uses 2048 units.

$$\mathrm{CS} = \frac{\mathrm{cov}\left(E(f,I)^{c_1}, E(f,I)^{c_2}\right)}{\sigma\left(E(f,I)^{c_1}\right) \times \sigma\left(E(f,I)^{c_2}\right)}. \tag{5.13}$$

where $E$, cov, and $\sigma$ denote the explanation map, covariance, and standard deviation, respectively. A good explanation method should have a score near to or below zero, while a score outside the $[-0.5, 0.5]$ range implies that the correlation between two maps is not statistically significant.

**(2) Hit:** it is a proxy that indicates if the model can retrieve the target class $c$ in its top-5 prediction when it just sees the explanation map and not the entire image. This proxy is formulated in Eq. 5.14.

$$\mathrm{Hit} = \begin{cases} 1 & : \ N_I \cap M_{I \odot E(f,I)^c} \\ 0 & : \ \text{otherwise} \end{cases} \tag{5.14}$$

where $N_I$ is the index of the predicted class $c$ by the model when it just sees the input image as input, and $M_{I \odot E(f,I)^c}$ is a set including the top-5 index of the predicted class when the model sees the explanation map. Here, $\odot$ is the Hadamard product.

**(3) Average Drop (AD):** it measures the average percentage drop in confidence for the target class $c$ when the explanation map $(I \odot E(f,I)^c)$ is fed to the model instead of the input image $I$. This metric is defined in Eq. 5.15, where lower is better.

$$\mathrm{AD} = \max\left(0, (y^c - o^c)\right)/y^c \tag{5.15}$$

where $o^c$ is the predicted score by model to which the the explanation map is fed.

However, these existing metrics have some limitations. CS does not consider the spatial distribution of saliency values and may not capture the perceptual differences between saliency maps. Hit is a binary measure that does not reflect the degree of confidence change for the target class. AD only considers the confidence drop for the target class and ignores the confidence changes for other classes. Moreover, these metrics do not account for the structural and feature similarity between the input image and the explanation map, which are important factors for human perception and interpretation.

To address these limitations, we propose new evaluation metrics that measure the Structural Similarity Index (SSIM), Feature Similarity Index (FSIM), and Mean Squared Error (MSE) between the input image masked by the explanation map and the original input image as the reference. These metrics capture different aspects of visual image quality, such as luminance, contrast, structure, phase congruency, gradient magnitude, and pixel-wise difference.

**(4) Structural similarity index (SSIM):** it is a perception-based measure that considers image degradation as a perceived change in structural information while also considering crucial perceptual phenomena [173]. In this context, SSIM measures the structural similarity index between the input image masked by the explanation

map and the input image as the reference. This metric returns a value in $(0, 1]$, where the higher is better, and is formulated in Eq. 5.16.

$$\text{SSIM}(I, \tilde{I}) = \frac{(2\mu_I\mu_{\tilde{I}} + e_1)(2\text{cov}(I, \tilde{I}) + e_2)}{(\mu_I^2 + \mu_{\tilde{I}}^2 + e_1)(\sigma_I^2 + \sigma_{\tilde{I}}^2 + e_2)}. \tag{5.16}$$

where $\tilde{I} = I \odot E(f, I)^c$, and $\mu$ and $\sigma$ are the mean and variance, respectively. In order to stabilise the division with weak denominator, $e_1 = (0.01 \cdot L)^2$ and $e_2 = (0.03 \cdot L)^2$ are used, where $L$ denotes the dynamic range of the pixel values and is set to 255 in this study.

**(5) Feature similarity index (FSIM):** it uses phase congruency and gradient magnitude, which reflect complementary components of visual image quality, to measure local image quality. This metric also includes a saliency measure for the image gradient feature, which weights each pixel's contribution to the overall quality score. This metric returns a value in $(0, 1]$, where the higher is better, and the mathematical formulation is given in 5.17.

$$\text{FSIM}(I, \tilde{I}) = \frac{\sum_{i=1}^{N} w_i \mu_{I_i} \mu_{\tilde{I}_i}}{\sqrt{\sum_{i=1}^{N} w_i \mu_{I_i}^2} \sqrt{\sum_{i=1}^{N} w_i \mu_{\tilde{I}_i}^2}} \times \frac{\sum_{i=1}^{N} w_i \text{cov}_{I,\tilde{I}}^{(i)}}{\sqrt{\sum_{i=1}^{N} w_i \sigma_I^{2(i)}} \sqrt{\sum_{i=1}^{N} w_i \sigma_{\tilde{I}}^{2(i)}}} \tag{5.17}$$

where $N$ is the number of feature maps, $w_i$ is the weighting function for the $i^{\text{th}}$ feature map, $\mu_{I_i}$ and $\mu_{\tilde{I}_i}$ are the mean values of the $i^{\text{th}}$ feature map in the two images, and $\text{cov}_{I,\tilde{I}}^{(i)}$, $\sigma_I^{2(i)}$, and $\sigma_{\tilde{I}}^{2(i)}$ are the covariance and variances of the $i^{\text{th}}$ feature map in the two images, respectively.

**(6) Mean squared error (MSE):** it is the second error moment and measures the average squared difference between the input image masked by the explanation map and the input image as the reference as in Eq. 5.18.

$$\text{MSE}(I, \tilde{I}) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \left( I_{i,j} - \tilde{I}_{i,j} \right)^2. \tag{5.18}$$

Finally, we propose a new metric that combines AD, SSIM, FSIM, and MSE into a single score that reflects the overall explainability of a model. The proposed metric, called AVerage eXplainability (AVX).

**(7) AVerage eXplainability (AVX):** it measures the harmonic mean of AD, SSIM, FSIM, and MSE and returns a value in $[0, 1]$ to ease of comparison as defined in Eq. 5.19.

$$\text{AVX} = 4 \left( \frac{1}{1 - \text{AD}} + \frac{1}{\text{SSIM}} + \frac{1}{\text{FSIM}} + \frac{1}{1 - \text{MSE}} \right)^{-1} \tag{5.19}$$

We employ two proxy variables, CS and Hit, that operate as regulators to adjust the AVerage eXplainability (AVX) score. These proxies capture distinct aspects of the

model explainability, offering a refined insight into how well the model's predictions correspond with human interpretability.

- **If** Hit $= 0$ **and** CS $\in [-0.5, 0.5]$: We define a penalty coefficient $\Delta = 1 - |y^c - o^c|$ and multiply AD, SSIM, FSIM, and MSE by $\Delta$ before measuring the harmonic mean. The purpose of this penalty coefficient $\Delta$ is to be to introduce a modulation factor for the existing metrics (AD, SSIM, FSIM, MSE) when Hit $= 0$ and CS $\in [-0.5, 0.5]$. In this case, the Hit $= 0$ condition indicates that the model failed to identify the target class $c$ in its top-5 predictions when provided with the explanation map alone. Meanwhile, the CS $\in [-0.5, 0.5]$ condition indicates that the correlation between the saliency maps for the top two classes predicted by the model is within an acceptable range. When both these conditions are met, it suggests that the model's failure is not due to grossly inaccurate or misleading saliency maps. Hence, the $\Delta$ penalty scales down the metrics to potentially lower their impact on the final AVX score, allowing for a more nuanced evaluation. The design of $\Delta$ helps to account for cases where the model's poor performance is not necessarily because of the explanation's quality but could be due to other factors, thus providing a more balanced view of the model's explainability.
- **If** Hit $= 0$ **and** CS $\notin [-0.5, 0.5]$: In this case, the framework applies penalty conditions to indicate significant shortcomings in the model's explainability, and we set AD and MSE to 1 and SSIM and FSIM to 0. Specifically, the condition Hit $= 0$ shows that the model fails to include the target class $c$ among its top-5 predictions when only the explanation map is considered. Moreover, the criterion CS $\notin [-0.5, 0.5]$ implies that the correlation between the saliency maps corresponding to the top two predicted classes falls outside a statistically acceptable range, thus raising concerns of either misleading or ambiguous explanations. To adapt the scoring function to this case, the framework sets the metrics of AD and MSE to their maximum penalty value of 1 while lowering the SSIM and FSIM to their minimum value of 0. These extreme metric values act as a strong indicator, considerably affecting the model's AVX score, thereby flagging the model's explainability as a point of urgent attention requiring further investigation.

## 5.2 Experimental Results

This section presents the experimental results of ADVISE and compares them with the existing state-of-the-art visual explanation methods. We use four popular CNN architectures, AlexNet [135], VGG16 [163], ResNet50 [167], and Xception [164], trained on the ImageNet dataset. We use existing and newly proposed evaluation metrics to measure the quality and effectiveness of visual explanations generated using various XAI methods.

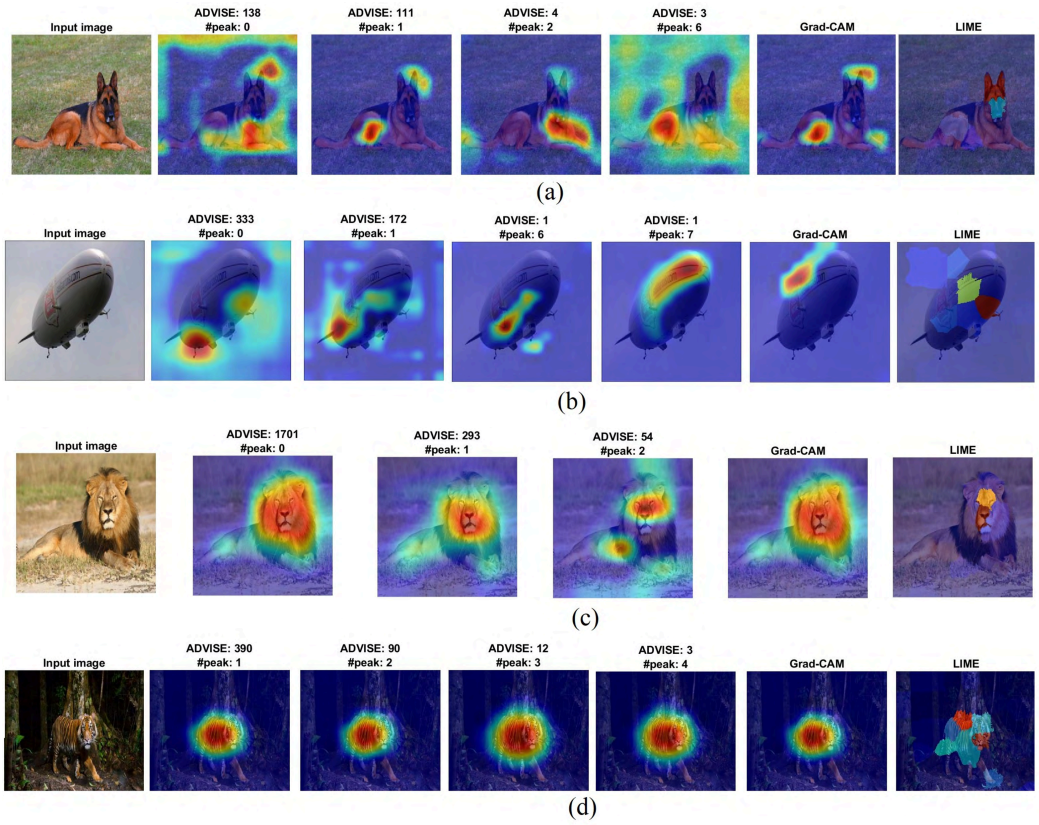



**Figure 5.4:** Visual explanations of four different inputs using (a) AlexNet, (b) VGG16, (c) ResNet50, and (d) Xception models. ADVISE highlighted the key features better than Grad-CAM and LIME.

Figure 5.4 shows the visual explanations of four inputs using AlexNet, VGG16, ResNet50, and Xception models. The proposed method ADVISE outperforms Grad-CAM and LIME in all cases by highlighting the key features with fewer units. For instance, in Figure 5.4(a), ADVISE uses only 111 units with 1 peak to highlight the head and ears of the German shepherd, while Grad-CAM and LIME uses entire features to generate less informative maps. In Figure 5.4(b), ADVISE uses only 1 unit with 7 peaks to highlight the balloon and its reflection, while Grad-CAM and LIME highlight regions unrelated to the air balloon class. In Figure 5.4(c), ADVISE uses only 293 unit with 1 peak to highlight the face and mane of the African lion, while Grad-CAM and LIME highlight more background regions. In Figure 5.4(d), ADVISE uses only 12 units with 3 peaks to highlight the face and stripes of the Asian tiger, while Grad-CAM and LIME highlight less salient regions. ADVISE highlights the salient regions of the input image with fewer units, while Grad-CAM and LIME use more units and generate less informative maps.

Next, we compare and quantify ADVISE with other state-of-the-art visual explanation methods. However, evaluating visual explanations is not a trivial task, as there are no ground-truth discriminative features for a trained CNN [174], and there is no consensus on the impact of explanations on the model's performance, trust,

and reliance. Therefore, we assume that a well-trained model would make predictions based on the features of the object itself [6], and we follow quantitative metrics used to evaluate image retrieval methods, saliency models and the novel evaluation protocol proposed in Eq. 5.19.

**Table 5.1:** The comparison of the ADVISE with Grad-CAM, Grad-CAM++, Score-CAM, and Layer-CAM visualisation methods on AlexNet, VGG16, ResNet50, and Xception.

| Architecture | Method | Metrics | | | | | | Time (s) | |
|---|---|---|---|---|---|---|---|---|---|
| | | Peak range | AD ↓ | SSIM ↑ | FSIM ↑ | MSE ↓ | AVX ↑ | GPU/Parallel | CPU |
| AlexNet [135] | **ADVISE** | 0 – 8 | **0.26** | **0.14** | **0.38** | **0.14** | **0.28** | **0.69** | 30.3 |
| | Grad-CAM | N/A | 0.39 | 0.05 | 0.26 | 0.32 | 0.13 | 1.06 | **1.64** |
| | Grad-CAM++ | N/A | 0.38 | 0.06 | 0.27 | 0.32 | 0.17 | 1.16 | 2.14 |
| | Score-CAM | N/A | 0.37 | 0.06 | 0.28 | 0.31 | 0.17 | 1.18 | 2.60 |
| | Layer-CAM | N/A | 0.33 | 0.07 | 0.31 | 0.28 | 0.19 | 1.48 | 3.33 |
| | LIME | N/A | 0.39 | 0.05 | 0.26 | 0.32 | 0.13 | 5.71 | 11.85 |
| VGG16 [163] | **ADVISE** | 0 – 7 | **0.26** | **0.14** | **0.40** | **0.15** | **0.29** | **1.56** | 6.91 |
| | Grad-CAM | N/A | 0.38 | 0.06 | 0.26 | 0.29 | 0.15 | 1.88 | **2.66** |
| | Grad-CAM++ | N/A | 0.38 | 0.07 | 0.27 | 0.28 | 0.19 | 2.01 | 3.36 |
| | Score-CAM | N/A | 0.37 | 0.09 | 0.30 | 0.29 | 0.22 | 2.21 | 3.87 |
| | Layer-CAM | N/A | 0.32 | 0.09 | 0.34 | 0.27 | 0.23 | 2.66 | 4.24 |
| | LIME | N/A | 0.38 | 0.06 | 0.26 | 0.29 | 0.15 | 22.18 | 57.95 |
| ResNet50 [167] | **ADVISE** | 0 – 5 | **0.26** | **0.15** | **0.43** | **0.17** | **0.31** | **1.46** | **6.37** |
| | Grad-CAM | N/A | 0.33 | 0.10 | 0.34 | 0.24 | 0.23 | 6.22 | 7.77 |
| | Grad-CAM++ | N/A | 0.36 | 0.11 | 0.35 | 0.24 | 0.26 | 6.62 | 8.56 |
| | Score-CAM | N/A | 0.35 | 0.11 | 0.37 | 0.22 | 0.27 | 7.02 | 9.18 |
| | Layer-CAM | N/A | 0.32 | 0.12 | 0.39 | 0.21 | 0.29 | 7.51 | 11.18 |
| | LIME | N/A | 0.33 | 0.10 | 0.34 | 0.24 | 0.23 | 7.68 | 31.61 |
| Xception [164] | **ADVISE** | 0 – 6 | **0.43** | **0.12** | **0.37** | **0.31** | **0.24** | **4.20** | 16.38 |
| | Grad-CAM | N/A | 0.68 | 0.04 | 0.20 | 0.59 | 0.10 | 5.92 | **8.12** |
| | Grad-CAM++ | N/A | 0.65 | 0.04 | 0.21 | 0.59 | 0.11 | 6.03 | 9.10 |
| | Score-CAM | N/A | 0.64 | 0.05 | 0.21 | 0.57 | 0.13 | 6.56 | 9.70 |
| | Layer-CAM | N/A | 0.57 | 0.08 | 0.27 | 0.49 | 0.19 | 7.07 | 10.34 |
| | LIME | N/A | 0.68 | 0.04 | 0.20 | 0.59 | 0.10 | 26.31 | 90.31 |

Table 5.1 shows the comparison of the ADVISE with Grad-CAM [90], Grad-CAM++ [91], Score-CAM [92], and Layer-CAM [93] visualisation methods. The study involved comparing the performance of four pretrained models (AlexNet [135], VGG16 [163], ResNet50 [167], and Xception [164]) on the ImageNet dataset [168] using five evaluation metrics (AD, SSIM, FSIM, MSE, and AVX). The comparison led to two observations. The first observation showed that ADVISE consistently outperformed other methods on all models, indicating that it is a highly competitive visual explainability method. The second observation indicated that Xception, despite having higher classification accuracy than other models on the ImageNet task, exhibited lower efficiency in visual explainability across all metrics when compared to the other pretrained models.

In our quest for this AVX decline in Xception, we examined the saliency maps produced by the ADVISE in shallow, middle, and deep layers (see an example in Figure 5.5). In contrast to the shallow and middle layers, which tend to highlight low-level visual features like edges and blobs distributed throughout the image, the deep layer of the Xception model tends to focus on the centre of the scene. Additionally, the other models we studied have different focal points within the image. This focus is known as the centre bias in saliency studies [175, 176], where most studies revealed that observers prefer to look more often at the centre of the image than at the edges.

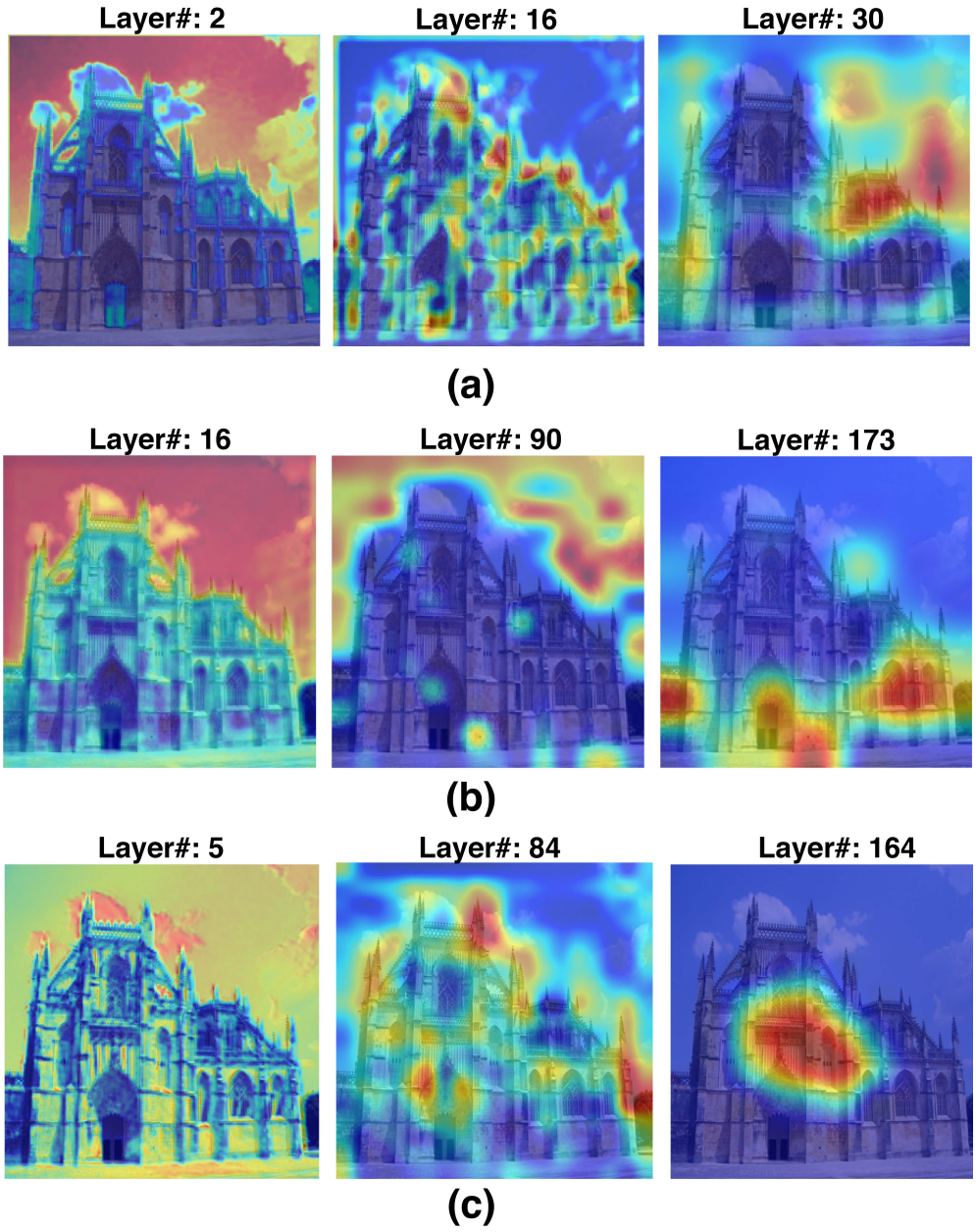



**Figure 5.5:** ADVISE outputs for shallow, middle, and deep layers of (a) VGG16, (b) ResNet50, and (c) Xception pretrained models on ImageNet.

However, Xception model exhibits a notable inclination towards centre bias, a characteristic that carries both merits and drawbacks. While it is more aligned with human cognitive skills for perceiving visual data, as explained by [88], the centre of mass of the saliency map is the Achilles Heel of many visual explanation methods, with path attribution methods offered to address it [19] but failing the sanity checks [177].

A combination of the Xception model's centre bias and the existing XAI methods' limitations seems to account for the reduced visual explainability observed for this

model. The Xception model's tendency to focus on the image centre matches human cognitive patterns, but it also hinders the generation of informative saliency maps. Current XAI techniques, such as path attribution methods, have difficulties adjusting to this bias and often do not pass sanity checks. Thus, the problem stems from an interplay between the model's features and the current explainability techniques' constraints, which requires further research to effectively overcome this challenge.

The specific challenges observed with the Xception model, although the proposed XAI method and quantitative metrics, which are supported by best practices, can evaluate the performance of different models in visual explanation, we still have a fundamental problem with the lack of *ground-truth explanations*. In fact, we aim to determine which methods best explain our model without knowing how it works. Evaluating supervised models is relatively straightforward since we have a test set. However, evaluating explanations is difficult since we do not exactly know how our model works and do not have the ground-truth for a fair comparison.

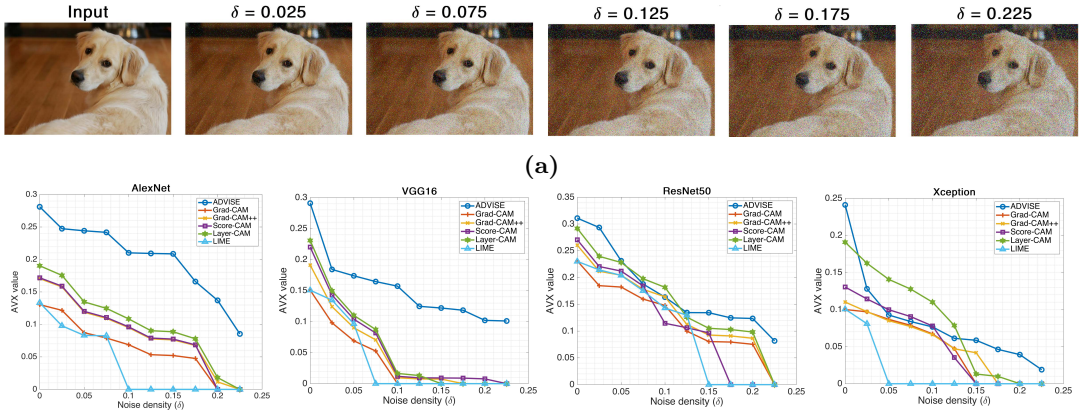



**Figure 5.6:** (a) An ablated image by randomly replacing pixels with the salt and pepper noise with the noise density of $\delta = [0.025, 0.075, 0.125, 0.175, 0.225]$. (b-e) Changes in the performance of the ADVISE and five additional visual explanation methods as a function of (AVX, $\delta$).

To address this challenge, we conduct an ablation study to explore the robustness and sensitivity of ADVISE and other visual explanation methods. The ablation study consists of two parts: (1) we ablate the input image by randomly replacing pixels with the salt and pepper noise counterparts; (2) we remove ReLU at the same time to explore the effect of negative gradients on scoring the feature map units and the visual explanation. To do this, we use 3,000 images selected from ImageNet and ablate them using the noise density of $\delta = [0.025, 0.05, 0.075]$. Figure 5.6a shows an ablated image, and Figure 5.6b–5.6e shows the proposed method's performance compared with other visual explanation methods.

While the AVX value of the ADVISE and other visual explanation methods degrades due to incorporating negative gradients and ablating the input images, the proposed feature scoring method, unlike other methods, could meet the sensitivity axiom [19] in this classification task because the AVX never reached 0. However, we

should mention that the pitfall of the ablation test is that if we artificially ablate pixels in an image, we end up with inputs that do not belong to the original data distribution. The question of whether or not users should feed their models with inputs that are not part of the initial training distribution is still being debated [178, 179, 180].

## 5.3 Discussion and Future Works

In this chapter, we introduced a method for both quantifying and visualising feature relevance in CNNs using adaptive KDE. The proposed approach enhances the quality and interpretability of visual explanations by effectively highlighting salient regions that significantly contribute to a model's predictive decisions. Moreover, we propose comprehensive evaluation metrics to assess the quality and effectiveness of visual explanations from multiple dimensions, such as class sensitivity, hit rate, confidence drop, structural similarity, feature similarity, and mean squared error.

Despite its advantages, our method shares common limitations inherent to many existing visual explanation techniques. One challenge is that visual explanations are not sufficient for providing a complete understanding of the model's decision process. Visual explanations only show what regions or features in the input image are relevant for the model's prediction but do not show why or how these regions or features are relevant. For example, visual explanations do not explain what concepts or rules the model learned from the data, how these concepts or rules are combined or activated in different layers or units, or how these concepts or rules relate to human-understandable semantics or logic. Therefore, we need to complement visual explanations with other forms of explanations, such as textual, symbolic, or causal explanations, that can provide more insights into the model's reasoning process.

Another gap lies in the current visual explanation methods' ability to compare different models effectively. While these methods highlight relevant features, they do not capture unique attributes—exclusive features one model learns while another fails to. Comprehending such exclusive features is crucial for model comparison, enhancement, and knowledge transfer processes during distillation. Current visual explanation methods and evaluation metrics do not adequately capture the nuanced differences between models or measure the extent of unique attributes learned during knowledge distillation. To fill this gap, we proposed a novel technique called Uni-CAM, which includes quantifying metrics to elucidate model-specific features in the knowledge distillation process and assess their significance for the target task. The following chapter elaborates on our contribution to enhancing the explainability of architectures benefiting from the knowledge distillation training approach.

## 5.4 Conclusion

In this chapter, we presented a new method for quantifying and visualising the feature relevance of CNNs using adaptive kernel density estimation. The proposed method, ADVISE, estimates each unit's importance score in the feature map using adaptive KDE to capture the local variation of the gradient values. Then, it generates saliency maps highlighting how much each pixel in the input image influences the model's decision for a given class. We also presented new evaluation metrics to measure the quality and effectiveness of visual explanations. We conducted extensive experiments on three benchmark datasets and four popular CNN architectures to evaluate and compare the proposed method with existing methods. The results proved that ADVISE outperformed other methods and generated more salient and informative saliency maps with fewer units.

# Chapter 6

## Explaining Knowledge Distillation: Visualising and Quantifying Knowledge Transfer

Knowledge distillation (KD) is a technique used to enhance the performance of Deep Neural Networks (DNNs) by transferring knowledge from a complex Teacher to a simpler Student model [101, 102, 103, 104, 105, 106]. Despite its growing popularity in various applications such as computer vision, a major challenge in KD is to provide effective and interpretable explanations of the knowledge transfer process from a Teacher to a Student. Existing visual explainability techniques for Convolutional Neural Networks (CNNs) [84, 85, 86, 87, 90] are not directly applicable for the specific context of KD. These techniques only explain single-model predictions and fail to capture the specific features or activations of the knowledge transfer that enhance the Student's performance.

Our study introduces novel techniques to enhance the explainability of the KD process. We achieve this by comparing the attention patterns and salience maps of a Student model, trained with the guidance of the Teacher's knowledge and data, with a Base model, trained solely on data. The core of our analysis are *Distilled features*, novel features acquired by the Student, and *Residual features*, features that the Student overlooked and only exist in the Base model. As KD is generally thought to improve the Student's performance over the Base model, these distilled features could provide insights into what knowledge is transferred during KD, while residual features could be irrelevant to the task or a consequence of Student overfitting during KD. Hence, we aim to visualise and quantify these features, making the KD process more interpretable.

Specifically, we investigate the following key questions: (1) How similar are the attention patterns and saliency maps of the Student and the Base model? (2) How do the features learned by the Student compare to the features learned by the Base model in terms of their relevance to the task? (3) How can we quantify and visualise the saliency maps of the knowledge acquired during KD (distilled features) and the features the Student overlooked (residual features)? (4) How does the difference in depth architecture between the Teacher and Student influence the KD process, and can we explain the observed impact?

To address these key questions, we propose **Uni**que **C**lass **A**ctivation **M**aps (*Uni-CAM*), a novel gradient-based visual explanation method designed to visualise the saliency maps of the distilled and residual features. Furthermore, we propose two novel metrics, Feature Similarity Score (*FSS*) and Relevance Score (*RS*), using distance correlation (dCor) [181] and partial distance correlation(pdCor) [181, 182]. *FSS*

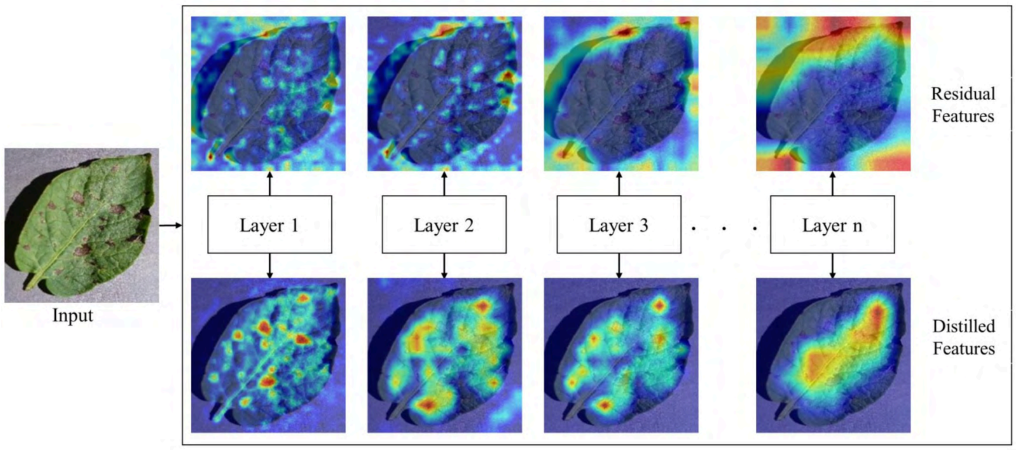



**Figure 6.1:** Saliency maps of distilled and residual features generated using *UniCAM*.

measures the similarity of the feature representations and the attention patterns between the Student and Base model. *RS* measures the relevance of the salient regions and the distilled features for the target task. To avoid potential architectural biases, we performed most experiments using the same architecture for the Student and Teacher models, using the Teacher as the Base model. We also performed an experiment where Student and Teacher architectures vary, to assess the fourth research question and evaluate the impact of architectural differences. We apply our methodology to three state-of-the-art KD techniques with different knowledge transfer strategies [183]: Response-based [107], Attention-based [105], and Overhaul feature-based [108].

Figure 6.1 shows how *UniCAM* visualises knowledge distilled and overlooked during KD. The distilled features focus on the leaf region, especially the areas crucial for plant disease classification. On the other hand, the residual features mainly highlight the background areas, which are irrelevant to the target task. The Student model learns to concentrate on the key aspects of the target object with the help of the knowledge transferred from the Teacher and thus achieves improved performance.

Our key contributions are fourfold: (1) we visualise and quantify the salient features localised by the Student and the Base model at various layers, demonstrating that KD facilitates the Student to learn more relevant features; (2) we introduce *UniCAM* that visualises the distilled and residual features during KD; (3) we propose novel metrics, *FSS* and *RS* to measure the similarity and relevance of the attention patterns, distilled and residual features; and (4) using *UniCAM*, *FSS* and *RS*, we demonstrate how smaller Student models struggle to learn relevant features from complex Teacher models, giving further insights to the findings by Mehdi et al. [184].

## 6.1 Methodology

Given a Teacher and a Student models, our goal is to explain and quantify distilled and residual features. Our approach uses Grad-CAM [90], distance correlation (dCor) [181], and partial distance correlation (pdCor) [181, 182] to introduce a novel visual explanation and metrics.

### 6.1.1 Preliminaries: distance and partial distance correlation

Distance correlation [181] measures the dependence between two random vectors that capture their multidimensional associations. For an observed random samples $(x, y) = (X_k, Y_k) : k = 1, \ldots, n$, where $n$ is the number of samples, the empirical distance correlation between $x$ and $y$, $R_n^2(x, y)$, is defined as:

$$R_n^2(x,y) = \begin{cases} \frac{V_n^2(x,y)}{\sqrt{V_n^2(x,x)V_n^2(y,y)}} & , V_n^2(x,x)V_n^2(y,y) > 0 \\ 0 & , V_n^2(x,x)V_n^2(y,y) = 0 \end{cases} \tag{6.1}$$

where $V_n^2(x, y)$, $V_n^2(x, x)$ and $V_n^2(y, y)$ are the squared sample distance covariance.

Partial distance correlation [181] extends dCor to measure the association between two random vectors after adjusting for the influence of a third vector. It is computed by projecting the distance matrices onto a Hilbert space and taking the inner product between the U-centered matrices. Let $(x, y, z)$ be random samples observed from the joint distribution of $(X, Y, Z)$, then the partial distance correlation between $x$ and $y$ controlling for $z$ is given by:

$$R^{*2}(x,y;z) = \begin{cases} \frac{(P_z^\perp(x)\cdot P_z^\perp(y))}{\|P_z^\perp(x)\|\cdot\|P_z^\perp(y)\|} & , \|P_z^\perp(x)\| \cdot \|P_z^\perp(y)\| \neq 0, \\ 0 & , Otherwise \end{cases} \tag{6.2}$$

where, $P_z^\perp(x)$, $P_z^\perp(y)$ is the orthogonal projection of $x$ and $y$ onto $z$, $(P_z^\perp(x)\cdot P_z^\perp(y))$ is the sample partial distance covariance and $\|.\|$ represents the Euclidean norm. Zhen et al. [182] used pdCor to condition multiple models and identify their unique features[1], which means removing the common features and assessing the remaining ones.

### 6.1.2 UniCAM: Unique Class Activation Mapping

We aim to generate the saliency maps of the distilled and residual features, highlighting their importance and attention patterns of the features, enabling a deeper understanding of KD. Grad-CAM [90] generates saliency maps based on the target class's gradients, revealing the relevance of features and providing a better understanding of the prediction. However, Grad-CAM is not suitable for KD, as it does

[1]Unique features are the features specific either to the Base model (residual features) or to the Student (distilled features).

not explain which features the Student acquires or misses from the Teacher. To address this limitation, we propose *UniCAM*, a novel gradient-based explainability technique for KD. *UniCAM* uses pdCor to adjust the feature representations for mutual influence and remove shared representations between the Student and Teacher. The remaining (unique) features are the distilled or residual features, which represent the knowledge that the Student model acquires or fails to learn from the Teacher.

Let $x_s$ and $x_t$ be the features extracted from a specific convolutional layer of the Student and the Teacher, respectively. *UniCAM* consists of the following main steps: (1) computing pairwise distance matrices for both $x_s$ and $x_t$, (2) normalising the distance matrices to obtain the adjusted distance matrices, $P^s$ and $P^t$, (3) measuring the mutual influence and subtracting the shared feature from each model's feature set, and (4) generating the heatmaps of the unique features. We will show in detail how to compute the distilled features, as the residual features can be computed in the same way.

Following the approach in [181], we first compute the pairwise distance matrix $D^{(s)} = (D^{(s)}_{i,j})$ as:

$$D^{(s)}_{i,j} = \sqrt{(x_i - x_j)^2 + \epsilon}. \tag{6.3}$$

Then, we normalise the distance matrix to obtain the adjusted pairwise distance matrix $P^{(s)}$. This normalisation is a form of U-centred projection that centres the distance matrix and accounts for the overall distribution of distances within each feature set.

$$P^{(s)}_{i,j} = \begin{cases} D^{(s)}_{i,j} - \dfrac{1}{n-2}\displaystyle\sum_{l=1}^{n} D^{(s)}_{i,l} - \dfrac{1}{n-2}\sum_{k=1}^{n} D^{(s)}_{k,j} \\ \qquad + \dfrac{1}{(n-1)(n-2)}\displaystyle\sum_{k=1}^{n}\sum_{l=1}^{n} D^{(s)}_{k,l}, & i \neq j; \\ 0, & i = j. \end{cases} \tag{6.4}$$

Next, we extract the unique features by adjusting for the mutual influence between the Student's and Teacher's features:

$$x_{s|unique} = P^{(s)} - \frac{\langle P^{(s)}, P^{(t)} \rangle}{\langle P^{(t)}, P^{(t)} \rangle} \cdot P^{(t)}. \tag{6.5}$$

where

$$\langle P^{(s)}, P^{(t)} \rangle = \frac{1}{n(n-3)} \sum_{i \neq j} \left( P^{(s)}_{i,j} \cdot P^{(t)}_{i,j} \right). \tag{6.6}$$

Then, we compute the importance of the unique features towards the model's prediction using the gradients with respect to these features and derive weights for each unit $k$ as:

$$\beta_k^{(x_{s|unique}, c)} = \frac{1}{N} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^{(x_{s|unique})}}, \tag{6.7}$$

where $A_{ij}$ is the activation at the $(i,j)^{th}$ location, $\beta_k^{(x_{s|unique},c)}$ is the weights of unit $k$ for class $c$, computed based on the unique features extracted, $x_{s|unique}$. $N$ is a normalisation factor to ensure the gradient's magnitude is standardised across different units. Finally, the *UniCAM* localisation maps for are generated by applying a ReLU function to emphasise the areas of importance:

$$L_{UniCAM}^{(x_{s|unique},c)} = \text{ReLU}\left(\sum_k \beta_k^{(x_{s|unique},c)} A^{(x_{s|unique})}\right). \tag{6.8}$$

### 6.1.3 Quantitative analysis of KD features

In addition to *UniCAM*, we propose two novel metrics —Feature Similarity Score (*FSS*) and Relevance Score (*RS*)—to quantify the distilled and residual features. To compute these metrics, we need to extract the features from the salient regions generated using *UniCAM* or Grad-CAM. Then, we use a perturbation technique proposed by Rong et al. [185], that modifies the image pixels based on their prediction relevance. This perturbation technique preserves the important pixels and replaces the rest with the weighted average of their neighbours. This way, the perturbed images retain the most salient features identified by *UniCAM* or Grad-CAM, while reducing the noise and redundancy of the irrelevant features. Fig. 6.2 shows this process with examples of input images, *UniCAM* explanations, and perturbed images.

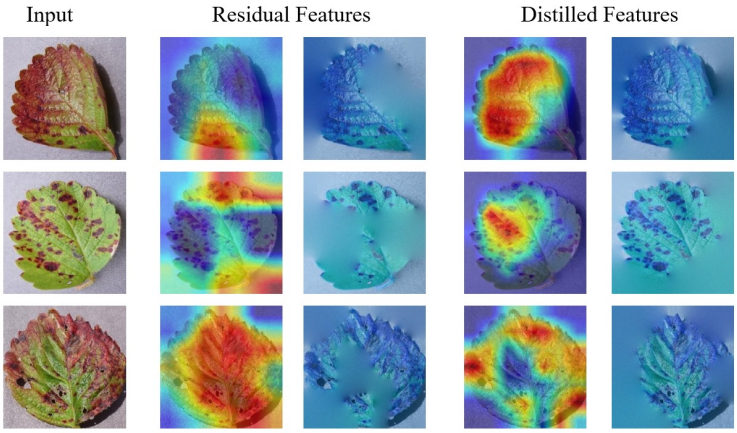



**Figure 6.2:** Visualisation of residual and distilled features after perturbation.

The feature extraction function takes the perturbed images in Fig. 6.2 as input and extract the features (See Eq. 6.9) from the corresponding layer as:

$$\hat{x}_s = f_s(I \odot \mathcal{H}),\ \hat{x}_t = f_t(I \odot \mathcal{H}) \tag{6.9}$$

where $f_s$ and $f_t$ are the feature extractor functions for the Student and Teacher, respectively, $I$ is the input image, $\mathcal{H}$ is the heatmap generated by *UniCAM* or Grad-CAM, $\odot$ is the element-wise multiplication operator. These features are the numerical

representations of the perturbed image that capture the essential information for prediction, and their dimension depends on the number of filters and the size of the activation maps in each layer. Hence, using these features, we quantify the similarity of the attention patterns and relevance of distilled and residual features.

**Feature similarity score (FSS):**

*FSS* measures how much the Student and the Teacher model agree on the most relevant features. Since distilled and residual features capture the unique features of each model, comparing their similarity using *FSS* is not logical. For this reason, we compare the similarity of the attention patterns of the Student and Teacher saliency maps generated using Grad-CAM and measure their degree of alignment as follows:

$$FSS = R^2(\hat{x}_s, \hat{x}_t) = \frac{1}{k}\sum_{i=1}^{k} \mathrm{dCor}(\hat{x}_{s_i}, \hat{x}_{t_i}), \tag{6.10}$$

where $k$ is the number of batches, $\hat{x}_{s_i}$ and $\hat{x}_{t_i}$ are the minibatch features of the Student and Teacher extracted from the highlighted region. *FSS* ranges from 0 to 1, where 0 means no similarity and values close to 1 indicates higher the attention pattern similarity.

**Relevance score (RS):**

*RS* quantifies the relevance of the distilled and residual features for the ground truth ($gt$) prediction. Following Zhen et al. [182], we used a pre-trained *BERT* [186] embedding of the label as ground truth. This allows the label to capture more semantic information than a one-hot encoding and allows for meaningful notions of distance between different ground truth $gt$ embedding. Hence we formulate *RS* as follows:

$$RS = R^2(\hat{x}_s, gt) = \frac{1}{k}\sum_{i=1}^{k} \mathrm{dCor}(\hat{x}_{s_i}, gt_i), \tag{6.11}$$

where $\hat{x}_{s_i}$ is the features extracted using Student and $gt_i$ is the ground truth of the corresponding batches. We replace $\hat{x}_{s_i}$ with $\hat{x}_{t_i}$ to compute the *RS* for the Teacher. Therefore, both *FSS* and *RS* offer a robust quantitative approach to evaluate the similarity of attention patterns and relevance of distilled features during KD.

## 6.2 Experiments

### 6.2.1 Datasets and Implementation Details

We evaluate the proposed method on three public datasets for image classification: ASIRRA (Microsoft PetImages) [187], CIFAR10 [188] and Plant disease classification dataset [189]. ASIRRA contains 25,000 images of cats and dogs, while CIFAR10 contains 60,000 images of 10 classes. These datasets are widely used as benchmarks for image classification tasks and have different levels of complexity and diversity. Plant disease classification dataset poses a more challenging and realistic problem of fine-grained image classification, where the differences between classes are subtle and require more attention details.

We performed various experiments to analyse and explain the KD process. First, we used similar architecture for the Student and Teacher, ResNet-50 [167], to avoid the bias of a more complex Teacher having more complex or relevant features than a simpler Student model. We analysed the performance of the Teacher and Student models, the similarity attention patterns and relevance of the distilled and residual features durign KD. In the second experiment, we analysed different combinations of ResNet-18, ResNet-50, and ResNet-101 as Teacher and Student models to explain the opaque KD process. We used the proposed methods to uncover the challenges of KD when the Teacher is complex and the Student is small. We applied our approach to three state-of-the-art KD methods for classification: response-based KD [107], overhaul feature-based KD [108], and attention-based KD [105]. We implemented the proposed method using PyTorch [190] and using open source codes from KD [191] and Grad-CAM [192].

### 6.2.2 Results

We trained the models using 5-fold cross-validation. We assessed the performance and visual explanations of Teacher and Student models trained with different KD. As shown in Fig. 6.3, the KD-trained models achieved higher accuracy than the corresponding Teacher model (Base model).

#### Comparison of Teacher-Student attention patterns:

We hypothesise that KD enhances the Student model's ability to learn more salient features and ignore irrelevant ones. To test this, we visualise and quantify the agreement and difference between the visual explanations of the Student and its corresponding Base model at various layers.

We compare the features learned by the Teacher and Student at the salient regions and measured their attention pattern similarity and relevance. Fig. 6.4 shows the

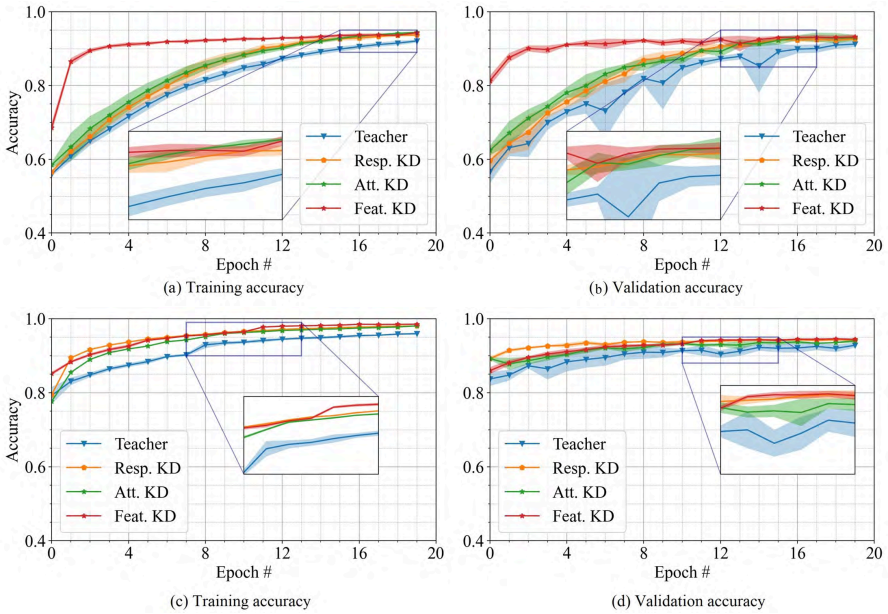



**Figure 6.3:** The training and validation accuracy( (a) and (b) ASIRRA, (c) and (d) CIFAR10. The shaded region is the standard deviation.

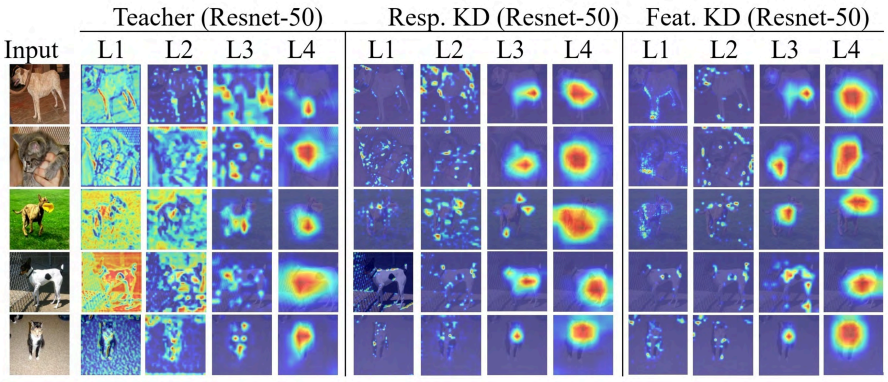



**Figure 6.4:** Grad-CAM visualisation of Teacher and Student models trained with various KD techniques at different layers.

Grad-CAM visualisation at L1, L2, L3, and L4 of the last residual blocks in the four layers of the ResNet-50. The Teacher model relies on low-level features such as edges and spreads the attention over the entire image, including the background, in the first and intermediate layers. The saliency maps generated by KD trained models, however, highlight more salient regions and focus on the object in all layers. This suggests that KD helps a model learn better features and improve its localisation ability by directing attention to more salient features earlier in the network.

We then use *FSS* and *RS* metrics to quantify the attention pattern similarity and relevance of the features between the Teacher and Student models. We apply these metrics to the features extracted from the localised regions at various layers of the models, which we obtain by using *UniCAM* and Grad-CAM. For each layer, we

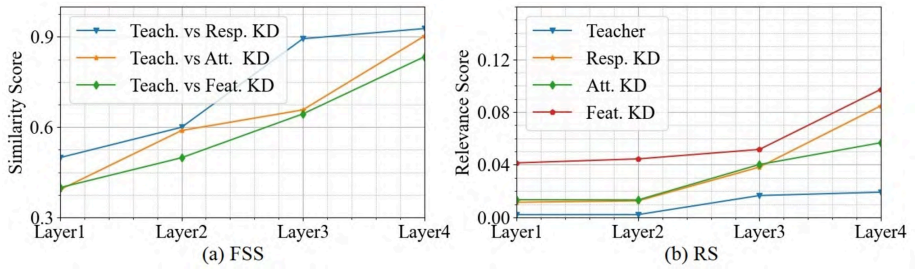



**Figure 6.5:** (a) the attention pattern similarity and (b) relevance of the salient regions between Student and Teacher, localised by Grad-CAM.

compute $FSS = R^2(\hat{x}_t, \hat{x}_s)$, where $\hat{x}_t$ and $\hat{x}_s$ are the relevant features extracted from the Teacher and Student models, respectively. We also compute $RS_t = R^2(\hat{x}_t, gt)$ and $RS_s = R^2(\hat{x}_s, gt)$ for the Teacher and Student models respectively, where $gt$ is the *BERT* embedding representing the ground truth of the predicted class.

Fig. 6.5(a) and (b) show the feature similarity of the attention patterns (*FSS*) and their relevance score (*RS*) between the Teacher and Student models across different layers, respectively. The *FSS* is higher for the deeper layers than for the input and intermediate layers, indicating that the Student models either learn more salient features in the input layers or fail to mimic the Teacher and learn more irrelevant features. However, the Grad-CAMs in Fig. 6.4 shows that the Student models have localised far better salient features than the Teacher model, especially in the input and intermediate layers. Therefore, the lower *FSS* at input and intermediate layers suggests that the Students have learned more relevant features that the Teacher model have not learned yet. Moreover, the Student models achieve higher *RS* than the Teacher across all layers, implying that the models trained with KD have learned more relevant features with the guidance of the Teacher knowledge.

In summary, Fig. 6.4 and Fig. 6.5 demonstrate that KD enables the Student models to acquire more relevant features, which enhance the prediction accuracy and ability to generalise. Furthermore, our proposed methods can explain when a student overfits and fails to distil adequate knowledge from the teacher (see exploring the capacity gap).

**Visualising and quantifying knowledge transfer:**

Here we use our proposed *UniCAM* method to visualise the unique features the student model acquired during KD. The saliency maps generated using *UniCAM* show that KD is not a simple feature copying process from the Teacher to the Student but a guided training process where the Teacher's knowledge assists the Student to learn existing or new features. This is demonstrated in Fig. 6.6 where distilled features are mainly concentrating on the primary object, while the residual features are concentrated on the background or seemingly less relevant parts of the object. Moreover, in the plant disease classification, distilled features accurately identify segments of

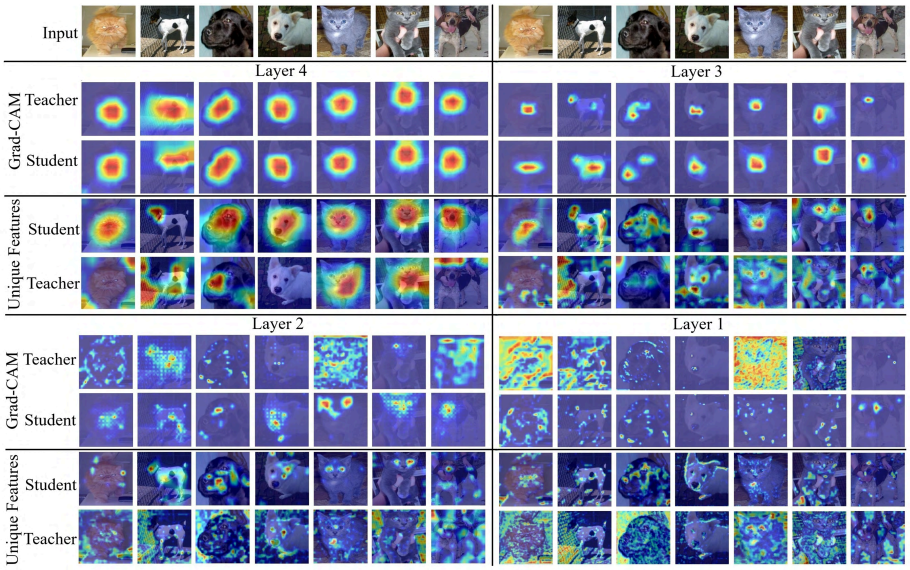



**Figure 6.6:** Sample visualisation of Teacher and Student features. Distilled features focus on the object, while residual features spread over the entire image.

**Table 6.1:** Relevance of features learned by Teacher model (ResNet-50) and Student models (ResNet-50) trained with different different distillation techniques.

| Dataset | KD-Technique | Layer# | Feature Relevance | | | |
|---|---|---|---|---|---|---|
| | | | Grad-CAM | | UniCAM | |
| | | | Salient Teacher fts. | Salient Student fts. | Residual fts. | Distilled fts. |
| ASIRRA [187] | Response-based | L1 | 0.0092 | **0.0189** | **0.0024** | 0.0017 |
| | | L2 | 0.0054 | **0.0130** | 0.0001 | **0.0040** |
| | | L3 | 0.0100 | **0.0365** | 0.0007 | **0.008** |
| | | L4 | 0.0141 | **0.0861** | 0.0043 | **0.006** |
| | Attention-based | L1 | 0.0092 | **0.0107** | **0.0049** | 0.0047 |
| | | L2 | 0.0054 | **0.0189** | 0.0022 | **0.0035** |
| | | L3 | 0.0100 | **0.0431** | 0.0045 | **0.0100** |
| | | L4 | 0.0141 | **0.0583** | 0.0082 | **0.0102** |
| | Feature-based | L1 | 0.0092 | **0.0465** | 0.0063 | **0.0101** |
| | | L2 | 0.0054 | **0.0453** | 0.0027 | **0.0048** |
| | | L3 | 0.0100 | **0.0570** | 0.0036 | **0.0196** |
| | | L4 | 0.0141 | **0.0953** | 0.0012 | **0.0258** |
| CIFAR10 [188] | Response-based | L1 | 0.0063 | **0.0304** | 0.0040 | **0.0155** |
| | | L2 | 0.0133 | **0.0378** | 0.0090 | **0.0148** |
| | | L3 | 0.0282 | **0.0432** | 0.0046 | **0.0113** |
| | | L4 | 0.0417 | **0.0585** | 0.0090 | **0.0106** |
| | Attention-based | L1 | 0.0063 | **0.0232** | 0.0043 | **0.0136** |
| | | L2 | 0.0133 | **0.0280** | 0.0099 | **0.0101** |
| | | L3 | **0.0282** | 0.0256 | **0.0087** | 0.0063 |
| | | L4 | 0.0417 | **0.0437** | 0.0017 | **0.0021** |
| | Feature-based | L1 | 0.0063 | **0.0311** | 0.0028 | **0.0185** |
| | | L2 | 0.0133 | **0.0388** | 0.0017 | **0.0153** |
| | | L3 | 0.0282 | **0.0457** | 0.0070 | **0.0117** |
| | | L4 | 0.0417 | **0.0794** | 0.0024 | **0.0150** |

leaves essential for disease classification, demonstrating that KD helps models learn more relevant features.

In addition, Table 6.1 quantifies the relevance of the features extracted from regions highlighted using Grad-CAM and *UniCAM*. *UniCAM* visually explains the class

discriminative features exclusive to the Teacher and Student models, whereas Grad-CAM generates saliency maps highlighting the features important for the prediction. The models trained with various KD techniques have higher *RS* than the equivalent Base model. Among the Students, overhaul feature distillation achieved better accuracy performance and learned more relevant features. Overhaul distillation transfers the intermediate feature representations of the Teacher, enabling the Student to learn more fine-grained and diverse features.

To further evaluate the unreliability of our proposed explainability technique to complex datasets, we applied it to plant disease classification. The distilled features in Fig. 6.7 primarily highlighted regions crucial for accurate prediction, while residual features tended to be distributed across areas irrelevant to the disease diagnosis. Furthermore, we visualised the disease-affected areas for Potato Early Blight and

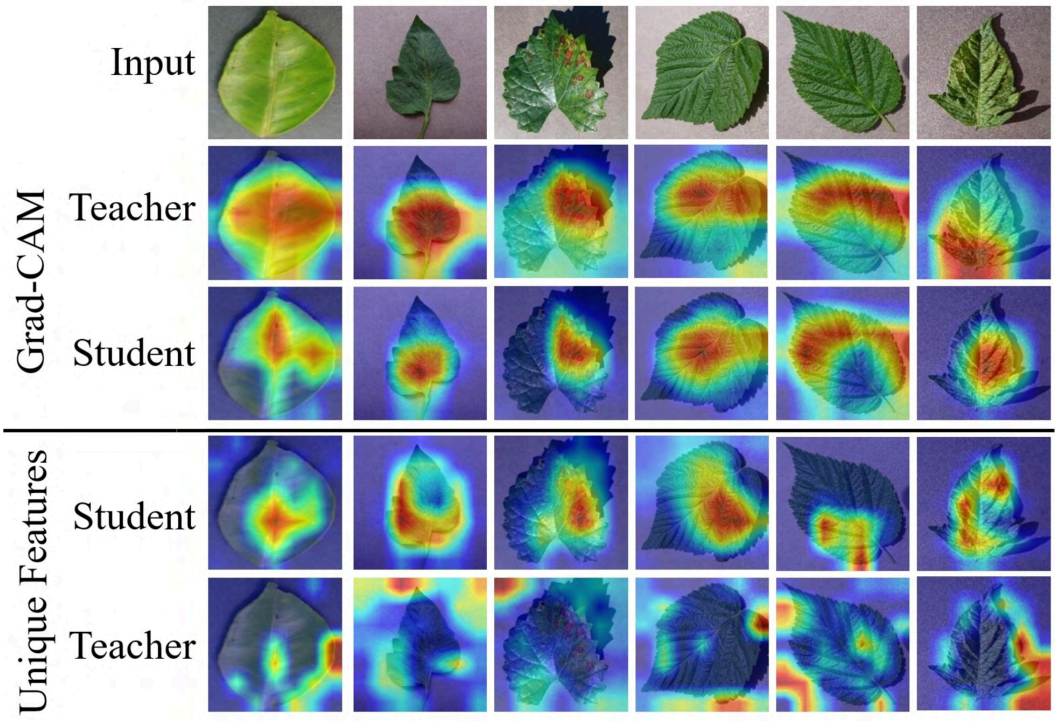
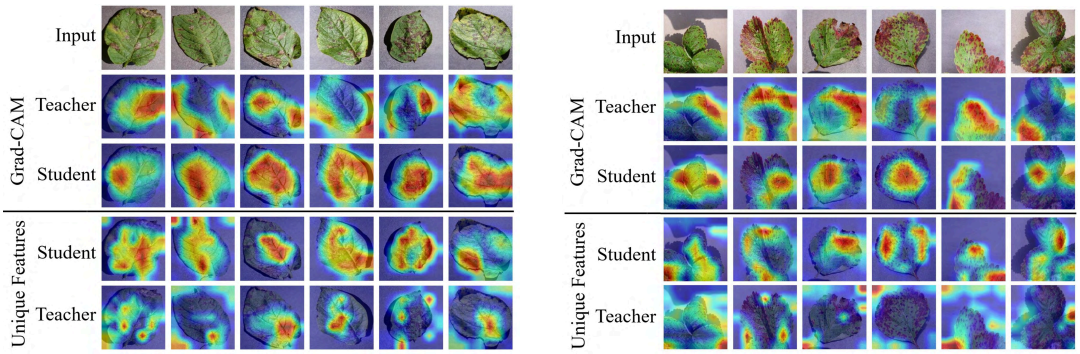


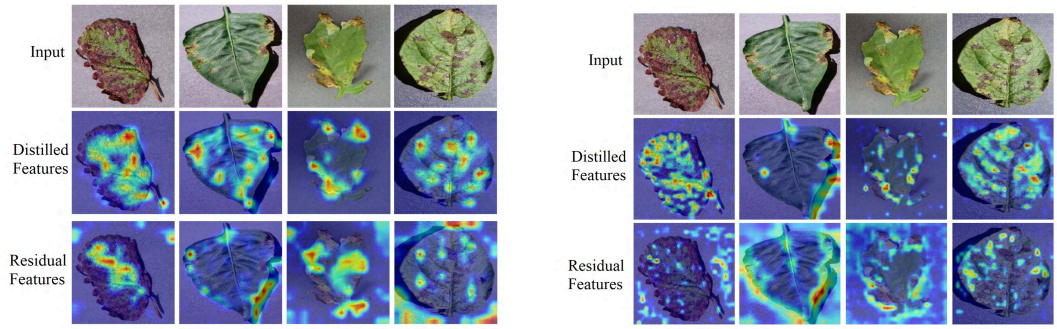

**Figure 6.7:** Sample visualisation of unique (distilled and residual) features in Plant disease classification.

Strawberry Leaf Scorch (Fig. 6.8). The KD-trained model shows improved localisation of crucial disease signs on the leaves. Finally, we explored the distilled and residual features from Layers 3 and 2 (Fig. 6.9). Distilled features predominantly localised the diseased regions in the input image, even for challenging cases. On the other hand, residual features highlighted areas with small or no influence on plant disease classification. Therefore the proposed visual explanation method explains the distilled and residual features, and it also demonstrates that knowledge guides the Student model to learn relevant features for the prediction.




**(a)** Potato Early Blight **(b)** Strawberry Leaf Scorch

**Figure 6.8:** Sample visualisation of distilled and residual features on Potato Early Blight and Strawberry Leaf Scorch plant disease classification.




**(a)** Layer-3 **(b)** Layer-2

**Figure 6.9:** Sample visualisation of Distilled and residual features on Plant disease classification from Layer-3 and Layer-2.

### Exploring the capacity gap impact:

The Student's performance often declines when there is a large architecture (capacity) gap between the Teacher and the Student [193, 194]. However, the Student's performance drop could be either attributed to its own inability to learn relevant features, or the Teacher's knowledge is overwhelming the student. To address this issue, our experiment adopts two distillation strategies involving ResNet-101 as a Teacher and ResNet-18 as a Student, which have a large capacity gap. In the first approach, we perform direct knowledge transfer from ResNet-101 to ResNet-18, while the second introduces an intermediate "Teacher assistant" [195], to bridge the capacity gap between ResNet-101 and ResNet-18. We use *UniCam* and Relevance Score (*RS*) to investigate the KD process in these settings, with a focus on how well the smaller model manages to learn relevant features. Our analysis suggests the large capacity gap may contribute to the distilled model's performance drop.

We first examine the impact of a large capacity gap on the knowledge transfer between Teacher and Student. We use ResNet-101 as the Teacher and ResNet-18 as the Student and apply KD to train the Student model. Fig. 6.10 shows the saliency

maps of the unique features of both models. We notice that the distilled knowledge is unrelated to the object and misses the relevant part of the input. To explore the cause of this performance drop, we train a ResNet-18 (Base Model) without KD and compare the unique features with the Student model. Fig. 6.11 demonstrates that in this setting the Base Model captures more relevant features than the Student model. This suggests that a large capacity gap impedes knowledge transfer, as the Student model cannot effectively learn from the complex Teacher's knowledge.

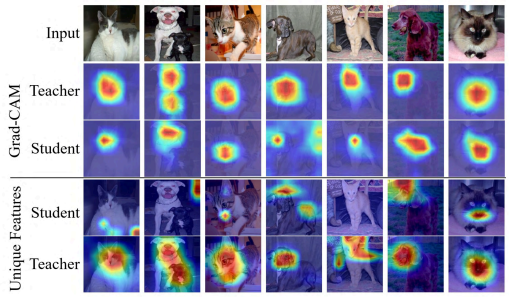



**Figure 6.10:** Relevant features learned by ResNet-18 (Student) distilled from ResNet-101 (Teacher).


**Figure 6.11:** Relevant features learned by Student (ResNet-18) distilled from ResNet-101 compared to Base Model.

To bridge the capacity gap, we use an intermediate Teacher assistant to enable a more effective and focused knowledge transfer from ResNet-101 to ResNet-18 via ResNet-50. Fig. 6.12 compares the saliency maps of the distilled features learned by two Students: ResNet-18 directly distilled from ResNet-101 (R18-R101) and ResNet-18 distilled from ResNet-101 through Teacher assistant ResNet-50 (R18-R50-R101). The saliency maps, visualised using *UniCAM*, reveal that the Teacher assistant helps learn more relevant features that highlight the object parts, while R18-R101 learns some irrelevant features and misses the salient features for the *gt* prediction. The Teacher assistant facilitates the Student model to learn compatible knowledge from the complex Teacher and provides more appropriate supervision and feedback.

We conducted a feature ablation analysis to assess the quality of the features learned by the Student trained with Teacher-assistant and its equivalent Base model. We used *UniCAM* to generate the saliency maps of the distilled and residual features of each model. Fig. 6.13 shows that the saliency maps of the distilled features are more focused on the salient regions of the input images, while the residual features are more dispersed.

Finally, Table 6.2 quantifies the relevance of the features learned by the Base model, and equivalent Student model at different layers. The model trained with the Teacher assistant has learned more relevant features compared to the model directly distilled from ResNet-101 and the Base model.

The empirical findings presented above indicate that the capacity gap between the Teacher and Student models influences the quality and efficiency of KD. We demonstrate the benefit of our methods in the analysis of the Student model's behaviour,

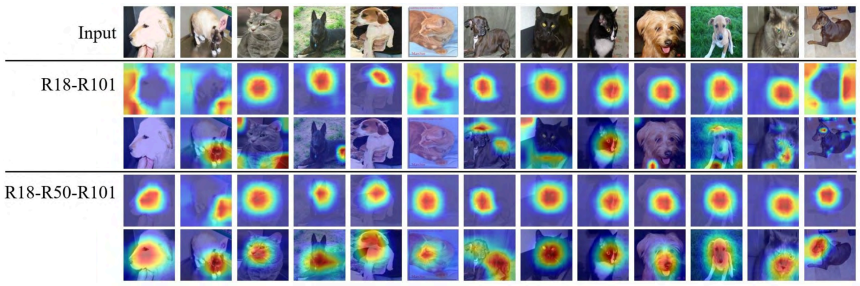



**Figure 6.12:** Grad-CAM ($2^{nd}$ and $4^{th}$ rows) and *UniCAM* ($3^{rd}$ and $5^{th}$ rows) visualisations.

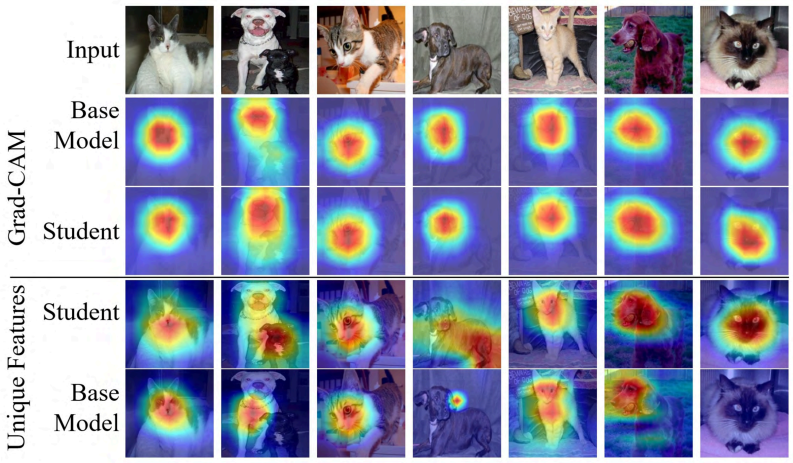



**Figure 6.13:** Comparison of distilled and residual features between Student (R18-R50-R101) and Base Model.

**Table 6.2:** Comparing visual explanations and distilled and residual features of ResNet-18 trained with and without Teacher assistant on the ASIRRA dataset.

| Layer# | Main features | | | Distilled/Residual features | | |
|---|---|---|---|---|---|---|
| | Base model | R18-R101 | R18-R50-R101 | Base model | R18-R101 | R18-R50-R101 |
| L1 | 0.0037 | 0.0022 | **0.0052** | 0.0014 | 0.0007 | **0.0050** |
| L2 | 0.0039 | 0.0035 | **0.0050** | 0.0016 | 0.0014 | **0.0031** |
| L3 | 0.0057 | 0.0045 | **0.0074** | 0.0012 | 0.0008 | **0.0060** |
| L4 | 0.0063 | 0.0052 | **0.0082** | 0.0018 | 0.0011 | **0.0076** |

both when it succeeds and when it fails to learn relevant features from the Teacher. Therefore, our visual explanation technique and metrics can aid to select the optimal Teacher-Student pairs and improving KD training.

## 6.3 Discussion and Future Works

This chapter presented novel techniques to explain and quantify the knowledge transfer during KD. We proposed *UniCAM*, a gradient-based visual explanation method to explain the distilled knowledge and residual features during KD. Our experimental results show that *UniCAM* provides a clear and comprehensive visualisation of the

features acquired or missed by the Student during KD. We also proposed two metrics: *FSS* and *RS* to quantify the similarity of the attention patterns and the relevance of the distilled knowledge and residual features. We demonstrate that KD helps the Student learn more useful features, and the saliency maps of the distilled features mainly focus on the target object. The proposed explainability technique can be applied to other applications beyond explaining KD, such as choosing the best model for fine-tuning or transfer learning for a given dataset. A possible extension of this work is to select the best model for fine-tuning based on the relevance of its features for the target dataset.

The proposed method has some limitations. We only conducted experiments on the classification task, which is one of the many possible tasks that benefit from KD. Thus, it would be interesting to explore the behaviour of KD for other tasks to investigate the robustness of the proposed method. This is part of our future work.

# Chapter 7

## Discussion

This chapter provides a comprehensive summary of the research journey undertaken in this thesis. It reflects on the achievements in selecting informative samples to reduce human effort and improve generalisation through Meta-learning, designing reliable and robust AI via Uncertainty-Guided Learning, and developing novel Explainability techniques to elucidate CNN decision-making. This summary shows how the research chapters relate and form a coherent story. The implications of these studies are outlined, and an outlook for future work is provided, highlighting the potential areas of research that could advance the field of unbiased, transparent, and ethical AI.

## 7.1 Summary

This thesis has conducted a systematic exploration of the dynamic field of computer vision. It represents several years of meticulous research, combining innovative thinking, rigorous analysis, and practical applicability.

**Chapter 1** established the foundation by identifying the needs and challenges within the field of computer vision. It introduced the research questions investigated in the following chapters and provided a theoretical background, focusing on fundamental concepts and theories that are essential to computer vision, thereby creating a solid basis for more specialised investigations.

**Chapter 2** presented a thorough review of related works and relevant literature, situating our research within the existing academic landscape.

**Chapter 3** explored Meta-learning: A Reinforcement Learning for Informative Sample Selection in Image Classification. This novel approach aimed to optimise the selection of samples, improving the efficiency and effectiveness of image classification models.

**Chapter 4** explored Uncertainty-Guided Learning: Active Learning with MC Dropout, a methodology that uses uncertainty estimates to select informative samples for annotation and enable a classifier with rejection. This chapter proposed introducing human-in-the-loop approaches that leverage uncertainty to enhance learning efficiency and reliability, both at training and inference time.

**Chapter 5** presented Adaptive KDE for Visual Explainability of CNNs: A Novel Approach to quantify and visualise feature relevance. This chapter developed a novel

way to understand and articulate the workings of Convolutional Neural Networks, contributing to the ongoing research about transparency and interpretability within machine learning.

**Chapter 6** focused on Explaining Knowledge Distillation: Visualising and Quantifying Knowledge Transfer. This exploration into the nature of knowledge transfer within model training not only provided valuable insights but also proposed practical tools for visualising and understanding this complex process.

The contributions in this thesis highlight the relentless pursuit of innovations that advance the field of computer vision towards more responsible, transparent, and unbiased applications. Through Meta-learning, Uncertainty-Guided Learning, Adaptive KDE, and Knowledge Distillation techniques, we have worked to select informative samples, create uncertainty-aware CNNs, propose novel Explainability methods, and explain KD techniques. These interconnected themes represent a mindful progression towards a more ethical and comprehensible implementation of machine learning. This thesis enriches the academic discourse and sets a benchmark for future exploration within the domain of computer vision by bridging the gap between cutting-edge research and some practical applications. The collaborative, accurate, and transformative approach showcased in this work stands as a testament to the possibility and promise of responsible and explainable AI.

## 7.2 Reflection and Outlook

Finally, we hope that the proposed methods and techniques for improving the performance, reliability, and explainability of DNNs for various computer vision tasks will inspire and motivate future research in this field. We believe that DNNs have great potential and value for solving complex and challenging problems in computer vision and beyond, but they also need to be enhanced and explained to ensure their quality and trustworthiness. We hope our work will contribute to advancing the state-of-the-art in DNNs and making them more accessible, explainable, responsible, and beneficial for humans.

This three-year-long PhD research journey started with a focus on selecting informative samples to avoid model overfitting and bias, essentially handling bias at the data level. This approach laid the foundation for understanding the causes and effects of overfitting and the possible solutions. The research then moved towards uncertainty, developing a methodology that allows a DNN model to abstain from decision-making when uncertain, enhancing the model's reliability and robustness. However, these approaches did not fully reveal the internal mechanisms of DNNs' decision-making. Therefore, the research further investigated the techniques explaining internal decision-making and proposed a novel visual explainability technique. Finally, the exploration culminated in proposing new visual explainability techniques to explain the decision-making processes within knowledge distillation, creating a

comprehensive and coherent progression from data-level explanations to deep insights into model functionality.

The future of explainability in DNNs is fascinating and multifaceted. Uncovering causality in explanations and balancing the complexity and understandability of deep models remain central challenges, offering exciting opportunities for future exploration. The quest for innovative methodologies to translate DNN decisions into human-comprehensible terms continues, with the aspiration to bridge the current gap between academic exploration and real-world implementation. The commitment to responsible and ethical AI practices guides this ongoing journey, shaping the landscape for future research and applications in DNNs.

## 7.3 Future Works

We have proposed several novel methods and techniques for improving the performance, reliability, and explainability of deep neural networks (DNNs) for various computer vision tasks. We have also discussed the proposed methods' implications, limitations, and future directions. Here, we will summarise the main feature works and suggest possible ways to extend and improve our research.

1. For the meta-learning method, we suggested improving it by using expert opinion and uncertainty for sample selection, handling multiple tasks and datasets, using more efficient reinforcement learning algorithms, integrating with an active learning framework, and testing on real-world datasets and applications.
2. For the uncertainty-guided learning method, we suggested improving it by using Bayesian optimisation to find the optimal dropout rate, using more complex distributions for the approximate posterior, and using textual, symbolic, or causal explanations to provide more insights into the model's reasoning process.
3. For the adaptive KDE method, we suggested exploring and improving its application across domains such as segmentation, detection and DNN models such as LSTM.
4. On explaining knowledge distillation method, we suggested improving it by applying it to various computer vision tasks and other applications that benefit from using KD.

## 7.4 Ethical Implications

The advent of Artificial Intelligence (AI) has brought about transformative changes across multiple sectors. However, alongside its myriad benefits, AI has also raised significant ethical concerns, particularly concerning data bias, decision transparency, and overall accountability. Pursuing responsible AI necessitates a comprehensive

approach that extends beyond technical proficiency to consider the broader social, ethical, and legal implications of these technologies.

Firstly, the issue of data bias is crucial. While this research has contributed methods for mitigating such bias, the algorithms and models can only be as equitable as the data on which they are trained. Even with balanced and unbiased training data, the potential for unintended discriminatory outcomes remains, thereby mandating continuous ethical vigilance.

Secondly, another ethical dimension introduced in this thesis is the aspect of decision uncertainty. Models often operate under conditions where data can be ambiguous or incomplete. The thesis aims for a more responsible application of AI by incorporating uncertainty into the decision-making process and deferring to expert opinions when necessary. This not only enhances the model's performance but also reduces the risk of making uninformed or potentially harmful decisions, especially in critical sectors like healthcare or public safety.

Thirdly, decision transparency is an ethical imperative as well as a technical requirement. Algorithms used in critical applications such as healthcare, criminal justice, and financial services wield substantial societal influence. Hence, the ability for these algorithms to 'explain themselves' via visual explanations is essential for ethical governance and democratic accountability.

In conclusion, the ethical considerations surrounding AI are complex and multifaceted. While this research aims to contribute to more responsible and transparent AI, it is crucial to acknowledge that the ethical discourse surrounding these technologies is an ongoing research requiring continual scrutiny. This thesis serves as one step towards addressing these ethical challenges but is by no means a conclusive solution.

# Chapter 8

## Contributions

This chapter summarises our research contributions and details the significant publications resulting from our work. These contributions reflect key developments and novel approaches in our field of study. Through a methodical process and rigorous investigation, our research has led to a series of publications, each contributing to the broader scientific community. This section highlights the specific achievements and places them within the wider context of academic discourse, emphasising the new perspectives the research offers.

## 8.1 First Paper

### 8.1.1 Bibliography Entry

Adhane, G., Dehshibi, M. M., & Masip, D. (2021). A deep convolutional neural network for classification of aedes albopictus mosquitoes. *IEEE Access*, 9, 72681-72690. ISI JCR IMPACT FACTOR: 3.476 (2021). **Q1**.

### 8.1.2 Abstract

Monitoring the spread of disease-carrying mosquitoes is a first and necessary step to control severe diseases such as dengue, chikungunya, Zika or yellow fever. Previous citizen science projects have been able to obtain large image datasets with linked geo-tracking information. As the number of international collaborators grows, the manual annotation by expert entomologists of the large amount of data gathered by these users becomes too time demanding and unscalable, posing a strong need for automated classification of mosquito species from images. We introduce the application of two Deep Convolutional Neural Networks in a comparative study to automate this classification task. We use the transfer learning principle to train two state-of-the-art architectures on the data provided by the Mosquito Alert project, obtaining testing accuracy of 94%. In addition, we applied explainable models based on the Grad-CAM algorithm to visualise the most discriminant regions of the classified images, which coincide with the white band stripes located at the legs, abdomen, and thorax of mosquitoes of the *Aedes albopictus* species. The model allows us to further analyse the classification errors. Visual Grad-CAM models show that they are linked to poor acquisition conditions and strong image occlusions.

### 8.1.3 Background

In our study, we have explored the application of a deep convolutional neural network for the classification of Aedes albopictus mosquitoes (Adhane, G., Dehshibi, M. M., & Masip, D., 2021). Published in IEEE Access, Volume 9, this research represents a relevant contribution to both deep learning and public health, by employing cutting-edge techniques to categorise a mosquito species of significant concern.

### 8.1.4 New Methods

This work aims to develop a deep convolutional neural network (CNN) for the classification of Aedes albopictus mosquitoes, also known as the Asian tiger mosquito or forest mosquito. Aedes albopictus is a vector of many viral diseases, such as dengue, chikungunya, Zika, and yellow fever, and poses a serious threat to public health worldwide. We propose a CNN model that can automatically identify and classify Aedes albopictus mosquitoes from images captured by a smartphone camera. The model is trained and tested on a large dataset of mosquito images collected from different regions and environments. The results show that the proposed CNN model achieves high accuracy and robustness in classifying Aedes albopictus mosquitoes, and outperforms existing methods. This work can provide a useful tool for mosquito surveillance and control, especially in resource-limited settings.

### 8.1.5 Results

In our research, we applied a deep convolutional neural network to classify Aedes albopictus mosquitoes, achieving significant success in accurately identifying the species. Through careful experimentation and detailed analysis, we fine-tuned our model to optimise performance, demonstrating its effectiveness in various real-world scenarios. A critical part of our study involved using Grad-CAM to characterise the Aedes albopictus mosquito parts, allowing us to understand how the CNN model makes its decisions and if it uses the same characteristics as entomologists. With the help of Grad-CAM, we discovered that, in addition to the parts of the mosquito that entomologists use to distinguish, the model has been utilising additional body parts such as the strips in the abdomen, the hairs in the antennae, and the thorax shape. This significant finding offers new insights that can help entomologists easily distinguish mosquito species from one another, enriching the existing mosquito species identification.

## 8.2 Second Paper

### 8.2.1 Bibliography Entry

Adhane, G., Dehshibi, M. M., & Masip, D. (2021). On the use of uncertainty in classifying Aedes Albopictus mosquitoes. *IEEE Journal of Selected Topics in Signal Processing*, 16(2), 224-233. **Q1**

### 8.2.2 Abstract

The re-emergence of mosquito-borne diseases (MBDs), which kill hundreds of thousands of people each year, has been attributed to increased human population, migration, and environmental changes. Convolutional neural networks (CNNs) have been used by several studies to recognise mosquitoes in images provided by projects such as Mosquito Alert to assist entomologists in identifying, monitoring, and managing MBD. Nonetheless, utilising CNNs to automatically label input samples could involve incorrect predictions, which may mislead future epidemiological studies. Furthermore, CNNs require large numbers of manually annotated data. In order to address the mentioned issues, this paper proposes using the Monte Carlo Dropout method to estimate the uncertainty scores in order to rank the classified samples to reduce the need for human supervision in recognising Aedes albopictus mosquitoes. The estimated uncertainty was also used in an active learning framework, where just a portion of the data from large training sets was manually labelled. The experimental results show that the proposed classification method with rejection outperforms the competing methods by improving overall performance and reducing entomologist annotation workload. We also provide explainable visualisations of the different regions that contribute to a set of samples' uncertainty assessment.

### 8.2.3 Background

In this paper, we further explore the classification of Aedes albopictus mosquitoes, focusing on the incorporation of uncertainty models within deep learning frameworks. Our research represents a novel approach in the field of signal processing, offering insights into how uncertainty can be leveraged to improve classification accuracy. The work, published in the IEEE Journal of Selected Topics in Signal Processing, Volume 16, Issue 2, contributes to the ongoing efforts to enhance vector control and disease prevention.

### 8.2.4 New Methods

Building upon previous research, this work introduces the concept of uncertainty modelling to the classification of Aedes albopictus mosquitoes. We propose and compare

various uncertainty models integrated with deep learning architectures to assess their impact on classification performance. The experimental design includes the use of large and diverse datasets encompassing various environmental conditions. The innovative methodologies employed in this study provide a foundation for future research and potential applications in real-world scenarios.

### 8.2.5 Results

In our exploration of the application of uncertainty in the classification of Aedes albopictus mosquitoes, we achieved substantial findings that open new avenues in vector control and public health. Through comprehensive experimentation, we not only identified the optimal uncertainty model to enhance classification but also proposed an active learning framework. This framework enables the selection of the most uncertain samples, forwarding them to experts only when the model is uncertain about its decisions. Such an approach allows experts to label only a fraction of the data, thereby enhancing performance and fostering trust in the system. Furthermore, we extended our work to uncover the reasons for uncertainty by employing B-LRP, finding that the sources of uncertainty often stemmed from damaged body parts and a noisy background, where the target object is smaller compared to the background. These insights guide the citizen's efforts in capturing images of mosquitoes, minimising confusion in species identification, and ultimately contributing to the accuracy and efficiency of classification. This multifaceted study underscores the potential of integrating uncertainty into deep learning models, enriching the existing body of knowledge and offering promising directions for future research.

## 8.3 Third Paper

### 8.3.1 Bibliography Entry

Adhane, G., Dehshibi, M. M., & Masip, D. (2022, August). Incorporating Reinforcement Learning for Quality-aware Sample Selection in Deep Architecture Training. In 2022 *IEEE International Conference on Omni-layer Intelligent Systems (COINS)* (pp. 1-5). IEEE.

### 8.3.2 Abstract

Many samples are necessary to train a convolutional neural network (CNN) to achieve optimum performance while maintaining generalisability. Several studies, however, have indicated that not all input data in large datasets are informative for the model, and using them for training can degrade the model's performance and add uncertainty. Furthermore, in some domains, such as medicine, there is insufficient labelled data to train a deep learning model from scratch, necessitating the use of transfer learning

to fine-tune a pretrained model in another domain. This paper proposes a transfer learning strategy based on partially supervised reinforcement learning (RL) to address these concerns by selecting the most informative samples while avoiding negative transfers from the dataset. We conducted several experiments on the benchmark image classification databases MNIST, Fashion-MNIST, and CIFAR-10 to create a fair test harness for assessing the performance of the proposed strategy, which can be extended to explore other domains in the future. The results show that the proposed strategy outperforms the classical training methods.

### 8.3.3 Background

The integration of reinforcement learning with quality-aware sample selection represents a pioneering stride in our ongoing research. This paper builds on the existing body of work that seeks to optimise the training process of deep learning architectures. Our innovative method uniquely utilises reinforcement learning to facilitate the dynamic selection of quality samples, focusing the training process on valuable instances that contribute to overall model accuracy. Presented at the 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS), our research marks a significant advancement in the field of deep learning, contributing new insights and techniques to the broader AI community.

### 8.3.4 New Methods

Our work introduces a novel reinforcement learning-based approach for quality-aware sample selection in deep architecture, we establish a dynamic mechanism that selectively focuses on high-quality samples, ignoring redundant or low-quality instances. This process leads to more efficient training, saving computational resources without sacrificing performance. Our method includes comprehensive experimentation across various domains, including image classification and natural language processing, revealing its applicability and effectiveness in different contexts.

### 8.3.5 Results

Our research culminated in the successful integration of reinforcement learning with quality-aware sample selection, paving the way for enhanced efficiency in deep architecture training. The proposed approach was rigorously tested on several deep learning tasks, revealing significant improvements in training time without a loss in model accuracy. The innovative utilisation of reinforcement learning to dynamically select valuable samples not only optimises computational resources but also provides insights into the potential synergy between reinforcement learning and other areas of artificial intelligence. This work opens new horizons in the ongoing pursuit of efficient and effective deep learning methodologies.

## 8.4 Fourth Paper

### 8.4.1 Bibliography Entry

Dehshibi, M. M., Ashtari-Majlan, M., **Adhane, G.**, & Masip, D. (2023). ADVISE: ADaptive Feature Relevance and VISual Explanations for Convolutional Neural Networks. *The Visual Computer*, Springer. **Q2**.

### 8.4.2 Abstract

To equip Convolutional Neural Networks (CNNs) with explainability, it is essential to interpret how opaque models make specific decisions, understand what causes the errors, improve the architecture design, and identify unethical biases in the classifiers. This paper introduces ADVISE, a new explainability method that quantifies and leverages the relevance of each unit of the feature map to provide better visual explanations. To this end, we propose using adaptive bandwidth kernel density estimation to assign a relevance score to each unit of the feature map with respect to the predicted class. We also propose an evaluation protocol to quantitatively assess the visual explainability of CNN models. We extensively evaluate our idea in the image classification task using AlexNet, VGG16, ResNet50, and Xception pretrained on ImageNet. We compare ADVISE with the state-of-the-art visual explainable methods and show that the proposed method outperforms competing approaches in quantifying feature-relevance and visual explainability while maintaining competitive time complexity. Our experiments further show that ADVISE fulfils the sensitivity and implementation independence axioms while passing the sanity checks.

### 8.4.3 Background

Our research introduces ADVISE, an innovative approach to adaptive feature relevance and visual explanations within the context of Convolutional Neural Networks. The development of ADVISE emanates from the increasing need for interpretability in deep learning models. Our approach uniquely combines statistical methods with visualisation techniques to provide insight into how different features impact decision-making within CNNs. The work is accepted in the Journal of The Visual Computer: International Journal of Computer Graphics and contributes to the broader field of explainable AI, offering a new perspective on transparency and understandability in machine learning models.

### 8.4.4 New Methods

ADVISE represents a novel contribution to the domain of visual explanations for deep learning, employing adaptive feature relevance to enhance understanding of model

decision-making. Our method utilises statistical analysis and tailored visualisation techniques to discern the importance of different features within CNNs. This approach not only allows researchers and practitioners to grasp the underlying mechanics of deep learning models but also fosters a higher degree of trust and reliability in their applications. The potential applications of ADVISE are vast, ranging from healthcare to security, reflecting its versatile and impactful nature.

### 8.4.5 Results

The results of our research highlight the effectiveness and applicability of ADVISE in delivering insightful visual explanations for CNNs. Through comprehensive experimentation and evaluation, we demonstrated that our method provides clear, understandable, and informative insights into how individual features influence model outcomes. These findings signify a substantial advancement in the field of explainable AI, extending the boundaries of transparency and interpretability in deep learning. Our work not only sets the stage for further investigation into the mechanisms of deep learning but also serves as a benchmark for developing responsible and ethical AI systems.

## 8.5 Fifth Paper

### 8.5.1 Title

Adhane, G., Vetter, D., Dehshibi, M. M., Masip, D., & Roig, G. (2024). On Explaining Knowledge Distillation: Measuring and Visualising the Knowledge Transfer Process. Submitted to *ECCV2024*.

### 8.5.2 Abstract

Knowledge distillation (KD) remains challenging due to the opaque knowledge transfer process from a Teacher to a Student. To address this, we introduce *UniCAM*, a novel gradient-based visual explanation method. *UniCAM* visually explains both the features that were learned or overlooked (ignored) by the student during distillation, providing a clear visual interpretation of the knowledge transfer. Extensive experiments on CIFAR10, ASIRRA, and Plant Disease datasets demonstrate its ability to provide a detailed and comprehensive explanation of both distilled and overlooked features. Our analysis reveals that the Student learns to focus on more relevant features in its initial layers, like textures and object parts. In contrast, an equivalent model trained without KD (Base model) has more diffused attention over the whole image, including the background. In the middle and deeper layers, the Student refines its focus localising more salient features and learning even more discriminative features. Furthermore, we propose two novel metrics: the *feature similarity score*

and *relevance score*. The feature similarity score measures the degree of similarity between the features learned by the Student and the Base model, while the relevance score measures the importance of the features for the task. Finally, we experimentally demonstrate that *UniCAM* and these novel metrics provide valuable insights into explaining the KD process.

### 8.5.3 Background

An essential milestone in this research journey was a scholarly visit to Goethe University of Frankfurt, where collaboration was established with the Computational Vision and Artificial Intelligence (CVAI) lab. This experience profoundly shaped the methodologies and insights presented in this paper. The research is a joint effort between the AIWELL group of the Open University of Catalonia (UOC) in Barcelona and the CVAI lab in Frankfurt. This collaboration enabled a more nuanced exploration of Knowledge Distillation (KD), which is central to this work. KD acts as a mechanism where a complex model, the "Teacher," imparts its acquired knowledge to a simpler "Student" model. Leveraging the combined resources and expertise of both the CVAI and AIWELL labs, we employed a range of metrics and visual explanation tools to scrutinise the KD process. Our work aims to make significant contributions to the increasingly important field of transparent and explainable AI systems

### 8.5.4 New Methods

The paper introduces new metrics and visualisation techniques to analyse the Knowledge Distillation process. These methods enable the authors to quantify and visually represent the transfer of knowledge between teacher and student models. By using metrics and visualisation tools, we explore the cases where KD fails as a result of knowledge gap between the teacher and student, contributing to a deeper understanding of deep learning models and enriching the field of AI.

### 8.5.5 Results

The results demonstrate the effectiveness of the proposed methods in explaining the KD process. We conduct extensive experiments with various CNN models and show that our approach provides clear and accurate measurement and visualisation of knowledge transfer. These findings contribute to explainable AI, setting a standard for interpreting complex machine learning processes and emphasising the importance of responsible and transparent AI practice.

## 8.6 Summary of Research Projects and Grants

During my tenure as a PhD researcher at the AIWell Lab, I was actively involved in the execution and management of two prominent research grants:

- **BECAUsE (BEyond model aCcurAcy: Uncertainty, Explainability, and Domain Adaptation)**: BEyond model aCcurAcy: Funded by the Ministerio de Ciencia Innovación y Universidades, RTI2018-095232-B-C21, with a budget of €77,000. The project aimed at advancing machine learning models by going beyond mere accuracy metrics to include aspects like uncertainty quantification, explainability, and domain adaptation.
- **SENTIENT (Responsible Artificial Intelligence for Human Well-being: Contextualized Human-Centric Perception)**. Sponsored by the Ministerio de Ciencia e Innovación, PID2022-138721NB-I00, with a budget of €114,125. This project was focused on developing AI algorithms that are context-aware and human-centric, with the objective of promoting well-being and ethical considerations in AI applications.

These projects provided a platform for groundbreaking research in the fields of machine learning, computer vision, and ethical AI, contributing to the advancement of knowledge and the creation of more robust, reliable, and responsible AI systems. I was also the recipient of the scholarship provided by the Universitat Oberta de Catalunya (UOC) for three years. I would like to acknowledge this financial support, which has been instrumental in the successful completion of my PhD study. This competitive grant has not only validated the academic merit and significance of my research but has also offered invaluable resources that have enriched the quality of work presented in this thesis. It is an honour to contribute to the scholarly community at UOC as a scholarship recipient.

# Bibliography


[1] Qizhe Xie et al. "Self-training with noisy student improves imagenet classification". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10687–10698.

[2] Hugo Touvron et al. "Fixing the train-test resolution discrepancy". In: *Advances in neural information processing systems* 32 (2019).

[3] Mohammad Shoeybi et al. "Megatron-lm: Training multi-billion parameter language models using model parallelism". In: *arXiv preprint arXiv:1909.08053* (2019).

[4] Zhilin Yang et al. "Xlnet: Generalized autoregressive pretraining for language understanding". In: *Advances in neural information processing systems* 32 (2019).

[5] Zachary C Lipton. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3 (2018), pp. 31–57. DOI: 10.1145/3236386.3241340.

[6] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58 (2020), pp. 82–115. DOI: 10.1016/j.inffus.2019.12.012.

[7] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[8] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. "Examples are not enough, learn to criticize! criticism for interpretability". In: *Advances in neural information processing systems* 29 (2016).

[9] Rahul Iyer et al. "Transparency and explanation in deep reinforcement learning neural networks". In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 2018, pp. 144–150.

[10] Joy Buolamwini and Timnit Gebru. "Gender shades: Intersectional accuracy disparities in commercial gender classification". In: *Conference on fairness, accountability and transparency*. PMLR. 2018, pp. 77–91.

[11] Shaima Tilouche, Vahid Partovi Nia, and Samuel Bassetto. "Parallel coordinate order for high-dimensional data". In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 14.5 (2021), pp. 501–515.

[12] Rich Caruana et al. "Case-based explanation of non-case-based learning methods." In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association. 1999, p. 212.

[13] William DuMouchel. "Data squashing: constructing summary data sets". In: *Handbook of Massive Data Sets* (2002), pp. 579–591.

[14] Justin Matejka and George Fitzmaurice. "Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing". In: *Proceedings of the 2017 CHI conference on human factors in computing systems*. 2017, pp. 1290–1294.

[15] Finale Doshi-Velez and Been Kim. "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608* (2017).

[16] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013).

[17] Ruth C. Fong and Andrea Vedaldi. *Interpretable Explanations of Black Boxes by Meaningful Perturbation*. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 3429–3437. https://doi.org/10.1109/ICCV.2017.371. 2017.

[18] Luisa M Zintgraf et al. "Visualizing deep neural network decisions: Prediction difference analysis". In: *arXiv preprint arXiv:1702.04595* (2017).

[19] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. *Axiomatic Attribution for Deep Networks*. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3319–3328. 2017.

[20] Scott M. Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2017, pp. 4768–4777. DOI: 10.5555/3295222.3295230.

[21] Xu Cheng et al. "Explaining knowledge distillation by quantifying the knowledge". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 12925–12935.

[22] Mengqi Xue et al. "KDExplainer: A Task-oriented Attention Model for Explaining Knowledge Distillation". In: *International Joint Conference on Artificial Intelligence*. 2021.

[23] Mariya Toneva et al. "An Empirical Study of Example Forgetting during Deep Neural Network Learning". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[24] Agata Lapedriza et al. "Are all training examples equally valuable?" In: *arXiv preprint arXiv:1311.6510* (2013).

[25] Angelos Katharopoulos and François Fleuret. "Not all samples are created equal: Deep learning with importance sampling". In: *International conference on machine learning*. PMLR. 2018, pp. 2525–2534.

[26] Tyler B Johnson and Carlos Guestrin. "Training deep models faster with robust, approximate importance sampling". In: *Advances in Neural Information Processing Systems* 31 (2018).

[27] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[28] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[29] Gereziher Adhane, Mohammad Mahdi Dehshibi, and David Masip. "Incorporating Reinforcement Learning for Quality-aware Sample Selection in Deep Architecture Training". In: *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*. IEEE. 2022, pp. 1–5.

[30] Gereziher Adhane, Mohammad Mahdi Dehshibi, and David Masip. "On the Use of Uncertainty in Classifying Aedes Albopictus Mosquitoes". In: *IEEE Journal of Selected Topics in Signal Processing* 16.2 (2022), pp. 224–233. DOI: 10.1109/JSTSP.2021.3122886.

[31] Anelia Angelova, Yaser Abu-Mostafam, and Pietro Perona. "Pruning training sets for learning of object categories". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 494–501.

[32] Murat Sensoy et al. "Uncertainty-aware deep classifiers using generative models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 5620–5627. DOI: 10.1609/aaai.v34i04.6015.

[33] Hwanjun Song et al. "Carpe Diem, Seize the Samples Uncertain" at the Moment" for Adaptive Batch Selection". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 1385–1394. DOI: 10.1145/3340531.3411898.

[34] Farid Ghareh Mohammadi, Hamid R Arabnia, and M Hadi Amini. "On parameter tuning in meta-learning for computer vision". In: *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. 2019, pp. 300–305.

[35] Alessandro Achille et al. "Task2vec: Task embedding for meta-learning". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 6430–6439.

[36] Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. "Investigating meta-learning algorithms for low-resource natural language understanding tasks". In: *arXiv preprint arXiv:1908.10423* (2019).

[37] Mike Lewis et al. "Pre-training via Paraphrasing". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 18470–18481. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/d6f1dd034aabde7657e6680444ceff62-Paper.pdf.

[38] Oriol Vinyals et al. "Matching Networks for One Shot Learning". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf.

[39] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical Networks for Few-shot Learning". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/cb8da6767461f2812ae4290eac7cb
Paper.pdf.

[40] Deanna Needell and Rachel Ward. "Batched stochastic gradient descent with weighted sampling". In: *Approximation Theory XV: San Antonio 2016 15*. Springer. 2017, pp. 279–306.

[41] Maya Kabkab, Azadeh Alavi, and Rama Chellappa. "Dcnns on a diet: Sampling strategies for reducing the training set size". In: *arXiv preprint arXiv:1606.04232* (2016).

[42] Xin Chen et al. "Catch: Context-based meta reinforcement learning for transferrable architecture search". In: *European Conference on Computer Vision*. Springer. 2020, pp. 185–202.

[43] Pascal Klink et al. "Self-Paced Deep Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 9216–9227. URL: https://proceedings.neurips.cc/paper/2020/file/68a9750337a418a86fe06c1991a1d64c-Paper.pdf.

[44] Kenan E Ak et al. "Incorporating reinforced adversarial learning in autoregressive image generation". In: *European Conference on Computer Vision*. Springer. 2020, pp. 18–34.

[45] Leonardo C da Cruz et al. "Enabling Autonomous Medical Image Data Annotation: A human-in-the-loop Reinforcement Learning Approach". In: *2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE. 2021, pp. 271–279.

[46] Alex Kendall and Yarin Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/2650d6089a6d640c5e85b2b88265dc2b-Paper.pdf.

[47] Armen Der Kiureghian and Ove Ditlevsen. "Aleatory or epistemic? Does it matter?" In: *Structural safety* 31.2 (2009), pp. 105–112.

[48] Radford M Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 2012.

[49] Yarin Gal and Zoubin Ghahramani. "Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference". In: *Fourth International Conference on Learning Representations, ICLR*. 2016, 42: 1–12.

[50] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in neural information processing systems* 30 (2017).

[51] Solomon Kullback and Richard A Leibler. "On information and sufficiency". In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

[52] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. "Variational inference: A review for statisticians". In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877.

[53] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 2016, pp. 1050–1059.

[54] Ryan-Rhys Griffiths et al. "Achieving robustness to aleatoric uncertainty with heteroscedastic Bayesian optimisation". In: *Machine Learning: Science and Technology* 3.1 (2021), p. 015004.

[55] Burr Settles. "Active learning literature survey". In: (2009).

[56] Ross D King et al. "Functional genomic hypothesis generation and experimentation by a robot scientist". In: *Nature* 427.6971 (2004), pp. 247–252.

[57] David Cohn, Les Atlas, and Richard Ladner. "Improving generalization with active learning". In: *Machine learning* 15 (1994), pp. 201–221.

[58] David D Lewis. "A sequential algorithm for training text classifiers: Corrigendum and additional data". In: *Acm Sigir Forum*. Vol. 29. 2. ACM New York, NY, USA. 1995, pp. 13–19.

[59] Zhao Xu et al. "Representative sampling for text classification using support vector machines". In: *Advances in Information Retrieval: 25th European Conference on IR Research, ECIR 2003, Pisa, Italy, April 14–16, 2003. Proceedings 25*. Springer. 2003, pp. 393–407.

[60] Raphael Schumann and Ines Rehbein. "Active Learning via Membership Query Synthesis for Semi-Supervised Sentence Classification". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 472–481. DOI: 10.18653/v1/K19-1044. URL: https://aclanthology.org/K19-1044.

[61] Chen Change Loy, Tao Xiang, and Shaogang Gong. "Stream-based active unusual event detection". In: *Asian Conference on Computer Vision*. Springer. 2010, pp. 161–175.

[62] Timothy M Hospedales, Shaogang Gong, and Tao Xiang. "Finding rare classes: Active learning with generative and discriminative models". In: *IEEE transactions on knowledge and data engineering* 25.2 (2011), pp. 374–386.

[63] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. "Deep Bayesian Active Learning with Image Data". In: *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 2017, pp. 1183–1192.

[64] Dan Wang and Yi Shang. "A new active labeling method for deep learning". In: *2014 International joint conference on neural networks (IJCNN)*. IEEE. 2014, pp. 112–119.

[65] Zhihao Liang et al. "Exploring Diversity-based Active Learning for 3D Object Detection in Autonomous Driving". In: *CoRR* abs/2205.07708 (2022).

[66] Lile Cai et al. "Exploring spatial diversity for region-based active learning". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 8702–8712.

[67] Sharat Agarwal et al. "Contextual diversity for active learning". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*. Springer. 2020, pp. 137–153.

[68] Wenbin Cai et al. "Active learning for classification with maximum model change". In: *ACM Transactions on Information Systems (TOIS)* 36.2 (2017), pp. 1–28.

[69] Wenbin Cai, Muhan Zhang, and Ya Zhang. "Batch mode active learning for regression with expected model change". In: *IEEE transactions on neural networks and learning systems* 28.7 (2016), pp. 1668–1681.

[70] Yawar Siddiqui, Julien Valentin, and Matthias Nießner. "Viewal: Active learning with viewpoint entropy for semantic segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9433–9443.

[71] Alex Holub, Pietro Perona, and Michael C Burl. "Entropy-based active learning for object recognition". In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE. 2008, pp. 1–8.

[72] Kevin Small and Dan Roth. "Margin-based active learning for structured predictions". In: *International Journal of Machine Learning and Cybernetics* 1 (2010), pp. 3–25.

[73] Melanie Ducoffe and Frederic Precioso. "Adversarial active learning for deep networks: a margin based approach". In: *arXiv preprint arXiv:1802.09841* (2018).

[74] Mingkun Li and Ishwar K Sethi. "Confidence-based active learning". In: *IEEE transactions on pattern analysis and machine intelligence* 28.8 (2006), pp. 1251–1261.

[75] Neil Houlsby et al. "Bayesian active learning for classification and preference learning". In: *arXiv preprint arXiv:1112.5745* (2011).

[76] Jia-Jie Zhu and José Bento. "Generative adversarial active learning". In: *arXiv preprint arXiv:1702.07956* (2017).

[77] Yanshan Xiao, Zheng Chang, and Bo Liu. "An efficient active learning method for multi-task learning". In: *Knowledge-Based Systems* 190 (2020), p. 105137.

[78] Seho Kee, Enrique Del Castillo, and George Runger. "Query-by-committee improvement with diversity and density in batch active learning". In: *Information Sciences* 454 (2018), pp. 401–418.

[79] Burr Settles, Mark Craven, and Soumya Ray. "Multiple-instance active learning". In: *Advances in neural information processing systems* 20 (2007).

[80] Nicholas Roy and Andrew McCallum. "Toward optimal active learning through monte carlo estimation of error reduction". In: *ICML, Williamstown* 2 (2001), pp. 441–448.

[81] Raghav Mehta et al. "Information gain sampling for active learning in medical image classification". In: *International Workshop on Uncertainty for Safe Utilization of Machine Learning in Medical Imaging*. Springer. 2022, pp. 135–145.

[82] Kasra Arnavaz et al. “Bayesian active learning for maximal information gain on model parameters”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 10524–10531.

[83] Boshuang Huang, Sudeep Salgia, and Qing Zhao. “Disagreement-based active learning in online settings”. In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 1947–1958.

[84] Lav Kumar Gupta, Deepika Koundal, and Shweta Mongia. “Explainable Methods for Image-Based Deep Learning: A Review”. In: *Archives of Computational Methods in Engineering* (2023), pp. 1–16.

[85] Matthew D Zeiler and Rob Fergus. *Visualizing and understanding convolutional networks*. In *European Conference on Computer Vision (ECCV)*, pp. 818–833. https://doi.org/10.1007/978-3-319-10590-1_53. 2014.

[86] Jason Yosinski et al. “Understanding neural networks through deep visualization”. In: *arXiv preprint arXiv:1506.06579* (2015).

[87] Andrej Karpathy and Li Fei-Fei. “Deep visual-semantic alignments for generating image descriptions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3128–3137.

[88] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. “Visualizing the Impact of Feature Attribution Baselines”. In: *Distill* 5.1 (2020), e22. DOI: 10.23915/distill.00022.

[89] Bolei Zhou et al. “Learning deep features for discriminative localization”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.

[90] Ramprasaath R Selvaraju et al. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[91] Aditya Chattopadhay et al. *Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks*. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847. https://doi.org/10.1109/WACV.2018.00097. 2018.

[92] Haofan Wang et al. *Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks*. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 111–119. https://doi.org/10.1109/CVPRW50498.2020.00020. 2018.

[93] Peng-Tao Jiang et al. “LayerCAM: Exploring Hierarchical Class Activation Maps for Localization”. In: *IEEE Transactions on Image Processing* 30 (2021), pp. 5875–5888. DOI: 10.1109/TIP.2021.3089943.

[94] Xiangwei Shi et al. “Zoom-cam: Generating fine-grained pixel annotations from image labels”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 10289–10296.

[95] Mohammad Mahdi Dehshibi et al. “ADVISE: ADaptive feature relevance and VISual Explanations for convolutional neural networks”. In: *The Visual Computer* (2023), pp. 1–13.

[96] Vitali Petsiuk, Abir Das, and Kate Saenko. "RISE: Randomized Input Sampling for Explanation of Black-box Models". In: *Proceedings of the British Machine Vision Conference (BMVC)*. 2018.

[97] 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*. 2016, pp. 1135–1144.

[98] Piotr Dabkowski and Yarin Gal. "Real time image saliency for black box classifiers". In: *Advances in neural information processing systems* 30 (2017).

[99] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. "Understanding deep networks via extremal perturbations and smooth masks". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 2950–2958.

[100] Zhongang Qi, Saeed Khorram, and Fuxin Li. "Visualizing Deep Networks by Optimizing with Integrated Gradients." In: *CVPR Workshops*. Vol. 2. 2019, pp. 1–4.

[101] Song Han, Huizi Mao, and William J. Dally. "Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2016. URL: http://arxiv.org/abs/1510.00149.

[102] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. "Data-free knowledge distillation for heterogeneous federated learning". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12878–12889.

[103] Sergey Zagoruyko and Nikos Komodakis. "Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=Sks9_ajex.

[104] Ying Zhang et al. "Deep mutual learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4320–4328.

[105] Peyman Passban et al. "Alp-kd: Attention-based layer projection for knowledge distillation". In: *Proceedings of the AAAI Conference on artificial intelligence*. Vol. 35. 2021, pp. 13657–13665.

[106] Jang Hyun Cho and Bharath Hariharan. "On the efficacy of knowledge distillation". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 4794–4802.

[107] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. "Distilling the knowledge in a neural network". In: *NIPS Deep Learning and epresentation Learning Workshop*. 2015.

[108] Byeongho Heo et al. "A Comprehensive Overhaul of Feature Distillation". In: *International Conference on Computer Vision (ICCV)*. 2019.

[109] Mary Phuong and Christoph Lampert. "Towards understanding knowledge distillation". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5142–5151.

[110] Lei Li et al. "Dynamic Knowledge Distillation for Pre-trained Language Models". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 379–389. DOI: 10.18653/v1/2021.emnlp-main.31. URL: https://aclanthology.org/2021.emnlp-main.31.

[111] Longteng Guo et al. "Non-Autoregressive Image Captioning with Counterfactuals-Critical Multi-Agent Learning". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. Ed. by Christian Bessiere. Main track. International Joint Conferences on Artificial Intelligence Organization, July 2020, pp. 767–773. DOI: 10.24963/ijcai.2020/107. URL: https://doi.org/10.24963/ijcai.2020/107.

[112] Yuenan Hou et al. "Inter-Region Affinity Distillation for Road Marking Segmentation". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 12483–12492. DOI: 10.1109/CVPR42600.2020.01250.

[113] Qizhen Lan and Qing Tian. "Gradient-Guided Knowledge Distillation for Object Detectors". In: *CoRR* abs/2303.04240 (2023). DOI: 10.48550/arXiv.2303.04240. URL: https://doi.org/10.48550/arXiv.2303.04240.

[114] Raed Alharbi, Minh N Vu, and My T Thai. "Learning Interpretation with Explainable Knowledge Distillation". In: *2021 IEEE International Conference on Big Data (Big Data)*. IEEE. 2021, pp. 705–714. DOI: 10.1109/BigData52589.2021.9671988.

[115] Seunghyun Lee and Byung Cheol Song. "Interpretable embedding procedure knowledge transfer via stacked principal component analysis and graph neural network". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 9. 2021, pp. 8297–8305.

[116] Seunghyun Lee and Byung Cheol Song. "Graph-based Knowledge Distillation by Multi-head Attention Network". In: *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*. BMVA Press, 2019, p. 141. URL: https://bmvc2019.org/wp-content/uploads/papers/0821-paper.pdf.

[117] Abhishek Gupta et al. "Meta-reinforcement learning of structured exploration strategies". In: *Advances in neural information processing systems* 31 (2018).

[118] Bernie Wang et al. "Improving Context-Based Meta-Reinforcement Learning with Self-Supervised Trajectory Contrastive Learning". In: *CoRR* abs/2103.06386 (2021). URL: https://arxiv.org/abs/2103.06386.

[119] Zohar Rimon, Aviv Tamar, and Gilad Adler. "Meta Reinforcement Learning with Finite Training Tasks - a Density Estimation Approach". In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 13640–13653.

[120] Shuyue Chen, Ran Wang, and Jian Lu. "A meta-framework for multi-label active learning based on deep reinforcement learning". In: *Neural Networks* 162 (2023), pp. 258–270. DOI: https://doi.org/10.1016/j.neunet.2023.02.045.

[121] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. "Cifar-10 (canadian institute for advanced research)". In: *URL http://www. cs. toronto. edu/kriz/cifar. html* 5.4 (2010), p. 1.

[122] Li Deng. "The mnist database of handwritten digit images for machine learning research". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[123] Han Xiao, Kashif Rasul, and Roland Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms". In: *arXiv preprint arXiv:1708.07747* (2017). arXiv: cs.LG/1708.07747 [cs.LG].

[124] RICHARD BELLMAN. "A Markovian Decision Process". In: *Journal of Mathematics and Mechanics* 6.5 (1957), pp. 679–684. ISSN: 00959057, 19435274. URL: http://www.jstor.org/stable/24900506.

[125] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[126] Mosquito Alert. *Mosquito Alert: A citizen platform for studying and controlling mosquitos, which transmit global diseases*. http://www.mosquitoalert.com/en/. [Online; accessed May 2021]. 2014.

[127] Yarin Gal and Zoubin Ghahramani. "A theoretically grounded application of dropout in recurrent neural networks". In: *Advances in neural information processing systems* 29 (2016).

[128] Gereziher Adhane, Mohammad Mahdi Dehshibi, and David Masip. "A Deep Convolutional Neural Network for Classification of Aedes Albopictus Mosquitoes". In: *IEEE Access* 9 (2021), pp. 72681–72690. DOI: 10.1109/ACCESS.2021.3079700.

[129] Yingzhen Li and Yarin Gal. "Dropout Inference in Bayesian Neural Networks with Alpha-divergences". In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. 2017, pp. 2052–2061.

[130] Kirill Bykov et al. *How Much Can I Trust You? – Quantifying Uncertainties in Explaining Neural Networks*. 2020. arXiv: 2006.09000 [cs.LG].

[131] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.

[132] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019, pp. 8024–8035.

[133] Balint Armin Pataki et al. “Deep learning identification for citizen science surveillance of tiger mosquitoes”. In: *Scientific reports* 11.1 (2021), pp. 1–12.

[134] Filipe Condessa, José Bioucas-Dias, and Jelena Kovačević. “Performance measures for classification systems with rejection”. In: *Pattern Recognition* 63 (2017), pp. 437–450.

[135] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012). In *Advances in Neural Information Processing Systems*, pp. 1097–1105, pp. 1097–1105.

[136] Azali Azlan et al. “Genome-wide identification of Aedes albopictus long noncoding RNAs and their association with dengue and Zika virus infection”. In: *PLoS neglected tropical diseases* 15.1 (2021), e0008351.

[137] Eleni Patsoula et al. “Molecular and morphological characterization of Aedes albopictus in northwestern Greece and differentiation from Aedes cretinus and Aedes aegypti”. In: *Journal of medical entomology* 43.1 (2006), pp. 40–54.

[138] Mohammad Mahdi Dehshibi and Jamshid Shanbehzadeh. “Cubic norm and kernel-based bi-directional PCA: toward age-aware facial kinship verification”. In: *The Visual Computer* 35 (2019), pp. 23–40. DOI: 10.1007/s00371-017-1442-1.

[139] Weitao Chen et al. “GCSANet: A Global Context Spatial Attention Deep Learning Network for Remote Sensing Scene Classification”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15 (2022), pp. 1150–1162. DOI: 10.1109/JSTARS.2022.3141826.

[140] Weitao Chen et al. “JAGAN: A Framework for Complex Land Cover Classification Using Gaofen-5 AHSI Images”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15 (2022), pp. 1591–1603. DOI: 10.1109/JSTARS.2022.3144339.

[141] Mingxing Tan, Ruoming Pang, and Quoc V. Le. *EfficientDet: Scalable and Efficient Object Detection*. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10778–10787. https://doi.org/10.1109/CVPR42600.2020.01079. 2020.

[142] Dana Alsagheer et al. *Detecting Hate Speech Against Athletes in Social Media*. In *IEEE International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pp. 75–81. https://doi.org/10.1109/IDSTA55301.2022.9923132. 2022.

[143] Sachin Mehta et al. *ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation*. In *European Conference on Computer Vision (ECCV)*, pp. 561–580. https://doi.org/10.1007/978-3-030-01249-6_34. 2018.

[144] Mona Ashtari-Majlan, Abbas Seifi, and Mohammad Mahdi Dehshibi. “A Multi-Stream Convolutional Neural Network for Classification of Progressive MCI in Alzheimer’s Disease Using Structural MRI Images”. In: *IEEE Journal of Biomedical and Health Informatics* 26.8 (2022), pp. 3918–3926. DOI: 10.1109/JBHI.2022.3155705.

[145] Long Chen et al. *SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning*. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6298–6306. https://doi.org/10.1109/CVPR.2017.667. 2017.

[146] Mohammad Mahdi Dehshibi et al. "A deep multimodal learning approach to perceive basic needs of humans from Instagram profile". In: *IEEE Transactions on Affective Computing* (2021), pp. 1–13. DOI: 10.1109/TAFFC.2021.3090809.

[147] Riccardo Guidotti et al. "A survey of methods for explaining black box models". In: *ACM computing surveys (CSUR)* 51.5 (2018), 93:1–42. DOI: 10.1145/3236009.

[148] Quanshi Zhang et al. "Interpreting CNNs via Decision Trees". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 6254–6263. DOI: 10.1109/CVPR.2019.00642.

[149] Sangwon Kim, Mira Jeong, and Byoung Chul Ko. "Lightweight surrogate random forest support for model simplification and feature relevance". In: *Applied Intelligence* (2021), pp. 1–11. DOI: 10.1007/s10489-021-02451-x.

[150] Alexander Binder et al. "Layer-Wise Relevance Propagation for Neural Networks with Local Renormalization Layers". In: *Artificial Neural Networks and Machine Learning – ICANN 2016*. Springer International Publishing, 2016, pp. 63–71. DOI: 10.1007/978-3-319-44781-0_8.

[151] Anh Nguyen et al. "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2016, pp. 3395–3403. DOI: 10.5555/3157382.3157477.

[152] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. "Explainable artificial intelligence: understanding, visualizing and interpreting deep learning models". In: *ITU JOURNAL: ICT DISCOVERIES* 1.S1 (2017), pp. 39–48.

[153] Aravindh Mahendran and Andrea Vedaldi. *Understanding deep image representations by inverting them*. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5188–5196. https://doi.org/10.1109/CVPR.2015.7299155. 2015.

[154] Yixuan Li et al. *Convergent Learning: Do different neural networks learn the same representations?* In *4th International Conference on Learning Representations (ICLR)*, pp. 196–212. 2016.

[155] Ahmed Taha et al. *A Generic Visualization Approach for Convolutional Neural Networks*. In *European Conferenc eon Computer Vision (ECCV)*, pp. 734–750. https://doi.org/10.1007/978-3-030-58520-4_43. 2020.

[156] Min Lin, Qiang Chen, and Shuicheng Yan. *Network in network*. In *Second International Conference on Learning Representations, (ICLR)*, pp. 1–10. 2014.

[157] Jeffrey Donahue et al. "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.4 (2017), pp. 677–691. DOI: 10.1109/TPAMI.2016.2599174.

[158] Sungyong Seo et al. *Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction*. In *Proceedings of the 11th ACM Conference on Recommender Systems*, pp. 297–305. https://doi.org/10.1145/3109859.3109890. 2017.

[159] David Bau et al. *Network Dissection: Quantifying Interpretability of Deep Visual Representations*. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3319–3327. https://doi.org/10.1109/CVPR.2017.354. 2017.

[160] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. "Feature visualization". In: *Distill* 2.11 (2017), e7. DOI: 10.23915/distill.00007.

[161] Abel Gonzalez-Garcia, Davide Modolo, and Vittorio Ferrari. "Do Semantic Parts Emerge in Convolutional Neural Networks?" In: *International Journal of Computer Vision* 126.5 (2018), pp. 476–494. DOI: 10.1007/s11263-017-1048-0.

[162] Bolei Zhou et al. *Object Detectors Emerge in Deep Scene CNNs*. In *3rd International Conference on Learning Representations (ICLR)*, pp. 1–12. 2015.

[163] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. In *3rd International Conference on Learning Representations (ICLR)*, pp. 1–14. 2015.

[164] François Chollet. *Xception: Deep Learning with Depthwise Separable Convolutions*. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807. https://doi.org/10.1109/CVPR.2017.195. 2017.

[165] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. *Learning Important Features through Propagating Activation Differences*. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3145–3153. https://doi.org/10.5555/3305890.3306006. 2017.

[166] Quan Zheng et al. *Shap-CAM: Visual Explanations for Convolutional Neural Networks Based on Shapley Value*. In *17th European Conference on Computer Vision – ECCV 2022*, pp. 459–474. https://doi.org/10.1007/978-3-031-19775-8_27. 2022.

[167] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90. 2016, pp. 770–778.

[168] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[169] Emanuel Parzen. "On Estimation of a Probability Density Function and Mode". In: *The Annals of Mathematical Statistics* 33.3 (1962), pp. 1065–1076. DOI: 10.1214/aoms/1177704472.

[170] Hideaki Shimazaki and Shigeru Shinomoto. "Kernel bandwidth optimization in spike rate estimation". In: *Journal of Computational Neuroscience* 29.1 (2010), pp. 171–182. DOI: 10.1007/s10827-009-0180-4.

[171] Adrian W. Bowman. "An alternative method of cross-validation for the smoothing of density estimates". In: *Biometrika* 71.2 (1984), pp. 353–360. DOI: 10.1093/biomet/71.2.353.

[172] Elizbar A. Nadaraya. "On Estimating Regression". In: *Theory of Probability & Its Applications* 9.1 (1964), pp. 141–142. DOI: 10.1137/1109020.

[173] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.

[174] Xiao-Hui Li et al. *An Experimental Study of Quantitative Evaluations on Saliency Methods*. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3200–3208. https://doi.org/10.1145/3447548.3467148. 2021.

[175] Ali Borji and James Tanner. "Reconciling Saliency and Object Center-Bias Hypotheses in Explaining Free-Viewing Fixations". In: *IEEE Transactions on Neural Networks and Learning Systems* 27.6 (2015), pp. 1214–1226. DOI: 10.1109/TNNLS.2015.2480683.

[176] Christian Wolf and Markus Lappe. "Salient objects dominate the central fixation bias when orienting toward images". In: *Journal of vision* 21.8 (2021), pp. 23–23. DOI: 10.1167/jov.21.8.23.

[177] Julius Adebayo et al. *Sanity Checks for Saliency Maps*. In *Advances in Neural Information Processing Systems*, pp. 1–11. 2018.

[178] Sara Hooker et al. "A benchmark for interpretability methods in deep neural networks". In: *Advances in neural information processing systems* 32 (2019). In *Advances in Neural Information Processing Systems*, pp. 1–12.

[179] Mukund Sundararajan and Amir Najmi. *The Many Shapley Values for Model Explanation*. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 9269–9278. 2020.

[180] Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. *Feature relevance quantification in explainable AI: A causal problem*. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pp. 2907–2916. 2020.

[181] Gábor J Székely and Maria L Rizzo. "Partial distance correlation with methods for dissimilarities". In: *The Annals of Statistics* 42.6 (2014), pp. 2382–2412.

[182] Xingjian Zhen et al. "On the versatile uses of partial distance correlation in deep learning". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*. Springer. 2022, pp. 327–346.

[183] Jianping Gou et al. "Knowledge distillation: A survey". In: *International Journal of Computer Vision* 129 (2021), pp. 1789–1819.

[184] Mehdi Rezagholizadeh et al. "Pro-kd: Progressive distillation by following the footsteps of the teacher". In: *Proceedings of the 29th International Conference on Computational Linguistics*. 2022, pp. 4714–4727.

[185] Yao Rong et al. "A Consistent and Efficient Evaluation Strategy for Attribution Methods". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 18770–18795.

[186] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[187] Jeremy Elson et al. "Asirra: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization". In: *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, Inc., 2007.

[188] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. Toronto, ON, Canada, 2009.

[189] David P. Hughes and Marcel Salathé. "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing". In: *CoRR* abs/1511.08060 (2015).

[190] Adam Paszke et al. "Automatic differentiation in PyTorch". In: *NIPS-W*. 2017.

[191] Het Shah et al. *KD-Lib: A PyTorch library for Knowledge Distillation, Pruning and Quantization*. 2020. arXiv: 2011.14691 [cs.LG].

[192] Jacob Gildenblat and contributors. *PyTorch library for CAM methods*. https://github.com/jacobgil/pytorch-grad-cam. 2021.

[193] Samuel Stanton et al. "Does Knowledge Distillation Really Work?" In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 6906–6919. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/376c6b9ff3bedbbea56751a84fffc10c-Paper.pdf.

[194] Seyed Iman Mirzadeh et al. "Improved knowledge distillation via teacher assistant". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 2020, pp. 5191–5198.

[195] Wonchul Son et al. "Densely Guided Knowledge Distillation Using Multiple Teacher Assistants". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9395–9404.