LABORATOIRE
D'INFORMATIQUE
& DES SYSTÈMES
UMR 7020

Centrale Marseille

NNT/NL : 2020AIXM0001/001ED184

THÈSE DE DOCTORAT

Soutenue à Aix-Marseille Université
Septembre 2024 par

Felipe TORRES FIGUEROA

Learning discriminative representations to interpret image recognition models

Spécialité
Mathématiques

École doctorale
Mathematiques et Informatique (184)

Laboratoire/Partenaires de recherche
Laboratoire d'Informatique & Systèmes
École Centrale Marseille
Aix-Marseille Université

Composition du jury

Frédéric JURIE, PR Univ. Caen Normandie	Rapporteur et Président de jury
Giorgos TOLIAS, Associate Professor CTU Prague	Rapporteur
Frédéric PRECIOSO, PR Université Côte d' Azur	Examinateur
Diane LARLUS, Principal Research Scientist and Team Lead NAVER Labs	Examinatrice
Stephane AYACHE Centrale Marseille, Aix Marseille Univ, CNRS, LIS.	Directeur de thèse
Ronan SICRE Centrale Marseille, Aix Marseille Univ, CNRS, LIS.	Co-directeur de thèse

LABORATOIRE
D'INFORMATIQUE
& DES SYSTÈMES
UMR 7020

Centrale Marseille

Affidavit

I, undersigned, Felipe Torres Figueroa, hereby declare that the work presented in this manuscript is my own work, carried out under the scientific direction of Stephane Ayache, Ronan Sicre and Yannis Avrithis; in accordance with the principles of honesty, integrity and responsibility inherent to the research mission. The research work and the writing of this manuscript have been carried out in compliance with both the french national charter for Research Integrity and the Aix-Marseille University charter on the fight against plagiarism.

This work has not been submitted previously either in this country or in another country in the same or in a similar version to any other examination body.

Marseille, June-11th, 2024

Liste de publications et participation aux conférences

Liste des publications réalisées dans le cadre du projet de thèse:

1. (Forthcoming) H. Zhang, F. Torres, R. Sicre, Y. Avrithis, S. Ayache. Opti-CAM: Optimizing saliency maps for interpretability. In *CVIU Journal, 2024*

2. F. Torres, H. Zhang, R. Sicre, Y. Avrithis, S. Ayache. A Learning Paradigm for Interpretable Gradients. In 19th *Internacional Conference on Computer Vision Theory and Applications (VISAPP). 2024. Oral Presentation*

3. F. Torres, H. Zhang, R. Sicre, S. Ayache, Y. Avrithis. CA-Stream: Attention-based pooling for interpretable image recognition. In *CVPR Workshop on Explainable AI. 2024*

Participation aux conférences et écoles d'été au cours de la période de thèse:

1. ELLIS Summer School on Large-Scale AI for Research and Industry. Modène, Italie. 2023.

2. 19th Internacional Conference on Computer Vision Theory and Applications (VISAPP). Rome, Italie. 2024.

3. 3rd Workshop of Explainable AI for Computer Vision at CVPR. Seattle, États-Unis d'Amérique. 2024.

Résumé

Les capacités de vision par ordinateur se sont améliorées au cours de la dernière décennie, une meilleure utilisation du matériel permettant aux ordinateurs de traiter davantage d'images plus rapidement, entraînant l'avènement de l'apprentissage profond. De plus, au cours de cette période, des architectures de modèles telles que les réseaux de neurones convolutifs et les transformers ont été introduites, permettant aux applications de vision par ordinateur de réaliser des tâches plus complexes. En particulier, les modèles de reconnaissance d'images sont désormais capables d'identifier et de reconnaître des éléments sur une image, même dans des conditions difficiles. Ces facteurs ont contribué à l'introduction de ces modèles dans la société.

Avec la diffusion des technologies de l' apprentissage profond au sein de la société, une nouvelle exigence a émergé pour ces méthodologies. Puisqu 'elles interagissent désormais et affectent directement les vies humaines, il est impératif de comprendre leur fonctionnement et de fournir des explications. Pour répondre à ces questions, un nouveau domaine de recherche a vu le jour: l 'interprétabilité et l 'IA explicable.

Dans cette thèse, notre objectif est de comprendre et de développer des modèles d'interprétabilité pour les modèles de reconnaissance d'images de pointe. Nous présentons et expliquons brièvement certains des modèles de reconnaissance d'images les plus performants et pertinents pour les Réseaux de Neurones Convolutifs et les Transformers. Ensuite, nous examinons les approches actuelles en matière d' interprétabilité conçues pour fournir des explications, ainsi que leurs protocoles d' évaluation. Nous faisons des observations sur ces méthodes et protocoles d'évaluation, mettant en évidence les difficultés rencontrées et suggérant des idées pour surmonter leurs limitations.

Opti-CAM Notre première contribution, s' appuie sur le raisonnement des Cartes d' Activation de Classe. En particulier, cette proposition optimise le coefficient de pondération requis pour calculer une carte de saillance, générant une représentation qui maximise la probabilité spécifique à la classe. Cette carte de saillance offre les meilleurs résultats selon les mesures d'interprétabilité, et met en évidence que le contexte est pertinent pour décrire une prédiction. De plus, une nouvelle métrique pour compléter l'évaluation de l'interprétabilité est dévoilée, remédiant aux lacunes de cette procédure.

Cross Attention Stream Notre deuxième contribution, est un ajout aux modèles actuels de reconnaissance d’images, améliorant les mesures d’interprétabilité. Inspiré par des modèles novateurs performants tels que les Transformers, nous construisons un flux qui calcule l’interaction d’une représentation de classe abstraite avec les caractéristiques profondes des réseaux neuronaux convolutionnels. Cette représentation est finalement utilisée pour effectuer la classification. Notre flux affiche des améliorations lors de l’évaluation quantitative, tout en préservant les performances de reconnaissance à travers différents modèles.

Gradient Denoise Enfin, notre dernière contribution présente un nouveau paradigme d’ entraînement pour les réseaux neuronaux profonds. De plus, ce paradigme débruite les informations de gradient des modèles profonds dans l’espace d’entrée. La représentation de rétropropagation guidée de l’image d’entrée est utilisée pour régulariser les modèles lors de leur phase d’ entraînement. En conséquence, nos modèles entraînés affichent des améliorations pour l’ évaluation interprétable. Nous appliquons notre paradigme à de petites architectures dans un cadre contraint, ouvrant la voie au développement futur dans des ensembles de données à grande échelle, ainsi qu’avec des modèles plus complexes.

Mots clés: Apprentissage Profond, reconaissance d’image, interpretabilité, explicabilité.

Abstract

Computer Vision capabilities improved in the past decade, a better utilization of hardware enabled computers to process more images faster, ensuing the dawn of deep learning. Moreover, over this timespan, model architectures such as convolutional neural networks and transformers have been introduced, enabling computer vision applications to conduct more complex tasks. In particular, image recognition models are now capable of identifying and recognizing elements on an image, even on challenging conditions. These factors have contributed towards the introduction of these models into society.

With the permeation of deep learning technologies within society, a new requirement emerged for these methodologies. Since they are now interacting and affecting human lives directly, it is mandatory to understand their functioning and provide explanations. To address these questions a new research field has emerged: interpretability and explainable AI.

In this thesis, our goal is to understand and further develop interpretability models for state-of-the-art image recognition models. We introduce and briefly explain some of the most relevant high performance image recognition models for both Convolutional Neural Networks and Transformers. Then, current interpretability approaches designed to provide explanations, as well as their evaluation protocols. We make observations upon these methods and evaluation protocols, highlighting difficulties upon them and suggesting ideas to address their limitations. In the following chapters we present our contributions.

Opti-CAM Our first contribution, builds upon the reasoning of Class Activation Mappings. In particular, this proposal optimizes the weighting coefficient required to compute a saliency map, generating a representation that maximizes class specific probability. This saliency map performs the best across interpretability metrics on multiple datasets. Plus, it highlights that context is relevant towards describing a prediction. Additionally, a novel metric to complement interpretability evaluation is unveiled, addressing shortcomings in this procedure.

Cross Attention Stream Our second contribution, is an addition to current image recognition models, enhancing interpretability measurements. Inspired novel high performing models such as Transformers, we construct a stream that computes the interaction of an abstract class representation, with deep features of convolutional neural networks. This representation is ultimately used to perform classification. Our Stream displays improvements on quantitative evaluation, as well as preserves recognition performance across different models.

Gradient Denoising Lastly, our final contribution presents a novel training paradigm for deep neural networks. Moreover, this paradigm denoises the gradient information of deep models in the input space. The guided backpropagation representation of the input image is used to regularize models during their training phase. As a result, our trained models display improvements for interpretable evaluation. We apply our paradigm to small architectures in a constrained setting, paving the way for future development in large scale datasets, as well as with more complex models.

Keywords: Deep Learning, image recognition, interpretability, explainability.

Remerciements

Je souhaite commencer mes remerciements avec ma famille: mes parents, mon frère et ma sœur. Avec eux je trouve la motivation pour continuer chaque jour, en plus; ils ont tous m'enseigné l'amour pour mon travail et l'importance de bien utiliser le connaisance pour le bien être de toutes les personnes. Je souhaite remercier aussi **Pablo Arbeláez**, directeur du CINFONIA mon premier tuteur sur l'IA, un des prèmieres personnes qu'ont cru à moi pour faire la recherche et qui m'a donné les bases pour pursuivre cette domain.

Contents

Affidavit 2

Liste de publications et participation aux conférences 3

Résumé 4

Abstract 6

Remerciements 8

Contents 9

List of Figures 11

List of Tables 13

Liste des acronymes 15

Introduction 17

1 Background 23
- 1.1 Introduction 23
- 1.2 Image Recognition Models 26
  - 1.2.1 Traditional Image Recognition models 27
  - 1.2.2 Convolutional Neural Networks 28
  - 1.2.3 Self-Attention Based Architectures 33
  - 1.2.4 Hybrid Architectures 36
  - 1.2.5 Discussion 38
- 1.3 Interpretability 39
  - 1.3.1 Transparency 41
  - 1.3.2 Post-Hoc Interpretability 44
  - 1.3.3 Evaluating Interpretability 51
  - 1.3.4 Discussion 57

2 Opti-CAM: Optimizing saliency maps for interpretability 60
- 2.1 Introduction 60
- 2.2 Motivation 61
- 2.3 Opti-CAM 62
- 2.4 Average Gain 64

2.5 Experiments . . . 64
2.5.1 Implementation details . . . 64
2.5.2 Datasets . . . 65
2.5.3 Evaluation . . . 65
2.6 Qualitative Evaluation . . . 66
2.7 Quantitaive Evaluation . . . 67
2.7.1 Image classification . . . 67
2.7.2 Weakly Supervised Approach . . . 71
2.8 Ablation study . . . 73
2.9 Discussion . . . 74
2.10 Conclusion . . . 75

3 CA-Stream: Attention-based pooling for interpretable image recognition 77
3.1 Introduction . . . 77
3.2 Cross Attention . . . 78
3.3 Cross Attention Stream . . . 80
3.4 Experimental Set-Up . . . 81
3.5 Qualitative Evaluation . . . 83
3.6 Quantitative Evaluation . . . 84
3.6.1 Classification accuracy . . . 86
3.6.2 Ablation . . . 87
3.7 Discussion . . . 90
3.8 Conclusion . . . 91

4 A learning paradigm for interpretable gradients 92
4.1 Introduction . . . 92
4.2 Interpretable Gradients . . . 93
4.3 Experiments . . . 95
4.3.1 Experimental Set-up . . . 96
4.3.2 Qualitative Evaluation . . . 96
4.3.3 Quantitative Evaluation . . . 97
4.3.4 Ablation Experiments . . . 98
4.4 Discussion . . . 99
4.5 Conclusion . . . 100

Conclusion 102
4.1 General remarks . . . 102
4.2 Future Work . . . 105

Bibliography 108

List of Figures

1.1 **AI act pyramid of risk levels**. Adapted from https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai 26

1.2 **Illustration of the convolution operation.** A kernel g operates over feature map f generating an updated representation $f \star g$ (M. Lin et al. 2013) 29

1.3 **Timeline** of milestone Deep Learning models. 30

1.4 **Visual representation** of Global Average Pooling (M. Lin et al. 2013). . 31

1.5 **Generalities** of the ResNet architecture. 31

1.6 **Illustration of the DenseNet architecture** (G. Huang et al. 2017) 32

1.7 **Transformer** self attention variants and encoder block. 34

1.8 **ViT** overview (Dosovitskiy et al. 2020). 35

1.9 **Conformer architecture** overview. (Peng et al. 2021) 37

1.10 **Malik's** three R of computer vision 38

1.11 **Proportion of articles published on ImageNet,** using image recognition models as backbone across the years. Original from https://paperswithcode.com/method/resnet 39

1.12 **Filter response** to learned classes alongside CNN depth (H. Lee et al. 2009). 47

1.13 **CAM** based methodologies overview. 48

1.14 **Score-CAM** computation process (H. Wang, Du, et al. 2019) 50

2.1 **Overview of Opti-CAM**. We are given an input image $\mathbf{x}$, a fixed network f, a target layer ℓ and a class of interest c. We extract the feature maps from layer ℓ and obtain a saliency map $S_\ell(\mathbf{x};\mathbf{u})$ by forming a convex combination of the feature maps ($\times$) with weights determined by a variable vector $\mathbf{u}$ (2.4). After upsampling and normalizing, we element-wise multiply ($\odot$) the saliency map with the input image to form a "masked" version of the input, which is fed to f. The objective function $F^c_\ell(\mathbf{x};\mathbf{u})$ measures the logit of class c for the masked image (2.6). We find the value of $\mathbf{u}^*$ that maximizes this logit by optimizing along the path highlighted in blue (2.5), as well as the corresponding optimal saliency map $S_\ell(\mathbf{x};\mathbf{u}^*)$ (2.7). 63

2.2 **Saliency maps obtained** by different methods for ImageNet (top two rows), Chest X-ray (row 3) and Kvasir (row 4) with VGG. Ground truth class shown on the left of the input image. 66

2.3 **Failure examples** of Opti-CAM regarding insertion/deletion. 70

3.1 **Visualization of eq. (3.2).** On the left, a feature tensor $\mathbf{F} \in \Re^{w \times h \times d}$ is multiplied by the vector $\boldsymbol{\alpha} \in \Re^{d}$ in the channel dimension, like in 1×1 convolution, where $w \times h$ is the spatial resolution and d is the number of channels. This is *cross attention* (CA) (Dosovitskiy et al. 2020) between the query $\boldsymbol{\alpha}$ and the key $\mathbf{F}$. On the right, a linear combination of feature maps $F^1, \dots, F^d \in \Re^{w \times h}$ is taken with weights $\alpha_1, \dots, \alpha_d$. This is a *class activation mapping* (CAM) (B. Zhou, Khosla, et al. 2016) with class agnostic weights. Eq. (3.2) expresses the fact that these two quantities are the same, provided that $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d)$ and $\mathbf{F}$ is reshaped as $F = (\mathbf{f}^1 \dots \mathbf{f}^d) \in \Re^{p \times d}$, where $p = wh$ and $\mathbf{f}^k = \text{vec}(F^k) \in \Re^{p}$ is the vectorized feature map of channel k. 79

3.2 **Cross Attention Stream (CA-Stream) applied to ResNet-based architectures.** Given a network f, we replace global average pooling (GAP) by a learned, attention-based pooling mechanism implemented as a stream in parallel to f. The feature tensor $F_\ell \in \Re^{p_\ell \times d_\ell}$ (*key*) obtained by stage Res-ℓ of f interacts with a CLS token (*query*) embedding $\mathbf{q}_\ell \in \Re^{d_\ell}$ in block CA-ℓ, which contains cross attention (3.3) followed by a linear projection (3.7) to adapt to the dimension of $F_{\ell+1}$. Here, p_ℓ is the number of patches (spatial resolution) and d_ℓ the embedding dimension. The query is initialized by a learnable parameter $\mathbf{q}_0 \in \Re^{d_0}$, while the output $\mathbf{q}_5$ of the last cross attention block is used as a global image representation into the classifier. 80

3.3 **Comparison of saliency maps** generated by different CAM-based methods, using GAP and our CA-Stream, on ImageNet images. The raw attention is the one used for pooling by CA-Stream. 83

3.4 **Raw attention maps** obtained from our CA-Stream on images of the MIT 67 Scenes dataset (Quattoni et al. 2009) on classes that do not exist in ImageNet. The network sees them at inference for the first time. . . 84

4.1 **Interpretable gradient learning**. For an input image x, we obtain the logit vector $y = f(x;\theta)$ by a forward pass through the network f with parameters θ. We compute the classification loss L_C by softmax and cross-entropy (4.1), (4.2). We obtain the standard gradient $\partial L_C / \partial x$ and guided gradient $\partial_G L_C / \partial x$ by two backward passes (dashed) and compute the regularization loss L_R as the error between the two (4.3),(4.5)-(4.7). The total loss is $L = L_C + \lambda L_R$ (4.4). Learning is based on $\partial L / \partial \theta$, which involves differentiation of the entire computational graph except the guided backpropagation branch (blue). 93

4.2 **Gradient comparison** of standard *vs.* our training on CIFAR-100 examples. 96

4.3 **Saliency map comparison** of standard *vs.* our training using different CAM-based methods on CIFAR-100 examples. 97

List of Tables

1.1 **Milestone Image Recognition Methods** Details of milestone Image recognition models in the era of Deep Learning on ImageNet 1k. ResNet* refers to the version using updated training protocols (Wightman et al. 2021) 30

1.2 **Interpretability approaches** grouping using Lipton (Lipton 2018) and Zhang's definitions (Y. Zhang et al. 2021) 41

1.3 **Interpretability Metrics** according to evaluation types 52

2.1 **Classification metrics** on ImageNet validation set, using CNNs. AD/AI: average drop/increase (Chattopadhay et al. 2018); AG: average gain (ours); ↓ / ↑: lower / higher is better; T: Average time (sec) per batch of 8 images. Bold: best, excluding Fake-CAM. 67

2.2 **Classification metrics** on ImageNet validation set, using transformers. AD/AI: average drop/increase AG: average gain (ours); ↓ / ↑: lower / higher is better. T: Average time (sec) per batch of 8 images. Bold: best, excluding Fake-CAM. 68

2.3 **AOPC/MS/ADCC** scores on ImageNet validation set. 69

2.4 **I/D: insertion/deletion** (Vitali et al. 2018) scores on ImageNet validation set; ↓ / ↑: lower / higher is better. 69

2.5 **Bounding box** study. Classification metrics on ImageNet validation set using VGG16. *B*: ground-truth box used by localization metrics; *I*: entire image; *S*: saliency map. AD/AI: average drop/increase (Chattopadhay et al. 2018); AG: average gain (ours); ↓ / ↑: lower / higher is better; bold: best, excluding Fake-CAM. 71

2.6 **Localization metrics** on ImageNet validation set. OM: *official metric*; LE: *localization error*; F1: *pixel-wise F_1 score*; BA: box accuracy; SP: standard pointing game; EP: energy pointing game; SM: *saliency metric.* ↓ / ↑: lower / higher is better. Bold: best, excluding Fake-CAM. 72

2.7 **Localization metrics** with ViT and DeiT on ImageNet validation set. OM: *official metric*; LE: *localization error*; F1: *pixel-wise F_1 score*; BA: box accuracy; SP: standard pointing game; EP: energy pointing game; SM: *saliency metric.* ↓ / ↑: lower / higher is better. Bold: best, excluding Fake-CAM. 72

2.8 **Ablation study** using VGG16 on 1000 images of ImageNet validation set. AD/AI: average drop/increase (Chattopadhay et al. 2018); AG: average gain (ours); ↓ / ↑: lower / higher is better; bold: best, excluding Fake-CAM. 74

3.1 **Interpretability metrics** of CA-Stream *vs.* baseline GAP for different networks and interpretability methods on ImageNet. 85
3.2 **Recognition and Interpretability metrics:** Evaluation between CA-Stream *vs.* baseline GAP for ResNet-50 on CUB and Pascal dataset. . . . 86
3.3 **Accuracy and parameters** of CA-Stream *vs.* baseline GAP for different networks and interpretability methods on ImageNet. #PARAM: total parameters; PARAM%: percentage of CA-Stream parameters relative to backbone. 87
3.4 **Different cross attention block design for CA-Stream.** Classification accuracy and parameters using ResNet-50 on ImageNet. #PARAM: parameters of CA-Stream only. 88
3.5 **Effect of stream placement** on accuracy, parameters and interpretability metrics for ResNet-50 on ImageNet. $S_\ell - S_L$: CA-Stream runs from stage ℓ to L (last); #PARAM: parameters of CA-Stream only. 89
3.6 **Effect of class agnostic *vs.* class specific representation** on accuracy, parameters and interpretability metrics of CA-Stream for ResNet-50 and different interpretability methods on ImageNet. #PARAM: parameters of CA-Stream only. 89

4.1 **Cosine Regularization Experiments:** on CIFAR-100 with ResNet-18 and MobileNet-V2. Accuracy and interpretability metrics are reported. 98
4.2 **Regularization selection:** Evaluation of the four proposed regularization with ResNet-18 on CIFAR-100. 99
4.3 **Regularization coefficient:** Evaluation of the regularization coefficient λ, using ResNet-18 with *Cosine Similarity* on CIFAR-100. 99

Liste des acronymes

AD

Average Drop. 52–54, 61, 64, 65, 67, 75

ADCC

Average Drop in Coherency and Complexity. 53

AG

Average Gain. 61, 64, 65

AI

Average Increase. 52, 54, 61, 64, 65, 67, 75

CAM

Class Activation Maps. 46, 58, 61, 103, 104

CNN

Convolutional Neural Network. 24, 25, 28–33, 35–38, 40, 41, 77, 100

D

Deletion. 54, 65

GAP

Global Average Pooling. 30, 31

GFLOPS

Giga Floating Point Operations Per Second. 40

GPU

Graphic Processing Unit. 29

HOG

Histograms of Oriented Gradients. 24, 27

I

Insertion. 54, 65

ILSVRC

ImageNet Large Scale Visual Recognition Challenge. 29

k-NN

k-Nearest Neighbors. 27, 28

MHSA

Multi Head Self-Attention. 33, 34

NASNet

Neural Architecture Search Network. 32

NLP

Natural Language Processing. 27, 33, 35, 44, 106

NN

Neural Network. 28

SVM

Support Vector Machine. 24, 27

ViT

Vision Transformer. 35–37

Introduction

Human curiosity has led to the desire of understanding the world we inhabit, prompting us to seek explanations for the phenomena we encounter. We derive this from the information we gather through sensory processing. Since most of the sensory information humans process pertains to vision, it could be argued that we live within a visual world. Conversely, this human curiosity has led to the development of technologies that have fundamentally altered the world: we have drastically changed our surroundings by building and adapting them to our needs. Moreover, societal development has been closely intertwined with technology: on one hand, the first human settlements date back to the surge of agriculture; while on the other hand, the industrial revolution started paving the world towards the modern era.

Computation, Computer Vision and Artificial Intelligence Currently, one technology that has taken prominence is computation, as it affects our lives directly and indirectly. This can be seen in our reliance on devices such as computers and cellphones. These products are the result of scientific breakthroughs and innovation. Nevertheless, to do science we need to process information, for which we have developed disciplines like mathematics and physics, which in turn can be aided with computation. Conversely, innovation within the last century has propelled computation further with the emergence of electronic computers. Aided by improvements in transistors and the rapid development of microprocessors, computers have become faster, smaller and more accessible; allowing for their adoption within society. In recent years, this technology has undergone a revolution with the surge and popularization of *Artificial Intelligence* (AI), a promising field with countless possibilities for changing and improving human lives.

Artificial Intelligence refers to a discipline in computer science, aimed at developing systems capable of performing tasks, usually achieved with human intelligence. For example, AI systems learn from data: they recognize patterns and make decisions based on the data itself. Moreover, AI benefits from techniques such as Machine Learning, Deep Learning, Natural Language Processing and Computer Vision. However, *what do we mean when we say a system learns?* In techniques such as Machine Learning, the goal is to develop models that given a certain collection of data, answer a specific task. Consequently, these models learn by updating their parameters based on the hidden structure of data and its statistics.

One particular field where Artificial Intelligence displays promise is *Computer Vision.* Computer vision aims to replicate human vision capabilities with a machine. This endeavor can be understood alongside three axes: *Recognition*, *Reorganization* and *Reconstruction* (Malik et al. 2016); the three fundamental tasks of this discipline. Through recognition, we identify and assign semantic values to elements in our environment. Conversely, on regrouping we organize elements in space, according to their characteristics or concepts. Finally, through reconstruction we identify elements in a scene, producing a model of the external world. In computer vision, we employ models to approximate these axes. Moreover, with the adoption of AI in computer vision, the capabilities of these models to emulate human vision have increased drastically, leading to their adoption in tasks such as recognition of individuals, processing of mail and medical diagnosis. In this thesis we take special interest in the task of image recognition. On one hand, it is the dimension that is the easiest to understand. On another hand, given its simplicity, it is used to prototype and produce methodologies intersecting the complementary dimensions of computer vision study.

Computer Vision is one major field that modern AI has impacted greatly, shown by the progress it has seen in the last decade. In particular, with Convolutional Neural Networks (CNNs), a breakthrough on image recognition occurred. Efficient computation of this operation enabled its usage on larger collections of data, allowing computer vision models to improve their capabilities. Furthermore, these models have benefitted from constant development, in turn allowing to perform more complex tasks over time. On one hand, this has led to the creation of technologies robust enough to build autonomous vehicles. On the other hand, complex tasks such as medical diagnosis are receiving AI tools to facilitate them. More recently, another breakthrough has taken place with the introduction of the transformer architecture. In particular, this architecture allows for a high degree of abstraction, successively avoiding issues that convolutions face, such as inductive bias and difficulties towards generalization. As a result of this, transformers have overcome convolutions in terms of performance but complexity as well. Consequently, its no longer so much a question whether *can a model achieve a given task?*, but rather a question on *how can this model perform this task?*. The main issue regarding these questions lies within the size and complexity of deep models, where providing interpretable explanations has lead to the surge of a novel field of research (O. Li et al. 2018, Guidotti et al. 2018, Bodria et al. 2021), interpretability and explainable AI.

Explainable AI Following the permeation of intelligent vision systems into society and their direct impact into human lives; understanding their inner-working and limitations has become critical. In particular, since their complexity has increased alongside their performance, we are interested in unfolding this property in order to answer questions regarding their outputs; specially when failure cases can negatively affect a life. For instance, considering the medical practice and the involvement of

AI in automated diagnosis, a misdiagnosis of a pathology can potentially derive in an incorrect treatment and loss of life. This is originally referred to, as the *Desiredata of Interpretability Research*, first proposed in *The Mythos of Model Interpretability* (Lipton 2018). Furthermore, in this work Lipton establishes the different properties a model should present in order to be considered interpretable: Transparency and Post-Hoc Interpretability. Current Interpretability research is grouped alongside these two properties.

Lipton suggests that a transparent model is one that can be summarized or explained in its entirely using few words or operations. However, due to the complexity displayed by current computer vision models, providing such interpretations for a model is challenging; in particular, most AI models nowadays contain parameters often counted in millions if not billions. Additionally, the computation of the sequence of operations requiring these parameters to produce an output is also complex in the sense that a forward pass often requires 10^9 operations (OpenAI 2018). On a more active manner, *Transparency* can be attained by the introduction of modifications to a model or its training procedure (Y. Zhang et al. 2021). Several works achieve this with the introduction of small decision trees to summarize the forward pass of a model; as well as with the addition of regularization terms during training encouraging elements of the model to represent semantic concepts (Bau et al. 2017, Wu, Hughes, et al. 2018). We expand upon this on subsection 1.3.1

Regarding post-hoc interpretability, Lipton suggests to leverage upon the complex structure of models and consequently provide explanations utilizing the already existing parameters within the network. This approach in turn allows for a large variance in methodologies since information can be extracted in a plethora of different manners in current CNNs and transformers. Moreover, this variance of explanations can be observed in the nature of the explanation itself: post-hoc interpretations are often presented via text as captions, and in images often using saliency maps, to name a few (Bach et al. 2015, Ribeiro et al. 2016 B. Zhou, Khosla, et al. 2016). On subsection 1.3.2 we explore these methods in more detail. Nevertheless, since explanations are computed to highlight relevant information describing the inference process of a model, they are not aligned to what a human would consider following the same questioning. For instance, an individual might identify the whiskers and ears of a cat as its defining characteristic; but a model can conversely highlight the snoot or eyes instead. On top of this, post-hoc interpretations can be obtained for any class a model consider: be it the correct one pertaining to an object of interest or one completely unrelated. Still, on practice researchers tend to focus on the first case mentioned, while instances where a model fails to provide a correct prediction should the ones where interest should be focused on.

Thesis objectives This PhD thesis aims at studying image recognition models and building upon them to propose novel model interpretability approaches. In particular, we aim at improving both recognition and interpretability capabilities of model predictions. While existing approaches may present these properties, some limitations still remain: a high computational cost, a lack of consensus regarding evaluation procedures, and a disconnect between human interpretability and model interpretability.

Regarding the aforementioned computational cost, we can approximate this issue following Lipton's interpretable properties. On one hand, *transparency* approaches often require training of additions to the network or of the network itself. We observe that this computational cost comes from these alterations: in order to explain the model, its performance should be maintained or not worsened. As a result of this, the model parameters are taken into consideration during this phase, although they are not modified. Moreover, an optimized training procedure can be introduced to achieve this; as well as the modifications themselves can be simple but meaningful. On the other hand, complexity on *post-hoc interpretability* approaches presents some variance. In particular, since they are built on the already existing parameters and computational graph of a model; their combination building up to an approach produces this variability. Moreover, we highlight that simple approaches often require direct computations within the model: a forward pass and a backward pass. Nevertheless, more complex approaches often involve several forward passes in order to generate a representation. In this aspect, we argue that these complex methodologies could be further simplified or made sparse in order to reduce their cost. To answer this requirement, in Chapter 4 and Chapter 3 we propose methodologies requiring one training procedure per dataset, and presenting lightweight inferences. Additionally, on Chapter 2, we propose a saliency map method that although a bit more expensive than traditional approaches, displays state-of-the-art interpretability properties.

On the topic of *consensus of evaluation*, it has been observed that with the release of novel interpretability approaches, comparisons are not consistent between articles. For instance, we observe inconsistencies in the measurements of Grad-CAM++ (Chattopadhay et al. 2018) when compared to its values reported on the article of Score-CAM (H. Wang, Du, et al. 2019). However, this is not the only case as this is repeated alongside multiple studies (K. H. Lee et al. 2021,H. Wang, Naidu, et al. 2020, Naidu et al. 2020). Because of this, a direct comparison across several methodologies does not exist. Therefore, there is no clarity regarding the true performance of approaches. To address this, a benchmark using a standardized evaluation procedure should be performed to clear these shortcomings. In this thesis we conduct standardized experimentation, experiments with equal evaluation objectives follow the same procedure as observed in Chapter 2 and Chapter 3

Furthermore, explainable methods do not suffer only from lack of consensus of standardization. In particular different families of approaches evaluate different objectives.

On one hand, some post-hoc interpretability methods measure the effects of prediction probability, by considering the product of a saliency map with an input image. On the other hand, other methodologies assess interpretability by measuring the accuracy of the explanations they provide instead. In this aspect, we argue that a standardized procedure should present have clearly defined objectives: measure the impact of explanations in prediction probability, and assess their recognition properties. We address this challenge setting our experimentation objectives clear: in Chapter 2 we evaluate recognition and localization properties of Opti-CAM, in Chapter 3 our Cross Attention Stream is subjected interpretable to object recognition evaluation, as in Chapter 4 as well.

Finally, on *alignment between human and model interpretations*, we observe that based on the learning procedures both actors perceive, interpretations differ. On one hand, human learning is not standard; associations of concepts may differ according to societal factors, as well as biochemical ones: important factors describing a prediction differ between individuals. On the other hand, although semantic concepts can be similarly highlighted by different models; these attributions share similarities in how they are addressed to what the model deems important. Taking into consideration this remark, we argue that human-centric interpretability approaches should be made explicitly different to model-centric ones. On top of this, model interpretability claims should be sustained with quantitative evaluation procedures. In this work, we present interpretability proposals aligned to model interpretability, and we validate each of our claims using quantitative methods.

Dissertation Outline This dissertation is aimed towards the development of interpretable image recognition models and is organized in the following manner: In Chapter 1 we introduce a background for image recognition models (Section 1.2) and the ensuing approaches developed to study the interpretability on them (Section 1.3). Additionally, we introduce evaluation procedures for these approaches which will be further used to evaluate our proposals.

In Chapter 2, we propose Opti-CAM as a methodology that generates optimized saliency maps highlighting the relevant regions on an image towards image classification. In Section 2.4 we extend existing evaluation metrics with a novel measurement for model confidence. On Sections 2.6 and 2.7 we evaluate the effect of these contributions towards interpretability assessment. Opti-CAM overall presents an approach that highlights saliency relating to the classifier, thus the saliency map generated performs the best in terms of interpretability metrics although is not highly aligned to human interpretations. On top of this, our novel metric *Average Gain* complements current interpretability evaluation metrics, quantifying benefits in prediction confidence using saliency maps. We follow this procedure to further evaluate our proposals.

Chapter 3 introduces the Cross Attention Stream, an approach that boosts existing architectures interpretable properties. We set up the modulus of this approach in Section 3.2 alongside its deployment on Section 3.3. In Sections 3.5 and 3.6 we demonstrate the benefits of using this proposal. In particular, our CA stream is a transparency approach that is evaluated through post-hoc interpretability evaluation methods. Moreover, our approach learns an abstract representation of the predicted class by the model; enhancing prediction probability, and improving interpretability properties.

Chapter 4 characterizes a gradient denoising approach with a gradient denoising methodology as an approach to enhance the training procedure of current models while improving interpretability properties. In Section 4.2, we define the gradient denoising protocol alongside the regularization proposals to do so. Section 4.3 illustrates the effects of this paradigm in the trained models and its effects on interpretability. This approach provides a preliminary study on the introduction of a regularization term during model training to improve upon both recognition and interpretability properties. Furthermore, this approach is mostly exploratory and still requires further developments to be consequently employed in larger collections of data.

Finally, we draw conclusions on our work and detail future research perspectives.

1 Background

Table of contents

1.1 Introduction . . . 23
1.2 Image Recognition Models . . . 26
1.2.1 Traditional Image Recognition models . . . 27
1.2.2 Convolutional Neural Networks . . . 28
1.2.3 Self-Attention Based Architectures . . . 33
1.2.4 Hybrid Architectures . . . 36
1.2.5 Discussion . . . 38
1.3 Interpretability . . . 39
1.3.1 Transparency . . . 41
1.3.2 Post-Hoc Interpretability . . . 44
1.3.3 Evaluating Interpretability . . . 51
1.3.3.1 Trust and Reliability . . . 52
1.3.3.2 Interpretable Object Recognition . . . 52
1.3.3.3 Causal Analysis . . . 54
1.3.3.4 Interpretable object localization . . . 55
1.3.4 Discussion . . . 57

1.1 Introduction

Understanding the processes behind visual recognition has been a prominent research question throughout human history. From the preliminary questionings by greek philosophers (Finger 2001) to physics based studies like those by Newton and Locke (Swenson 2010), and more recently with theories like *Unconscious Inference* (Gullstrand 1909) and *Gestalt* (Wagemans et al. 2012), many proposals to understand and describe this process have been brought forth. In modern times, vision has been studied on a medical level following neuroscience. To illustrate, in order to understand responses to stimuli, areas of the brain such as the striate cortex have been subject to inquiry (Hubel et al. 1959). Moreover, vision recognition is not only studied in fields such as physics, medicine and psychology; with advancements on computer science, computational approaches and theories started emerging regarding this domain. One such study that proved seminal in this domain is that of David Marr (Poggio 1981, Marr 2010). Most notably, Marr addressed vision on three levels: computational, algorithmic and implementation. In particular, upon the computational level, Marr pondered around issues that the visual system answers and their explanation; this ultimately led to the formulation of fundamental tasks within computer vision such as object recognition and reconstruction.

Following Marr's proposals and the ensuing research on computer vision, researchers centered their attention at developing methodologies towards performing these fundamental tasks. Starting with preliminary works on reconstruction of 3D objects in space, the development of computer vision models then followed specialized approaches for specific tasks. In this thesis, we are interested on models designed at performing image recognition and, more specially, in understanding their functioning and providing explanations for their inner workings.

Image recognition models capabilities have improved over time with constant development, closely aligned with the increase in computing power. This evolution has lead to steady advancements in both performance and complexity. On its early approaches, image recognition models relied on handcrafted feature extraction methods in conjunction with traditional machine learning algorithms. However, this reliance on these features ultimately limits these methodologies capabilities to capture intricate visual patterns. One particular approach that had a strong initial impact and high performance was Histograms of Oriented Gradients (HOG) (Dalal et al. 2005). In this methodology, gradient information is used to train a Support Vector Machine (SVM) to perform pedestrian recognition. While achieving high performance in the dataset it is designed, HOG ultimately fails in data collections where complexity is higher (Dollar et al. 2012).

It is not only with the increase of computational power that computer vision has improved over time. With the development, popularization and spread of the internet; large collections of data are formed. These aggregations can be extremely specific for a given end, or quite general representing the common interests of its users. These compilations have continued to grow both in volume and variety. Still, several curated collections are introduced by researchers to experiment and control the development of models such as MNIST (LeCun et al. 1998), BSDS (Martin et al. 2001), Pascal VOC (Everingham, Van Gool, et al. n.d.) and most notably, ImageNet (Russakovsky et al. 2015) and MS-COCO (T.-Y. Lin, Maire, et al. 2014).

Furthermore, with the resurgence of convolutional neural networks and the ensuing advent of deep learning, a paradigm shift in this field occurred. This transition led to the elimination of the need for handcrafted feature extraction; instead, deep learning allowed Convolutional Neural Networks (CNNs) to act as both feature extractor and classifiers on themselves. Moreover, based on the aggregation of convolutions as their fundamental units, a CNN is able to learn hierarchical representations of data, extracting intricate features directly. Consequently, this shift resulted in improvements in accuracy and performance in various image recognition tasks.

However, it was observed that performance of CNNs was achieving a plateau, and the introduction of novel architectures stagnated over time. Additionally, these models encountered challenges capturing long range dependencies, in turn limiting their capacity to construct global representations of data and affecting their generalization capabilities. Transformers however, have shown remarkable improvements not only in the image domain but also in language related tasks, revolutionizing these fields and paving the groundwork for future research and developments. Nevertheless, one particular CNN that has stood the test of time is ResNet (He, X. Zhang, et al. 2016), a model that incorporates residual connections between layers, allowing training in complex tasks and datasets

In section 1.2 we explore some of the most important approaches based on machine learning designed towards image recognition. In particular, in subsection 1.2.2, we introduce the basis of deep learning and CNNs. Moving on, in subsection 1.2.3 we make mention of the Transformer architecture, its building block and how it has reshaped the landscape of image recognition. Finally, in subsection 1.2.4 we display approaches that make use of combinations of the last two forementioned approaches.

With the adoption of deep image recognition into society; understanding the inner workings of these models has become a top priority. We shine light into some fields where this is the case:

- **Facial Recognition** This is a fine-grained classification task, that can be mostly associated with identification and re-identification of individuals. In this aspect, understanding model predictions is associated with accountability, ethical considerations and safety. (Selinger et al. 2020, Andrejevic et al. 2020).

- **Automated Medical Diagnosis** In medical imaging, the education required to read and provide analysis, often requires experience based on personal expertise (Nakashima et al. 2013). Examples of automated diagnosis encompass melanoma detection, bone age assessment and most recently, COVID-19 diagnosis (Yu et al. 2016, Escobar et al. 2019, S. Huang et al. 2021). The medical domain is of special care as human lives are directly at stake, therefore understanding predictions is highly desired.

- **Self-driving Vehicles** Over the past decade, advancements in this field have lead to discussions regarding the impact of the adoption of these sorts of vehicles within smart cities (Duarte et al. 2018, Millard-Ball 2018). Nevertheless, their navigation is not completely perfect, and it is also possible to attack it, leading to possible traffic accidents (Dixit et al. 2016); in this case, accountability is then again taken into consideration.

We observe that interpretability needs in these fields and real world applications follows Lipton's discussion on the *Desiderata of Interpretability Research* (Lipton 2018). In this aspect, we expect that the explanations that any given approach help us *trust* any model. Conversely, we expect a model to be explained in a *causal* manner according to its explanations. In particular, we expect these remarks be *informative* and shine light on similar examples that the model processes. Finally, we expect interpretable explanations to guarantee that outputs of a model are *fair* and *ethical*.

As a response to these requirements, the AI act was recently proposed in Europe. In this document, a set of rules was established such that individual and business safety rights are respected when it comes to AI (Madiega 2021). Specifically, this act defines these rules based on a hierarchy of risks posed to society; we present this hierarchy in Figure 1.1. With the acceptance of these rules, it is intended to address and limit risks created by AI applications.

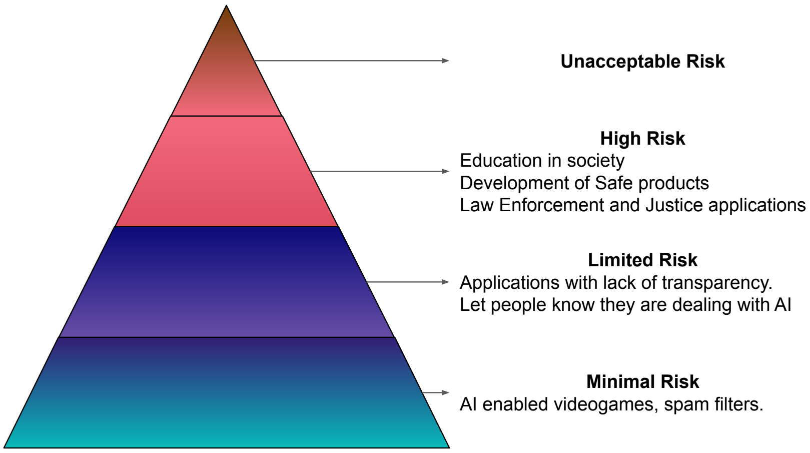

Unacceptable Risk
High Risk
Education in society
Development of Safe products
Law Enforcement and Justice applications
Limited Risk
Applications with lack of transparency.
Let people know they are dealing with AI
Minimal Risk
AI enabled videogames, spam filters.

Figure 1.1: **AI act pyramid of risk levels**. Adapted from `https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai`

With these considerations leading the *desiredata of interpretability study*, we further investigate some of the most important works on interpretability. In section 1.3 we explore preliminary ideas of this field. Delving deeper, in subsection 1.3.1 we discuss efforts aimed at transparency of machine learning approaches. In contrast, in subsection 1.3.2 we explore and study some of the most relevant studies on post-hoc interpretability. Conversely, in subsection 1.3.3, we outline the evaluation metrics used for evaluation the aforementioned works. To understand how these proposals are evaluated in their claims of interpretability, we introduce and explain evaluation methodologies in subsection 1.3.3.

1.2 Image Recognition Models

Image recognition is a subtask of computer science that aims at replicating human vision capabilities with a machine. From its early developments with David Marr's postulates on vision (Poggio 1981, Marr 2010), this field has been approached on a computational level since the 1980s. Following this, early works in computer vision paved the way for the application of classical machine learning models on image recognition. In the beginning, researchers focused on fundamental challenges such as edge detection and image segmentation. The 1970s and 1980s witnessed pioneering efforts, with techniques like the Hough Transform for line detection and the development of feature-based methods (Duda et al. 1972). These early approaches set the groundwork for the later integration of classical machine learning models, as they provided preliminary insight into how visual information could be analyzed and processed computationally. The emergence of classical machine learning models, and their application in the 1980s, marked a shift towards more sophisticated image recognition methodologies.

1.2.1 Traditional Image Recognition models

Traditional image recognition approaches based on traditional machine learning algorithms are a two-step process. The first step involves the extraction of features from the data, and the last step lies in the training of a classifier. Examples of this can be found in methodologies such as the Scale Invariant Feature Transform (SIFT) (Lowe 1999), Bag of Words (BOW) (Csurka et al. 2004), and HOG. SIFT, stands out as a powerful keypoint detector that incorporates image alignment. More importantly, its robustness towards scale and rotation led to widespread adoption in real world applications (Cruz-Mota et al. 2012). Complementary to SIFT and adopting ideas based on Natural Language Processing (NLP), the concept behind the BOW descriptor originated from the identification of keywords on a text, in order to identify its contents (Harris 1954). On itself BOW extracts features akin to SIFT, clustering them to generate a dictionary which will ultimately form the possible words to describe images. Each image is recognized by the frequency of which certain words are used to describe it. Finally, HOG follows a simpler approach, where image gradient information is organized in histograms describing the orientation of image components.

Once image descriptors are extracted, classifiers are trained using algorithms such as SVM (Cortes et al. 1995), Random Forests (Ho 1995) and k-Nearest Neighbors (k-NN) (Cover et al. 1967, Fix et al. 1989). SVMs are classifiers that operate by finding the most optimal hyperplane in the feature space to discriminate between classes. However, challenges arise when the assumption of data being linearly separable is not met. This difficulty was addressed with the introduction of the kernel trick (Hofmann et al. 2008), where data is transformed to another space where this is more straightforward.
Complementary to SVMs, Random Forest Trees is a methodology that constructs a plethora of decision trees, where each tree uses a random subset of the training data, and each node uses a random subset of features to make a decision. This randomness ensures diversity among trees, providing a degree of robustness towards overfitting.

On a much simpler note, k-NN operates based on the assumption of data being contained in k different categories. The classification method involves assigning one category to a data point that is the closest in feature space to a class centroid, obtained through K-Means (MacQueen et al. 1967).

Although many of the early image recognition models were based on traditional machine and statistic methods, neuroscience research still inspired scholars to propose alternative approaches. Moreover, with studies on the visual cortex regarding receptive fields (Hubel et al. 1959), the development of Neural Networks (NNs) started. Notably, the Neocognitron (Fukushima 1975) sparked the inception of Ns, introducing kernel operations, hierarchical feature aggregation and non-linearities. These contributions stand out as they are key components in most recent image recognition models. Still, the hierarchical properties and aggregation of the Neocognitron did not really achieve a great deal of momentum on early days; around this time, other image recognition models were being used yielding better results, examples of this can be seen with the amount of traditional machine learning based methods that dominated this task around that time. Nevertheless, getting close to the dawn of the year 2000, Yann LeCun proposed LeNet to perform digit recognition (LeCun et al. 1998) the first modern CNN.

1.2.2 Convolutional Neural Networks

Starting with LeNet, the convolution took prominence as the fundamental building block of most current image recognition models. In the domain of computer vision, the convolution is an interaction $(f \star g)$ between a feature map (f), and a kernel (g), as shown in Figure 1.2. In particular, the convolutional kernel g is mediated by its area determined by width and height, influencing directly its receptive field. The receptive field answers to the area within the input space covered by the convolutional kernel. Furthermore, during convolution, the kernel slides over the feature map, computing the dot product between the kernel g, over the area it covers in the input map centered around each pixel. Consequently, in deep layers of a CNN this computation encompasses larger regions of the input image, allowing the model to capture long range dependencies and enhance its capabilities. Additionally, convolutions present similarities to SIFT, such as their ability to construct representations invariant to image alterations, and utilize receptive fields for feature extraction.

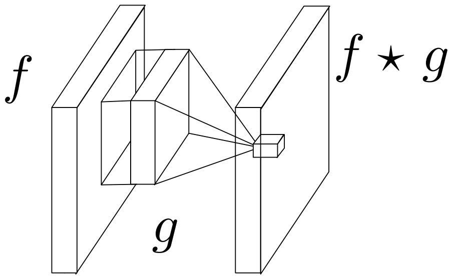


Figure 1.2: **Illustration of the convolution operation.** A kernel g operates over feature map f generating an updated representation $f \star g$ (M. Lin et al. 2013)

On top of convolutions, LeNet led to the introduction of more components of CNNs: pooling operations and non-linearities. Pooling operations are used to reduce the spatial resolution of feature maps, which in turn aids convolutions in capturing features from long range dependencies within the image. Moreover, via pooling it is possible to capture the most relevant features within a neighborhood. Conversely, non-linearities such as Sigmoid and ReLU (Fukushima 1969), are designed to capture complex relationships within data, stopping the model collapsing into a linear operation. Furthermore, these operations also contribute with stability: they maintain values within feature maps and the gradient in ranges in which the network can operate with. Additionally, it is possible to control the flow of information within the network with operations such as Dropout. This operation randomly deactivates units on a convolutional layer, forcing the model to learn more robust representations as it cannot rely on a set of previously learned features consistently. Still, the key contribution leading to the success of LeNet was not only the usage of convolutions, pooling and non-linearities; but its training process, that guided by gradient descent to optimize, ultimately enabled CNNs to outperform traditional computer vision methods for document recognition.

With the advent of the 2010s and the initiation of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al. 2015) a proper environment for further development of models was established. Unlike prior datasets, ImageNet was composed of images that presented more complexity than earlier datasets. Instead of catalog-like compositions; elements in this dataset closely resemble those found in the wild, featuring multiple classes or instances of a class within a single image.

Upon its release, several traditional approaches were trained and evaluated on this collection, achieving a low performance. However, CNNs regained prominence with the introduction of AlexNet (Krizhevsky et al. 2012). Inspired by LeNet, Krizhevsky designed a CNN that incorporated additional convolutions and, more importantly, facilitated faster computation through effective communication with the Graphic Processing Unit (GPU). AlexNet gained notoriety by emerging as the winner of the 2012 ILSVRC, achieving a top-1 classification accuracy difference of nearly 10% compared to previous year winners (Berg et al. 2010, Sánchez et al. 2011). This substantial improvement in recognition capabilities led to a paradigm shift in various machine learning tasks, laying the foundation for the deep learning revolution. Following this success, several CNNs were introduced in the decade of 2010, and since then; a considerable amount of models have been proposed. Nevertheless, we can establish a timeline with the milestone models that influenced the most this development, as seen in Figure 1.3 and iterate upon the performance of some of these models in Table 1.1.

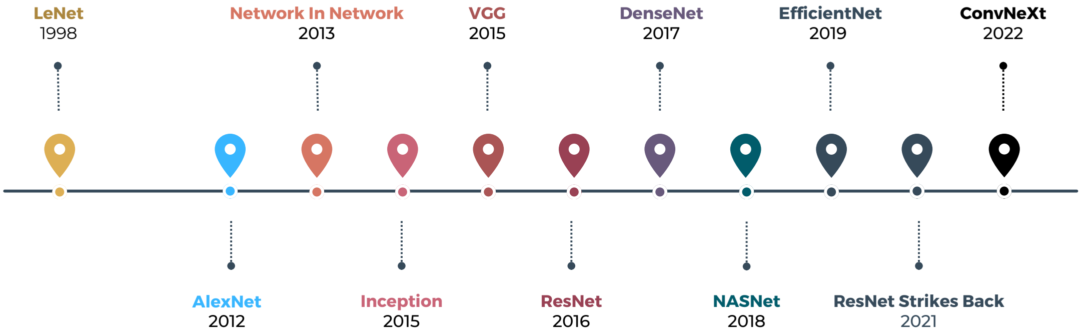


Figure 1.3: **Timeline** of milestone Deep Learning models.

Method	Release Year	Acc@1	Acc@5	Params	GFLOPS
AlexNet	2012	51.52	79.07	61.1M	0.71
VGG-16	2015	73.36	91.52	138.4M	15.47
ResNet50	2016	76.13	92.86	25.6M	4.09
EfficientNet-B0	2019	77.69	95.32	5.3M	0.38
ViT-Base	2020	81.07	95.32	86.6M	17.56
ResNet-50*	2021	80.86	95.43	25.6M	4.09
ConvNeXt-Base	2022	84.06	96.87	88.6M	15.36

Table 1.1: **Milestone Image Recognition Methods** Details of milestone Image recognition models in the era of Deep Learning on ImageNet 1k. ResNet* refers to the version using updated training protocols (Wightman et al. 2021)

In the year following the publication of AlexNet, an updated form of mapping feature maps into classification embeddings was proposed in the shape of Global Average Pooling (GAP) (M. Lin et al. 2013). This pooling protocol generates a representation taking the average value of each feature map channel, as shown in Figure 1.4. Furthermore, GAP reduces dimensionality and regularizes the model using global aggregation; in turn improving classification performance.

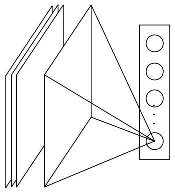
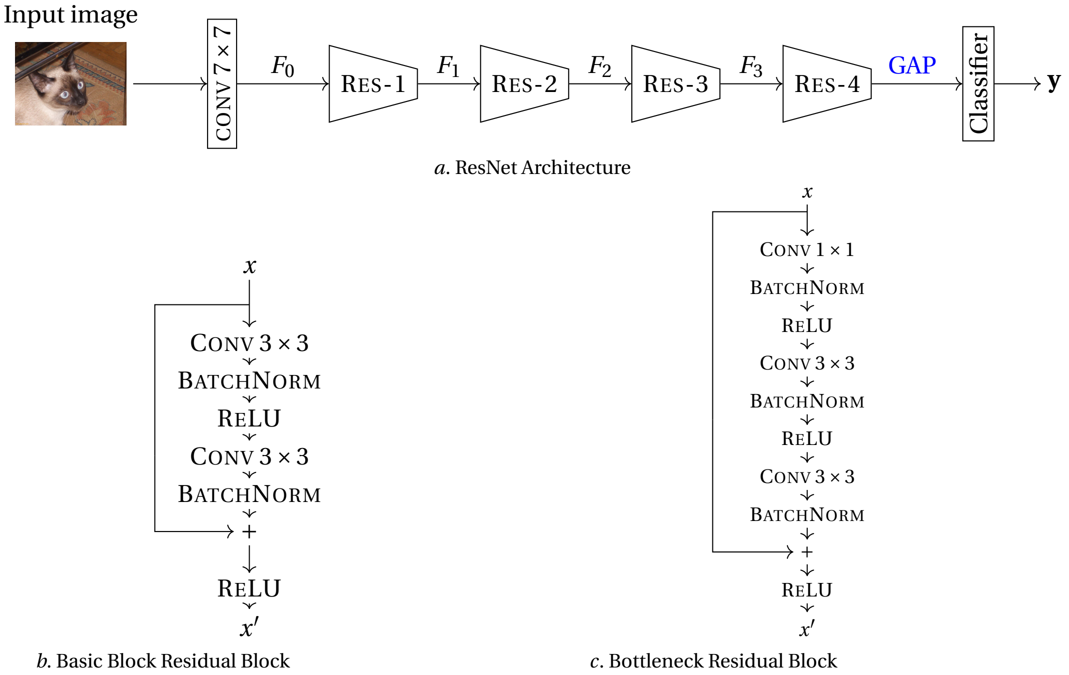


Figure 1.4: **Visual representation** of Global Average Pooling (M. Lin et al. 2013).

Similar to 2012 and 2013, 2015 saw the proposal of two milestone models: the Inception architecture (Szegedy et al. 2015) and VGG models (Simonyan and Zisserman 2015). On one hand, the Inception architecture was designed to learn features in different scale. To achieve this, the *Inception Block* introduced the *Inception Block*, which captures multiscale behavior by incorporation of convolutional kernels with sizes of 5 × 5, 3 × 3 and 1 × 1. On the other hand, VGG models were built with a simplistic design, relying solely on 3 × 3 convolutions. For a change, VGG is shown to be an excellent feature extractor network. Led by the desire to increase depth of CNNs, VGG and Inception attempted to make models deeper; nevertheless this was not possible. As models get deeper, the gradient becomes zero when flowing from deep layers to shallow layers, denying updates to their parameters. This is known as the vanishing gradient issue (Pascanu et al. 2013). To address this issue, the ResNet architecture was proposed (He, X. Zhang, et al. 2016).



a. ResNet Architecture

b. Basic Block Residual Block

c. Bottleneck Residual Block

Figure 1.5: **Generalities** of the ResNet architecture.

The ResNet architecture is designed with the idea of residual connections as its building block. On itself, a residual block generates outputs via the summation of its input and a linear mapping of it. In detail, a residual block takes a feature map and projects it to different dimensions, regularizes it through a bottleneck or with activations, and then sums this representation with the original input. This in turn enhances the network capabilities to scale in size, leading to improvements in performance while being easier to optimize. This architecture maintains its relevancy because of its modularity and the aforementioned scaling properties. For instance, some of the most important CNN based object detectors are designed using ResNet as backbone (Ren et al. 2015, T.-Y. Lin, Goyal, et al. 2017, He, Gkioxari, et al. 2017). A thorough representation of this architecture, as well as the residual connection variants is presented in Figure 1.5.

Similarly to the residual connections introduced in ResNet, DenseNet (G. Huang et al. 2017) was proposed with the idea of connecting all layers operating within matching feature-map sizes. We illustrate this on Figure 1.6. In particular, this architecture enables the training of very deep models, as these connections facilitate feature reuse and identity mappings, thus negating the effect of vanishing gradients. However, one issue of DenseNet is its lack of modularity and ease of use. Due to a great number of neurons being interconnected by design, introducing of modifications such as Non Local Blocks (X. Wang et al. 2018) or Squeeze Excitation Blocks (Hu et al. 2018) is challenging. In contrast, in architectures such as ResNet this procedure is simple.

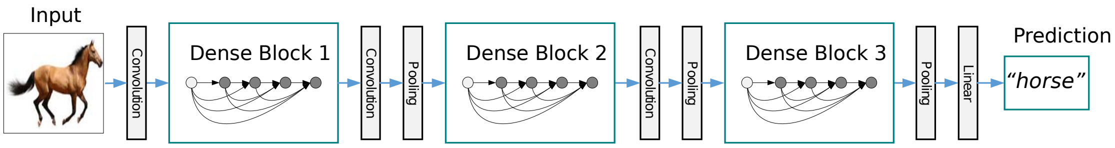


Figure 1.6: **Illustration of the DenseNet architecture** (G. Huang et al. 2017)

Moreover, it is arguable that convolutional block design can be improved if a model learns on itself the best configuration possible for a specific task. This is idea is embodied by the Neural Architecture Search Network (NASNet) (Zoph et al. 2018) where the model learns a fundamental building block on a small dataset, and it is then transferred into a larger one. Some of the biggest drawbacks for this model, are its computational load and the dependency on search space selection towards optimization. One last milestone attempt at improved architecture design was proposed in 2019 with EfficientNet (Tan et al. 2019). In sharp contrast to its contemporaries approaches at scaling; EfficientNet proposes a *compound coefficient* instead.

Still, some of the aforementioned architectures were not proposed with such contributions, and as such it is possible to suggest that an unfair comparison is performed, as well as an incomplete study on said model capabilities. One such answer to this issue was proposed for ResNet in 2021 (Wightman et al. 2021). In this approach, ResNet was retrained under updated training regimes, achieving a high classification performance,

rivaling that of transformers.
Finally, with the advent of transformer based image recognition models in the early years of the 2020 decade, CNNs started being outshone by these models; however, an architecture incorporating the key principles of these models was presented, ConvNeXt (Zhuang Liu et al. 2022). This family of models addressed some shortcomings of transformer models (mentioned in subsection 1.2.3) and proposed their mitigation with the modernization of a ResNet architecture, enhancing its performance not only in image recognition, but also in segmentation and detection. Still, having mentioned transformers, we dedicate the following subsection to their description, their basic unit and some milestone models.

1.2.3 Self-Attention Based Architectures

One key point to remember is that Computer Vision is not isolated within Artificial Intelligence; advancements in this field have, conversely, contributed to complementary domains, such as NLP. Furthermore, proposals made for that domain have found applications into image recognition; one such development is that of Transformers. The Transformer architecture was initially proposed in 2017 with the article *Attention is All You Need* (Vaswani et al. 2017). This model addressed limitations in existing methodologies for NLP such as LSTMs and RNNs, particularly their struggles in capturing long range dependencies and efficient training.

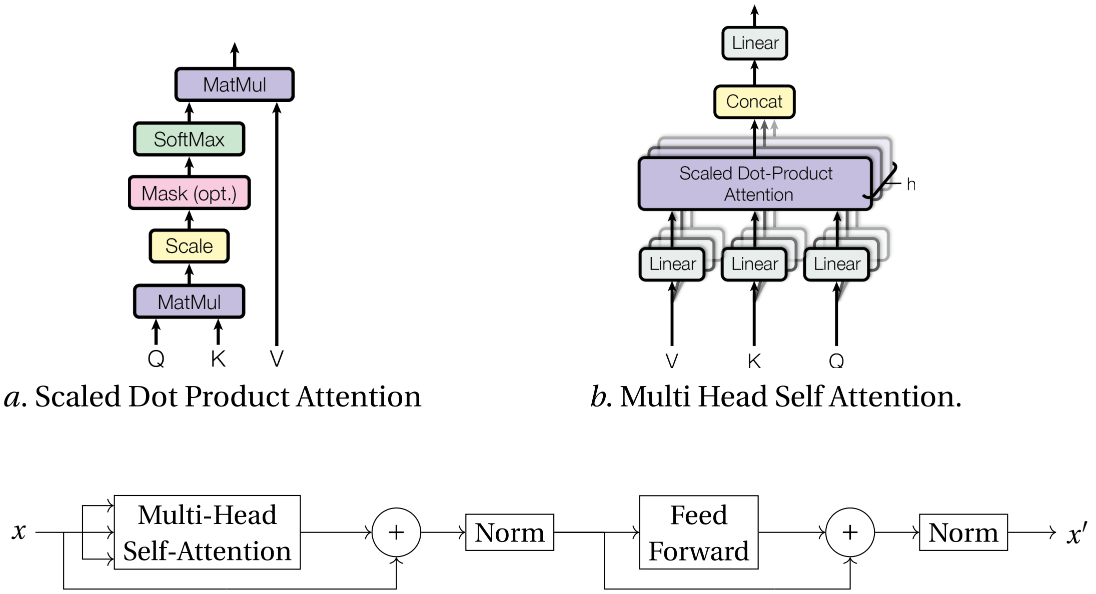
Similarly to the impact AlexNet had on image recognition, transformers revolutionized the landscape of NLP with their key component: *Self-Attention*. This function assigns weights to different input sequences, enabling focus on relevant information. This is defined in the following manner:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1.1}$$

where Q,K,V are embedding matrices that represent *queries*, *keys* and *values*, each with a dimension d_k. The softmax activation function is employed to emphasize relevant information for each product between Q and V. Conversely, the scaling coefficient $\sqrt{d_k}$ uses the square root to prevent softmax from entering regions with small gradients, particularly important as large values of d_k produce this outcome. Conversely, for each token *self-attention* is then defined as an average of all values (rows of V), weighted by the attention (corresponding row of A).

$$\textsc{sa}(X_\ell) := AV \in \Re^{t_\ell \times d_\ell}. \tag{1.2}$$

This function can be further parallelized by broadcasting these embeddings into different heads, where the dot products are ultimately easier to compute; this in turn is called Multi Head Self-Attention (MHSA). Self attention has no parameters what-



a. Scaled Dot Product Attention

b. Multi Head Self Attention.

c. Transformer encoder block.

Figure 1.7: **Transformer** self attention variants and encoder block.

soever: this function is utilized in blocks that ultimately form the encoder part of the transformer. One encoder block comprises a residual operation where the input embedding is updated with the product of MHSA, normalized, and then this representation is further recombined with the residual operation atop of a feed forward network. This is best explained in Figure 1.7.

Another pivotal characteristic setting transformers apart from previous approaches on NLP and convolutions is its capability to process data in a global context. In sharp contrast to convolutions discussed in subsection 1.2.2, the self-attention operation is not constrained by parameters like width and height: its receptive field is the whole input. Instead, this operation is guided the number of embeddings that will represent the whole of data (*i.e.* the number of different chunks a sentence is split into) and the aforementioned d_k, which represents the dimensions to which the data is projected.

Continuing on with transformers within the NLP domain, several additions to this architecture have resulted in further improvements. One notable example is the inclusion of the classification token ([CLS]) as proposed in BERT (Devlin et al. 2018). The rationale behind incorporating this token can be traced back to the previously discussed concept of globality in transformers. On its own, the [CLS] token serves as an abstract representation of a class, collecting information of the embedding as a whole. Guided by the self-attention function, this representation is expected to encapsulate the most pertinent information to describe the input data. As a consequence of this, the CLS token is used for classification purposes.

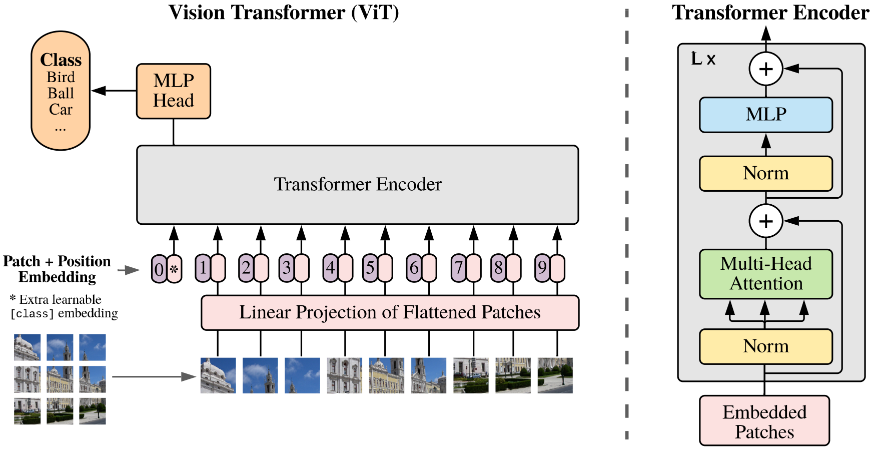


Figure 1.8: **ViT** overview (Dosovitskiy et al. 2020).

The success of transformers on NLP tasks did not go unnoticed by the Computer Vision community, and in late 2020, the first fully attention based architecture Vision Transformer (ViT) was proposed. This architecture emerges as an adaptation of the transformer model for the realm of computer vision. The crucial contribution lies in treating the image as a sequence of patches, processed analogously to a sequence of tokens in a NLP application. Figure 1.8 provides an overview of this approach.

In similar manner to prior breakthroughs like the effective convolution computation and the emergence of transformers in NLP, the introduction of ViT further revolutionized the landscape on Computer Vision tasks. However, these methodologies exhibit specific characteristics concerning their predictive capabilities. In particular, when trained using a conventional approach for ImageNet object classification, the performance of this model is relatively low compared to a ResNet trained under the same conditions. However, when scaling to more modern datasets such as ImageNet-21k and JFT-300M (Sun et al. 2017), which contain close to 10 times the number of images than those of ImageNet-1k, transformers clearly outperform their convolutional counterparts. Interestingly, this is hypothesized to be primarily due to transformer scalability. In particular, transformers are capable of removing inductive bias which in turn is highly dominant in CNNs. Still, this makes these models prone to overfitting in data collections sufficiently small where this bias still remains useful.

In addition to ViT, several other transformer architectures have emerged, contributing to the diversity and advancement of the field. One noteworthy example is PiT (Pooling in Transformer) (Heo et al. 2021), which re-introduces the concept pooling into transformers. To benefit from the hierarchical structure that is found in CNNs, this operation is included with a modification for the pooling operation, where the image tokens are pooled using depth-wise convolution. This inclusion of pooling is sustained by the measurement of *attention entropy*, where this metric shows the spread and concentration of *attention*; where in ViT this interaction is rather spread, and in PiT it is largely concentrated. Another significant development is Swin Transformer (Ze Liu et al. 2021). Swin utilizes a shift-based windowing mechanism, allowing for efficient information processing across different scales. This design promotes enhanced modeling capabilities and facilitates the learning of intricate patterns in images.

Indeed, since its proposal, ViT has dominated image recognition tasks. Still contrary the behavior CNNs present as excellent feature extractors, ViTs underperform in areas such as image reconstruction and image reorganization. In particular, given the locality behavior and the inductive bias that these models present; these aforementioned tasks are consequently carried out in a better manner with CNNs. Nevertheless, similar to CNNs borrowing ideas from transformers as seen in *ConvNeXt*; early stages in transformers dedicated to image patch encodings are replaced with convolutional layers, inheriting these characteristics for further downstream tasks. These models in turn are known as hybrid architectures.

1.2.4 Hybrid Architectures

Following the locality properties of CNNs and globality of ViTs; it is proved that incorporating ideas from these architectures is feasible, generating a model that attends in to both local information and scales well with training data. In particular, some preliminary studies towards these combinations performed on ViT, suggested modifications of the architecture starting with the encoder. Starting with what is arguably the most basic CNN design, LeViT (Graham et al. 2021) replaces the original patch encoding with feature extraction using LeNet; as well as including a distillation head to lead the training process. These contributions ultimately helped LeViT to outperform ViT all the while not requiring large scale image classification datasets to train as well as having a faster inference time.

As a continuation of LeViT, PatchConvNet (Touvron, Cord, El-Nouby, et al. 2021) was subsequently proposed. Building upon the LeViT design, PatchConvNet further expands on the classifier by replacing the original MLP head with *Attention-Based Pooling*. Interestingly, this pooling mechanism integrates a ([CLS]) matrix; that in sharp comparison to the stand-alone token, is able to capture class-specific information. PatchConvNet ultimately improves recognition properties found in LeViT, while also showing promise in terms of segmentation and detection, suggesting that the model has built-in interpretability by the capability of visualizing the aforementioned class-specific attention. Nevertheless, the key contribution behind this architecture is derived from the convolutional stem, taken from LeViT.

One key characteristic that CNNs display is a certain degree of robustness regarding the optimizer choice during the training stage. This phenomenon was observed to be

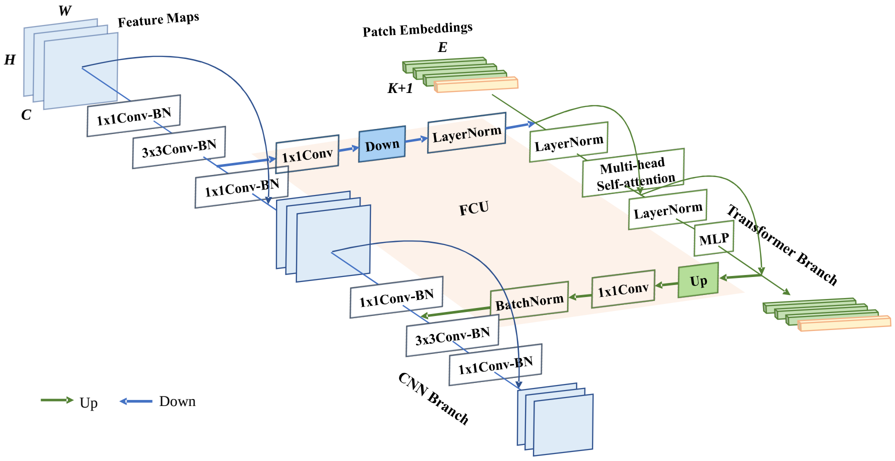


Figure 1.9: **Conformer architecture** overview. (Peng et al. 2021)

the opposite regarding ViT, the optimizer choice is crucial for these models. Similarly to LeViT, by the addition of early convolutions it is argued that this issue can be overcome, as proposed on *Early convolutions help transformers see better* (Xiao et al. 2021). Moreover, in this work a direct comparison between ViT trained using a patchifier, versus using convolutions; where *Optimizability*[1] is used to establish differences.

In addition of including convolution atop ViTs, researchers have sought to propose ideas to combine the differences in processing between transformers and CNNs. In an approach similar to Siamese networks, it is possible to interchange information and update features from one CNN to a transformer by implementing a coupling unit to lead this communication. One approach that encompasses this design idea is the *Conformer* (Peng et al. 2021), where the *Feature Coupling Unit* (FCU) is used to introduce information from a convolutional branch into a transformer one, and in the opposite direction as well. In particular, Conformer uses a ResNet architecture for its convolutional branch, while maintaining the first convolutions to process the image into image patches. For the transformer branch, this model introduces a modified ViT, where the number of encoder blocks is one element less than the totality of residual blocks. Moreover, the aforementioned FCU is introduced to update features in between residual blocks. Ultimately, any prediction is given by the average prediction between both branches. A detailed explanation of this model is found in Figure 1.9.

[1] As defined by Xiao et al. 2021, it refers to the presence of difficulties characterizing the training process of deep models while varying their optimizers, data augmentation and performance drop when scaling in size

1.2.5 Discussion

In computer vision, the development of image recognition models has been crucial towards the advancement of different tasks such as image segmentation and object localization. Supported by the theory of the interaction of Malik's three Rs of computer vision (Malik et al. 2016), Progress of one of these three fields, leads to strides in complementary areas. As described in the previous section, many of the proposals demonstrated therein act as excellent feature extractors. This capacity in turn, facilitates adjacent tasks such as segmentation (regrouping) and reconstruction. With this in mind, we suggest that model development and design account for interactions between the Rs (better seen in Figure 1.10); and consequently, model development being rooted mostly in image recognition. We acknowledge image recognition as one foundational task on computer vision.

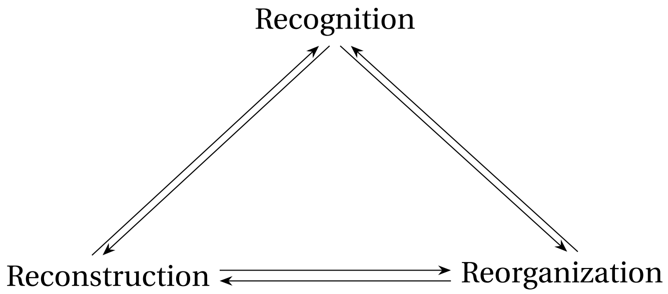


Figure 1.10: **Malik's** three *R* of computer vision

Furthermore, this domain has seen constant evolution in recent years. Following the resurgence of CNNs after the introduction of AlexNet, a plethora of image recognition models were proposed. Still, while these models are variate in structural units, complexity, and depth; the model formulation itself is not the solely determining factor of performance.

Similarly to the points described by *A Metric Learning Reality Check* (Musgrave et al. 2020), where a revision of metric learning methodologies revealed biases in the evaluation of novel methodologies and the enhanced power of previous methods when optimized under better conditions; overall performance evaluation of architectures and methodologies often lacks fair comparison due to advancement in optimization techniques. This is clearly demonstrated in *ResNet Strikes Back* (Wightman et al. 2021). Nevertheless, it is also possible to consider that CNNs may be approaching a plateau in their capabilities, similar to traditional computer vision methods when applied to ImageNet. As a response of this, we take special interest on Transfomers, given their recent adoption and overall their promising capabilities.

Delving into transformers, we remark the promise that they display given recent advances in tasks such as text recognition, text generation and notably in vision, on image generation, captioning and recognition. Comparable to the surge of convolution based methods in the early 2010s, the paradigm shift previously mentioned is already taking place as we can observe on figure Figure 1.11. However, the enhancement of recognition capabilities is highly dependent on the amount and quality of data that is used in their design and optimization. Since recognition capabilities on ImageNet are nearing saturation point for the dataset, we question whether it is truly a feat of model generalization or a severe case of overfitting.

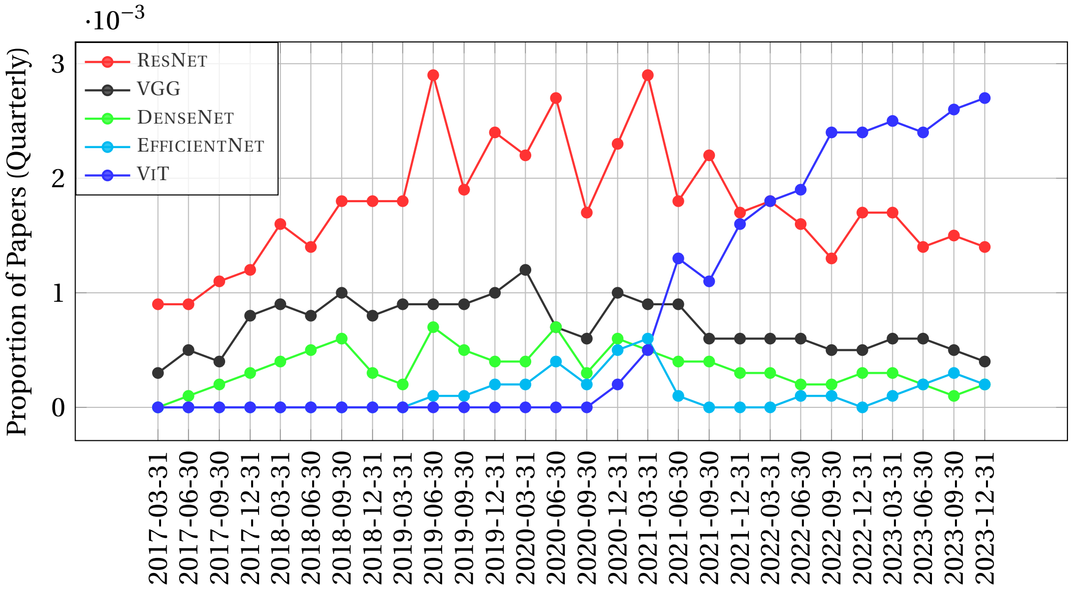


Figure 1.11: **Proportion of articles published on ImageNet,** using image recognition models as backbone across the years. Original from https://paperswithcode.com/method/resnet

1.3 Interpretability

As deep learning based models for Computer Vision have continued to improve in their recognition properties, their structure and functioning have become more opaque; in turn making these technologies seen as black boxes. A black-box model is defined as a model for which its interpretation is not straightforward for humans (Petch et al. 2022). In recent years with the assimilation of deep learning into everyday tasks, and the implicit effect these models are having on human lives; the novel research field of *Interpretability* has been brought forth to open up this black-box behavior. Researchers have approached interpretability alongside different directions. Starting with the work of O. Li et al. 2018, interpretability is suggested to present two categories, *Transparency* and *Post-hoc Interpretability*. For the former, Lipton argues that it follows modifications of the model or the training process, in order to explain the inner-workings of the model. Conversely, for the latter Lipton builds upon the black-box behavior of the model, providing explanations instead based on inputs and outputs; without adding any further modifications for the model or altering its training process.

Considering these properties, as machine learning models grew in complexity; their transparent properties vanished proportionally to their size. It can be argued that traditional models offer themselves to transparency due to their straightforward formulation and inherent properties. Conversely, regarding deep models, we find that it is after their size and complexity that their interpretable properties get hindered. Common CNNs rely on convolution as their corner stone, coupled with non-linear operations such as ReLU (Fukushima 1975), Sigmoid, and Softmax (Hopfield et al. 1985) among others. This aggregation of convolutions on one hand enables these models to process large quantities of data, and to a certain extent generalize; on the other hand, it also results in an extensive parameter count, often reaching of millions, and most recently, even billions (OpenAI 2018). The computational load required for inference, typically measured in Giga Floating Point Operations Per Second (GFLOPS) further compounds complexity. We expand on this in following sections.

Complementary to Lipton's proposal, Guidotti et al. 2018 points that interpretability is contained along different dimensions. For instance, a model can be understood on its entirety following a *Global* interpretation; whereas in situations where only the reasons leading to a specific prediction, a *Local* interpretation is found. Guidotti also considers time, more specifically *Time Limitation* as its availability is strictly correlated to the scenario where the model is used. Finally, the *Nature of User Expertise* covers the last dimension of interpretability; where knowing the experience of a user in a given task is considered to be a key aspect describing the interpretability of a model.

In recent years, Y. Zhang et al. 2021 suggested that three different dimensions encompass this study. The first dimension answers to the nature of an approach; where it can be either *Passive* or *Active*. The first direction in this dimension correlates to Lipton's *Post-hoc interpretability*, the latter follows the aforementioned *Transparency* property. Zhang's second dimension is addressed towards the type of explanations, where the order of explanatory power follows a hierarchy starting on its base level with examples, into attributions, leading into hidden semantics and finishing in rules. On one hand, the last dimension that Zhang covers explanations in the input space, where a *Local* explanation describes the network prediction following individual samples. On the other hand, a *Global* explanation describes the network as a whole. This last dimension is similar to the first point described by Guidotti et al. 2018. Most importantly, Zhang's dimensions are not used separately to categorize an approach, instead according to the methodology properties and the nature of the explanation provided.

Property	Attribution Type	Method	Reference
Transparency	Rule Based	Tree Regularization for Deep Models	(Wu, Hughes, et al. 2018)
		Regionally Faithful Explanations	(Wu, Parbhoo, et al. 2020)
	Hidden-Semantics	Network Dissection	(Bau et al. 2017)
		Interpretable CNN	(Q. Zhang et al. 2018)
		DAUnit	(Böhle, Fritz, et al. 2022)
		B-Cos Networks	(Böhle, Singh, et al. 2024)
	Prototype-Based	Case-Based Reasoning	(O. Li et al. 2018)
		This looks Like That	(Chen et al. 2019)
		Protopool	(Rymarczyk et al. 2022)
	Attribution-Based	Right for the Right Reasons	(Ross et al. 2017)
		Saliency Learning	(Ghaeini et al. 2019)
		Saliency Guided Training	(Ismail et al. 2021)
		LFI-CAM	(K. H. Lee et al. 2021)
		-	(H. Zhou et al. 2022)
Post-hoc Interpretability	Gradient-Based	Exp. Vectors	(Baehrens et al. 2010)
		Standard Gradient	(Simonyan, Vedaldi, et al. 2014)
		Guided Gradient	(Springenberg et al. 2014)
		LRP	(Bach et al. 2015)
		Integrated Gradient	(Sundararajan et al. 2017)
		Smoothgrad	(Smilkov et al. 2017)
	Learning-Based	-	(Dabkowski et al. 2017)
		FIDO	(C. Chang et al. 2019)
		CASM	(Zolna et al. 2020)
		Masker	(Phang et al. 2020)
		-	(Schulz et al. 2020)
	Masking-Based	LIME	(Ribeiro et al. 2016)
		Meaningful Perturbation	(R. C. Fong et al. 2017)
		RISE	(Vitali et al. 2018)
		Extremal perturbations	(R. Fong et al. 2019)
	Attention-Based	Raw-Attention	(Abnar et al. 2020)
		Rollout-Attention	
		TIBAV	(Chefer et al. 2021)
	CAM-Based	CAM	(B. Zhou, Khosla, et al. 2016)
		Grad-CAM	(Selvaraju et al. 2016)
		Grad-CAM++	(Chattopadhay et al. 2018)
		Score-CAM	(H. Wang, Du, et al. 2019)
		Axiom-CAM	(Fu et al. 2020)
		Ablation-CAM	(Desai et al. 2020)
		Layer-CAM	(Jiang et al. 2021)

Table 1.2: **Interpretability approaches** grouping using Lipton (Lipton 2018) and Zhang's definitions (Y. Zhang et al. 2021)

In this thesis we study interpretability according to Lipton's properties; nevertheless, as shown in the following subsections, we acknowledge their strengths and weaknesses, while denoting some of the most prominent works in each dimension. In subsection 1.3.1 we discuss *transparency*, its properties according to Lipton and its difficulties describing deep models, alongside a survey following Zhang's active dimension. Conversely, in subsection 1.3.2, we study *post-hoc interpretability* similarly to transparency. We present a brief summary of these methods in Table 1.2.

1.3.1 Transparency

Following Lipton's work, we further define the transparency property of models as the opposite of opacity or black-box behavior. Still, similar to the aforementioned works, transparency is considered at the model level (*simulatability*), individual components (*decomposability*) and the training algorithm (*algorithmic transparency*). Delving into detail, *Simulatability* on a strict definition answers to the capability of a model to be fully contemplated by an individual on its entirety at once. Currently, with the increase of model complexity this understanding level is then redefined as the capacity of a model to be readily presented to a user with visual or textual artifacts. This in turn can be demonstrated by *post-hoc interpretations*.

Regarding *decomposability*, Lipton suggests that an interpretable model has to present an intuitive explanation, regardless of it being an input, parameter or calculation. In this dimension it is then stated that a computation follows an association between features, in accordance to *intelligibility* as proposed in (Lou et al. 2012). Still, Lipton also points that this notion should not be accepted blindly as some parts of the computation steps can be fragile processing and calculation; for instance, image normalization can lead to prediction mismatch during inference.

Finally, Lipton argues that *algorithmic transparency* applies on the level of the learning algorithm itself. In short, this applies to the error surface and assumption of the existence of an optimal and/or unique solution to the problem. Nevertheless, this dimension is not met for deep learning models as the training process is not completely understood.

While Lipton's transparency properties appear to be applicable and theoretically sound, transparency is not fully studied alongside them. Furthermore, designing an approach or model covering these properties is a complicated task for deep models. However, with a brief modification of their definition, transparent properties can be further applied to deep image recognition. *Simulatability*, can be supported with *post-hoc interpretability* explanations. Conversely, deep learning models can be further *decomposed* into functional units; for instance, the ResNet architecture consists of residual blocks, where the structure of each block and its design has been studied thoroughly. Finally, concerning *algorithmic transparency*, although deep learning models present high dimension error surfaces and a high amount of local minima; still, ensuring a deterministic behavior leads to replicability, in turn demonstrating part of the training process.

According to the interpretable dimensions described by (Y. Zhang et al. 2021), transparency is suggested as an active change of the architecture of the network or the training process in order to provide explanations. Furthermore, according to the type of explanations provided by an interpretable approach, transparency methods can be explained according to logic rules, hidden semantics, attributions and explanations by example.

Rule-based methods. In this group, rules are used to provide an explanation. To do this, Wu et al. proposed training a decision tree on top of a deep learning model, acting as a regularization term and an approximation of said deep model (Wu, Hughes, et al. 2018, Wu, Parbhoo, et al. 2020). This process involves two steps. First a binary tree is trained, mapping inputs to the model prediction. Then the average path length between root to leaf nodes on the decision tree is calculated, covering all data points. Ultimately, Zhang argues that this regularization improves interpretability by forcing a model to be described by a decision tree.

Hidden semantics-based methods. This family of methods focuses on the filters and properties found alongside in different depths of the model. These approaches are based in observations of filter response to inputs at different depths, and the semantic information contained by them (B. Zhou, Bau, et al. 2019). One particular approach studying this is that of (Q. Zhang et al. 2018), introducing a loss term that encourages deep filters to represent a single concept. More recently, as an answer to semantics only being extracted from deep layers, and the information flow not being taken into consideration to provide an explanation; the work of Böhle et al. (Böhle, Fritz, et al. 2022, Böhle, Singh, et al. 2024) introduced the idea of *B-Cos networks* and alignment overall with the *Dynamic Alignment Unit.* In these works, alignment between weight-input alignment is emphasized by the inclusion of the B-cos transform and dynamic alignment unit. Moreover, the interpretations these methods provide insight over the entirety of the network, and they are possibly integrated into existing architectures.

Prototype-based methods. On this topic, Li et al. designs an architecture that incorporates an autoencoder and a standard classification network (O. Li et al. 2018). In detail, this model reduces dimensionality of inputs, producing high quality features for classification. The encoded features are then used to produce a probability distribution over the dataset classes. Interestingly, during the forward pass on the prototype network, these encoded features are transformed into prototype vectors; ultimately used to train the classifier. Therefore, the classification algorithm is distance-based on the low dimensional learned feature space. Another approach showcasing prototype learning is that contained within *This Looks Like That* (Chen et al. 2019). In this approach, Chen proposes a prototype layer that learns prototypes on top of existing architectures, aligning learn a set of prototypes to specific categories. One difficulty these prototype parts face, is their heavy computation and difficulty to train. As a response of this, ProtoPool (Rymarczyk et al. 2022) is recently proposed, allowing for prototype reutillization and differentiable assignment of prototypes to classes; speeding up the training process

Attribution-based methods. Similar to the regularization induced by the inclusion of decision trees, local attributes are improved during training through regularization (Ismail et al. 2021, H. Zhou et al. 2022, Ross et al. 2017, Ghaeini et al. 2019, K. H. Lee et al. 2021). These methods are often used in conjunction with *post-hoc interpretability* approaches, where an observation made on the explanation proposal is further taken into consideration to be modified during training. For instance, on LFI-CAM (K. H. Lee et al. 2021), an attention branch is included to learn attention feature importance during training; allowing for a straightforward saliency map computation while improving recognition properties. Similarly, saliency-guided training (Ismail et al. 2021) minimizes the Kullback Leibler divergence between the output of original and masked images.

1.3.2 Post-Hoc Interpretability

According to Lipton, post-hoc interpretability provides information directly from learned models. Complementary to this, Zhang points out that *passive* interpretability methods extract logical rules or understandable patterns from models. As noted in Table 1.2, a discrepancy in the ratio of *active-passive* methods exist, where the *passive* dimension presents more studies. Furthermore, according to the nature of explanation Lipton groups methods into *textual explanations, visualizations, explanations by example* and *local explanations*. In particular, Lipton argues that humans justify decisions verbally, suggesting it is possible to train a model to generate explanations describing the predictions of another model. For instance, NLP-based approaches for image captioning can be utilized to provide textual explanations (McAuley et al. 2013). Nevertheless, in recent years image-captioning approaches have become increasingly complex, introducing a degree of uncertainty into the explanations provided, as these recent systems are now transformer-based and their interpretability is not comprehensively studied yet. Lipton closes this discussion addressing that this kind of interpretations are open to scrutiny (J. Chang et al. 2009).

To provide interpretations on a visual manner according to Lipton, we find *explanations from visualization, explanations by example* and *local explanations*. The goal of explanations from visualization is to determine qualitatively what the model has learned. One instance of this is found on the work by Mahendran et al. 2015, where an image is forwarded through a discriminative CNN generating a representation. The authors then demonstrate that the original image can be recovered from deep-level representations by performing gradient descent on randomly initialized pixels.

Complementary to *explanations by visualization*, Lipton proposes *explanations by example*. These explanations are derived from explaining decisions for a model using examples deemed the most similar. In detail, once a model is trained it is possible to extract information from learned representations; these in turn can be used to explain a new sample by its proximity in this representation space to similar observations. On deep image recognition models, we point to the works of (Kim et al. 2014, Doshi-Velez et al. 2015), where case-based approaches are proposed for interpreting generative models. More recently, the work proposed by Rombach et al. 2020, where an Invertible Neural Network is connected at different stages of a model, disentangling deep representations and providing an inverse transformation into accessible semantic concepts which can be used to point at similar examples of a given input.

Regarding *local explanations*, Lipton acknowledges that describing the entirety of the mapping by a CNN is a difficult endeavor; and as such, it can be explained on a *local* manner. This local behavior is widely considered across interpretability studies, as previously mentioned in section 1.3 (Guidotti et al. 2018, Y. Zhang et al. 2021). Generally speaking, local explanations are often extracted via the computation of a *saliency map* highlighting the most important regions on an image describing a prediction. Studies describing local explanations are varied and many of them are build on top of transparency approaches; where salient properties can be enhanced via training or optimization. Conversely, local explanations are grouped into: gradient-based methods, learning-based methods, masked-based methods and class activation based-methods.

Gradient-based methods. This family of approaches leverages the property of image recognition models, wherein forwarding an example through a model allows for the visualization of the response of its weights during backpropagation in correlation to a specific prediction. This response is visualized in input space, where strong gradient information is expected in regions of the image containing information relevant to the corresponding prediction (Baehrens et al. 2010, Simonyan, Vedaldi, et al. 2014). However, gradient information often contains noise, making it challenging to interpret these modifications in a straightforward fashion (Adebayo et al. 2018). Consequently, modifications to backpropagation calculation have been proposed, including considering only positive values (Springenberg et al. 2014), adding noise on the input space for denoising (Smilkov et al. 2017), introducing rules and axioms to constrain the computation (Sundararajan et al. 2017) and proposal of concepts like relevance (Bach et al. 2015) to compute the importance pixel interaction leading to a prediction.

Learning-based methods. Approaches in this family of methods exhibit a behavior intersecting *transparency* based approaches, where an additional network or branch is learned atop an existing architecture to produce an explanation map for a given input. However, the lack of modifications on the model for interpretation is the key factor differentiating these approaches from *active* interpretability methods. Specific examples of learning methods comprise modifications inclusion of generative models to fill in information occluded from the classifier (C. Chang et al. 2019), masking salient points on the map to manipulate scores on the classifier (Dabkowski et al. 2017), inclusion of a *masker* side network to collect information on different levels of the network and produce a high resolution explanation (Phang et al. 2020), addition of a decoder to generate masks and a different classifier predicting the masked inputs (Zolna et al. 2020), and finally, a bottleneck on top of intermediary layers of the network to learn a saliency map per sample at the specific depth of the network (Schulz et al. 2020).

Occlusion or masking-based methods. Continuing on with modifications on the input space, by purposefully masking certain regions on the image and measuring the subsequent loss in predictive power, a saliency map can be produced. In particular, the masking process can provide extremal perturbations by measuring the maximal effect upon the activation of neurons leading to predictions (R. C. Fong et al. 2017, R. Fong et al. 2019), learning a simple masking model around the prediction to provide saliency maps that are locally faithful to the classifier (Ribeiro et al. 2016, Vitali et al. 2018) and iteratively by random masking the input image, probing the model and obtaining a saliency map from the linear combination between prediction weights and the random mask that was used to generate them.

Attention-based techniques. Given the innate properties of attention based architectures, most interpretability methods are ill fitted towards explaining the inference process of these models. Nevertheless, with the current design of these models it is possible to address salient information, although in a class-agnostic manner. In particular, in the computation of self attention, this information is highlighted by the product of *query* and *key* vectors by focusing on the [CLS] token; this in turn is called *Raw Attention.* Conversely, since attention is computed at every layer of a transformer, the flow of information can be traced across different layers across the network in an approach named *Rollout Attention* (Abnar et al. 2020). Inspired by *rollout attention* and gradient methods such as *LRP* (Bach et al. 2015) and the inadequacy property of CAM methods on transformers, a novel computation of relevance across layers providing high quality attribution maps is introduced, allowing at the same time for a quantitative study of the interpretable properties of these methodologies (Chefer et al. 2021).

CAM-based methods. By taking into consideration feature map information contained across different layers of a network, it is possible to gain insight into which image regions are highlighted at different depths within the model. However, understanding feature maps, which are often high-dimensional, present a considerable challenge. Conversely, information contained from the classifier layers of the model can be directly associated with classes. For example, the product of the weights of the classifier and the last feature map before GAP can generate a saliency map, known as Class Activation Maps (CAM) (B. Zhou, Khosla, et al. 2016). These approaches typically utilize feature information from deep layers in conjunction with a set of weights correlated with class information to compute explanations (Selvaraju et al. 2016, Chattopadhay et al. 2018, H. Wang, Du, et al. 2019, Fu et al. 2020, Desai et al. 2020 Jiang et al. 2021).

In this thesis we take particular interest in proposing and evaluating our interpretability methodologies based on CAM. In particular, we find CAM to be an appealing approach because of its ease of use and adaptation to interpret existing models. Moreover, it can be argued that while it is possible find and use a plethora of the most popular models towards recognition; providing explanations ought to be similarly easy to use, a property that CAM-based models share.

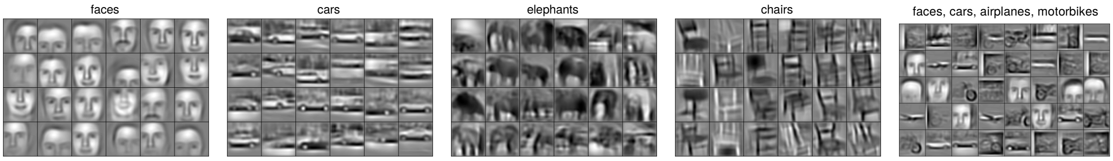


Figure 1.12: **Filter response** to learned classes alongside CNN depth (H. Lee et al. 2009).

CAM-based saliency maps One property that CNNs display is the presence of semantic information found within the deepest layers prior to the classifier. In detail, these models operate similarly manner to the visual cortex in the brain; basic textures and their orientation is processed in shallow layers, whereas deep layers associate this information into concepts (Hubel et al. 1959). Regarding CNNs, on one hand filter responses and low level information is captured by shallow layers on the model; on the other hand, semantic aggregation takes place within the deep layers of the model (H. Lee et al. 2009). This characteristic in turn is the main motivation behind making use of CAM methods: by addressing interpretations on the layers prior to the classifier, we reconstruct the features deemed salient according to the flow of information within a model. We present a representation of filter responses from texture to semantics in Figure 1.12.

Notation Consider a classifier network : $f\mathcal{X} \to \Re^C$ that maps an input image $\mathbf{u} \in \mathcal{X}$ to a logit vector $\mathbf{y} = f(\mathbf{u}) \in \Re^C$, where $\mathcal{X}$ is the image space and C is the number of classes. We denote by $y_c = f(\mathbf{u})_c$ the predicted logit and by $p_c = \text{softmax}(\mathbf{y})_c := e^{y_c} / \sum_j e^{y_j}$ the predicted probability for class c. For layer ℓ with K_ℓ channels, we denote by $A_\ell^k = f_\ell^k(\mathbf{u}) \in \Re^{h_\ell \times w_\ell}$ the feature map for channel $k \in \{1, \ldots, K_\ell\}$, with spatial resolution $h_\ell \times w_\ell$. Because of ReLU non-linearities, we assume that feature maps are non-negative. Similarly, we denote by $S_\ell \in \Re^{h_\ell \times w_\ell}$ a 2D saliency map.

Given a layer ℓ and a class of interest c, we consider saliency maps given by the general formula:

$$S_\ell^c(\mathbf{u}) := h\left(\sum_k w_k^c A_\ell^k\right), \tag{1.3}$$

where w_k^c are weights defining a linear combination over channels and h is an activation function, we present a visualization of this process in Figure 1.13. It is through the calculation of these weights or weighting coefficients that a plethora of saliency methods have been proposed using CAM. Informally, this can be referred to as *The Many Flavors of CAM*. Moreover, given the flexibility in defining this coefficient, we

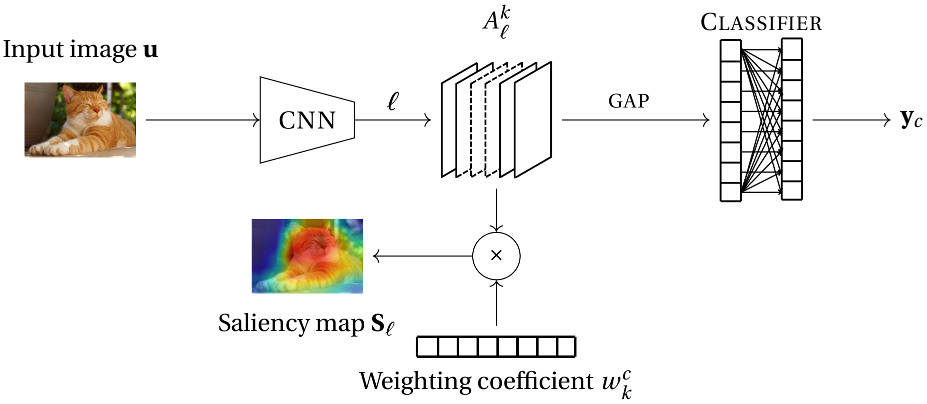


Figure 1.13: **CAM** based methodologies overview.

outline milestone alternatives for this calculation.

CAM (B. Zhou, Khosla, et al. 2016) is the original proposal for this family of methods. This approach is defined uniquely for the last layer L before the classifier and h the identity mapping and w_k^c is defined as the classifier weights that map the k-th channel in the feature map with class c. This first definition of class activation methods is not general, should a classifier be composed of a MLP; w_k^c would have to account for the multiple interactions across the stages on the aforementioned layer, which ultimately is not straightforward.

Grad-CAM (Selvaraju et al. 2016) is proposed as a generalization of *CAM*. Following the inconvenience of the calculation of the weighting coefficient on multi layered classifiers, this approach considers instead the flow of information on the network following the gradient generated by backpropagation following the logit of the class of interest. By doing so, the feature map to be used for an explanation can be selected from any layer ℓ at different depths of the network. Moreover, the identity mapping on this approach is $h = \text{ReLU}$ and the weighting coefficient is calculated in the following manner:

$$w_k^c := \text{GAP}\left(\frac{\partial y_c}{\partial A_\ell^k}\right), \tag{1.4}$$

where GAP is global average pooling. The motivation for ReLU is that we are only interested in features that have a positive effect on the class of interest, *i.e.* pixels whose intensity should be increased in order to increase y_c. Lastly, this approach also considers the computation of back-propagation using gradient refinements such as *guided backpropagation* (Springenberg et al. 2014), allowing for saliency maps to highlight more salient regions within the image.

Grad-CAM++ (Chattopadhay et al. 2018) continues the trend of refining the generated saliency map via gradient modifications. In particular, the computation of the gradient follows partial derivatives as a way to address shortcomings on the original Grad-CAM. Specifically, Grad-CAM is not robust to multiple instances of the same class within one image, and overall the localization of the attribution usually fails to be located over the entirety of the object of the class of interest. The computation of w_k^c for Grad-CAM is defined in the following manner:

$$w_k^c := \mathrm{GAP}\left[\frac{\frac{\partial^2 y_c}{(\partial A^k)^2}}{2\frac{\partial^2 y_c}{(\partial A^k)^2} + \mathrm{GAP}\left(A^k \frac{\partial^3 y_c}{(\partial A^k)^3}\right)}\right] \cdot \mathrm{ReLU}\left(\frac{\partial y^c}{\partial A^k}\right) \tag{1.5}$$

Similar to Grad-CAM, Grad-CAM++ uses $h = \mathrm{ReLU}$, is not computational expensive, provides high quality saliency maps with better localization properties and is yet another generalization of prior approaches. Nevertheless, one crucial contribution proposed alongside this attribution method consists of an evaluation methodology for saliency maps, explained in detail in subsection 1.3.3.

Score-CAM (H. Wang, Du, et al. 2019) is also defined for any layer ℓ with $h = \mathrm{ReLU}$ and weights $w_k^c := \mathrm{softmax}(\mathbf{u}^c)_k$. Softmax normalization considers positive channel contributions only and attends to few feature maps. Inspired by the idea of comparing the increase in confidence obtained by forwarding an image masked with the saliency map relating to class c, a vector $\mathbf{u}^c \in \Re^{K_\ell}$ compares a known baseline image $\mathbf{u}_b$ with the input image $\mathbf{u}$, for all channels k:

$$u_k^c := f(\mathbf{u} \odot n(\mathrm{up}(A_\ell^k)))_c - f(\mathbf{u}_b)_c, \tag{1.6}$$

where $\odot$ is the Hadamard product. For this to work, the feature map A_ℓ^k is adapted to $\mathbf{u}$ first: up denotes up-sampling to the spatial resolution of $\mathbf{u}$ and:

$$n(A) := \frac{A - \min A}{\max A - \min A} \tag{1.7}$$

is a normalization of matrix A into $[0,1]$. While Score-CAM does not need gradients, it requires as many forward passes through the network as the number of channels in the chosen layer, which is computationally expensive. Score-CAM is better understood in a visual manner as in Figure 1.14

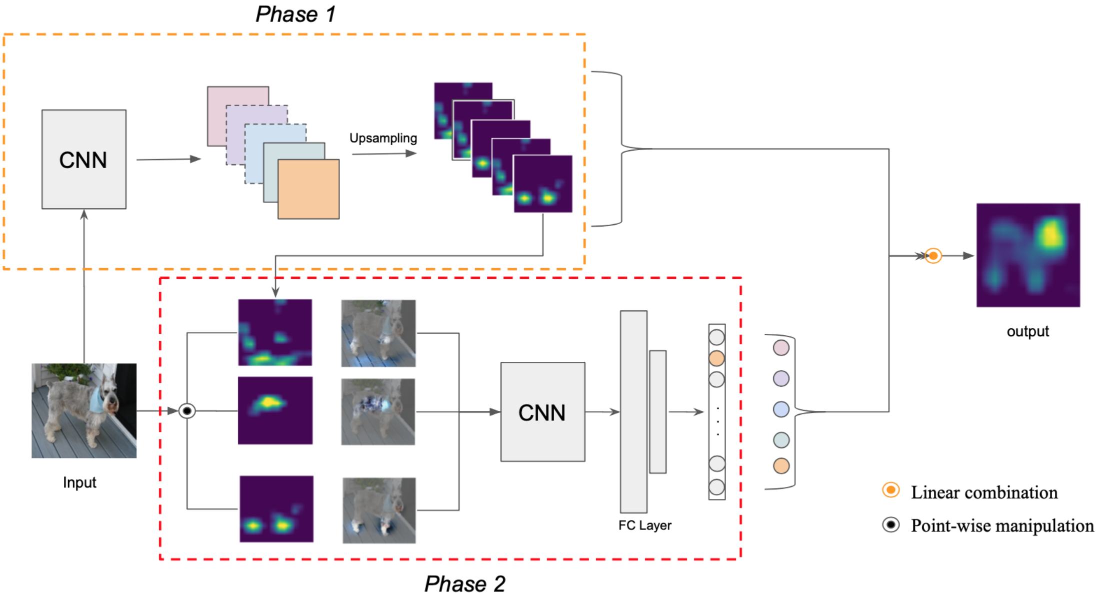


Figure 1.14: **Score-CAM** computation process (H. Wang, Du, et al. 2019)

Axiom-CAM (Fu et al. 2020) Departing from the usage of the increase in confidence and gradient information, this approach incorporates logical principles to guide the calculation of the saliency map. In particular, two axioms are introduced *sensitivity* and *conservation*. On one hand *sensitivity* considers that the importance of each feature map has to be equivalent to the score change caused by its removal. On the other hand, *conservation* points that changes in the saliency map come from a redistribution of the class score. In other words, sensitivity considers the loss of predictive power by the removal of high importance feature maps; *conservation* instead ensures that class scores are mainly dominated by feature maps rather than external factors. By maintaining the activation function $h = \text{ReLU}$, the formulation of the axiom-based weighting coefficient follows:

$$w_k^c := \text{GAP}\left(\frac{A^k}{\text{GAP}(A^k)}\frac{\partial y^c}{\partial A^k}\right). \tag{1.8}$$

Ablation-CAM (Desai et al. 2020) continuing the trend of computing w_k^c without making use of gradients, masking methods can be further incorporated in the production of saliency maps. More specifically, this approach determines the weight of individual feature maps via ablation perturbation. Going deeper, this approach evaluates the predictive power y_k^c of an individual channel k in feature map A^k when all other activations are zeroed out. In one way, this could be seen as similar to increase in confidence to produce the linear combination as in *Score-CAM*. Nevertheless, in the current approach, modifications are done on feature space and the predictive power is compared with that of predictions using A^k with all the feature maps unaltered.

Retaining activation $h = \text{ReLU}$, the formulation of this weighting coefficient follows:

$$w_k^c := \frac{y^c - y_k^c}{y^c} \tag{1.9}$$

Layer-CAM (Jiang et al. 2021) although CAM can be extracted at any point of a model following the generalities described previously, they do not take into consideration the flow of information and relevant features along the depth of a model. Nevertheless, with backpropagation gradient can be probed alongside the whole depth of a model; as such, this approach computes a saliency map by averaging the up-sampled attribution maps obtained at different depths from the model, generating a representation accounting for this information.

1.3.3 Evaluating Interpretability

Following the *desiredata of model interpretabiltiy*, proposing interpretations does not compose the only step in understanding the prediction process of a model; in order to claim to be interpretable, the effect of its explanations should be quantifiable for an approach or model. Lipton addresses this in the discussion section of *The Mythos of Model Interpretability*, where he points:

> *To be meaningful, any assertion regarding interpretability should fix a specific definition. If the model satiesfies a form of transparency, this can be shown directly. For post-hoc interpretability, papers ought to fix a clear objective and demonstrate evidence that the offered form of interpretation achieves it.*

On one hand, this definition is not completely up-to-date for deep image recognition models. In particular, given the difficulties of the original definition of transparency covering deep models, especially with the introduction of active interpretability approaches. A direct display of transparency is not straightforward, particularly given that some of these proposals involve modifying the behaviour of models or lack inherent interpretability. Conversely, post-hoc interpretations are frequently found in the input space. As a result, the impact of these modifications is traditionally evaluated through human centric assessments of *trust* or *reliablity*. In the other hand, a quantitative approach, involves probing changes in the predictive power of the input masked by the saliency map (*interpretable object recognition*), randomly perturbing the input guided by saliency information described by the attribution map (*causal analysis*), and considering these attributions in a weakly supervised object localization manner (*interpretable object localization*). We present a brief summary of these metrics in Table 1.3:

Evaluation Type	Name	Reference
Interpretble Object Recognition	Average Drop	(Chattopadhay et al. 2018)
	Average Increase	
	Coherence	(Poppi et al. 2021)
	Complexity	
	Average Drop in Coherency and Complexity	
Causal Analysis	Insertion	(Vitali et al. 2018)
	Deletion	
Interpretable Object Localization	Unit Interpretability	(Bau et al. 2017)
	Official Metric	-
	Localization Error	-
	Pixelwise F1	-
	Box Accuracy	(Choe et al. 2020)
	Standard Pointing Game	(J. Zhang et al. 2017)
	Energy Pointing Game	(H. Wang, Du, et al. 2019)
	Saliency Metric	(Dabkowski et al. 2017)

Table 1.3: **Interpretability Metrics** according to evaluation types

1.3.3.1 Trust and Reliability

Following the desiredata of model interpretability. In this, the interpretabiltiy properties of *trust* and *informativeness* are the main driving force of this approach. To begin with, it is desired to be able to trust a model on its predictions, the examples for which it is right and how often is it; this in turn can be seen via informativeness. Given the black-box behaviour of image recognition models, informativeness can be gained by taking interest in salient regions of an image leading to a prediction. Therefore, assessing human trust of a saliency map describing a prediction is traditionally used to assess interpretability (Ribeiro et al. 2016, B. Zhou, Khosla, et al. 2016, Selvaraju et al. 2016, Chattopadhay et al. 2018, Bau et al. 2017). Nevertheless, this evaluation methodology is not favoured in recent times, as human interpretations are not always aligned with what the model or classifier deems important; conversely, this kind of evaluation can be often biased or not replicable at all.

1.3.3.2 Interpretable Object Recognition

Proposed based on the capability of current image recognition models to provide outputs over the distribution of classes that it was trained in, for any given input. Conversely, since saliency maps are class-specific, a logical manner of assessing their interpretability properties involves masking on the input space and measuring the changes of the predictive power of the model with regards of the highlighted regions provided by the saliency map. This methodology is first proposed by Chattopadhay et al. 2018, where in addition of traditional human evaluation of saliency maps, their predictive capabilities are measured using Average Drop (AD), Average Increase (AI).

Notation Let p_i^c and o_i^c be the predicted probability for class c given as input the i-th test image $\mathbf{u}$ and its masked version respectively. Masking refers to element-wise multiplication with the saliency map, which is at the same resolution as the original image with values in $[0,1]$. Let N be the number of test images. Class c is taken as the ground truth.

AD quantifies how much predictive power, measured as class probability, is lost when we only mask the image; lower is better:

$$\mathrm{AD} := \frac{1}{N}\sum_{i=1}^{N}\frac{[p_i^c - o_i^c]_+}{p_i^c}\cdot 100. \tag{1.10}$$

AI, also known as *increase in confidence*, measures the percentage of images where the masked image yields a higher class probability than the original; higher is better:

$$\mathrm{AI} := \frac{1}{N}\sum_{i}^{N}\mathbb{1}_{p_i^c < o_i^c}\cdot 100. \tag{1.11}$$

AD and AI are not defined symmetrically. AD measures changes in class probability whereas AI measures a percentage of images. It is possible that the percentage is high while the actual increase is small. Hence, it is possible that an attribution method improves both. Indeed, (Poppi et al. 2021) observes that a trivial method called Fake-CAM outperforms state-of-the-art methods, including Score-CAM, by a large margin. Fake-CAM simply defines a saliency map where the top-left pixel is set to zero and is uniform elsewhere. This questions the purpose of AD and AI. Addressing this, Poppi proposes Average Drop in Coherency and Complexity (ADCC), a metric that combines AD and two complementary measurements: *Coherency* and *Complexity*. In detail:

Coherency answers the requirement that the CAM-attribution of one image has to equate to the corresponding attribution of the explanation map obtained by the combination of said attribution map and the original input image. This is defined using the Pearson Correlation Coefficient between both attributions following:

$$\mathrm{Coherency}(u) = \frac{\mathrm{Cov}\left(\mathrm{CAM}_c(\mathbf{u}\odot\mathrm{CAM}_c(\mathbf{u})), \mathrm{CAM}_c(\mathbf{u})\right)}{\sigma\mathrm{CAM}_c(\mathbf{u}\odot\mathrm{CAM}_c(\mathbf{u}))\sigma\mathrm{CAM}_c(\mathbf{u})} \tag{1.12}$$

Complexity is addressed to the requirement of CAM attributions being as *simple* or less complex as possible. It is defined using the L_1 norm as a proxy of complexity:

$$\mathrm{Complexity}(\mathbf{u}) = ||\mathrm{CAM}_c(\mathbf{u})||_1 \tag{1.13}$$

ADCC takes into consideration the previously introduced metrics and Average Drop (AD), encompassing these metrics taking their harmonic mean in the following manner:

$$\mathrm{ADCC}(\mathbf{u}) = 3\left(\frac{1}{\mathrm{Coherency}(\mathbf{u})} + \frac{1}{1 - \mathrm{Complexity}(\mathbf{u}) + \frac{1}{1-AD(\mathbf{u})}}\right)^{-1} \tag{1.14}$$

1.3.3.3 Causal Analysis

Inspired by the work described in (R. C. Fong et al. 2017, R. Fong et al. 2019), instead adding perturbations on an image level in order to generate and attribution map, Vitali proposes an evaluation scheme where images are modified according to their saliency maps, and the subsequent change of predictive power is measured. This methodology contemplates two styles of modifications: on one hand, the input image is blurred entirely, and the information is reconstructed by the Insertion (I) of pixels following the saliency map; conversely, information can be steadily removed from the image by iterative Deletion (D) of salient pixels (Vitali et al. 2018).

Notation let x be an input image with the corresponding saliency map s^c and N the number of pixels removed per step, we calculate the predicted probability p_n^c at step n for groundtruth class c using the model f. Similar to AD and Average Increase (AI) Vitali considers s^c to maintain the same image resolution as x and be normalized in values $[0,1]$.

Insertion measures the increase of probability when introducing pixels from an input image into a blurry version of itself, following the salient order described by the saliency map. Best described in algorithm 1:

Algorithm 1: Insertion Algorithm

```
Input: black-box $f$, image $x$, saliency map $s^c$, number of pixels $N$ removed per step.
Output: insertion score $ins$. $n \leftarrow 0$
$x' \leftarrow \mathrm{Blur}(x)$
$p_n^c \leftarrow f(x)$
while $x \neq x'$ do
    According to $s$, set the next $n$ pixels in $x'$ to corresponding pixels in $x$
    $n \leftarrow n+1$
    $p_n^c \leftarrow f(x')$
$ins \leftarrow \mathrm{AreaUnderCurve}(p_n^c \mathrm{vs.} i/n, \forall i = 0, \dots n)$
return $ins$
```

Deletion measures the decrease of probability as pixels are zeroed out in an input image, following the salient order described by the saliency map. Best described in algorithm 2:

```
Algorithm 2: Deletion Algorithm
  Input: black-box $f$, image $x$, saliency map $s^c$, number of pixels $N$ removed per step.
  Output: deletion score $del$.
  $n \leftarrow 0$
  $p_n^c \leftarrow f(x)$
  while $x$ has non-zero pixels do
    According to $s$, set the next $n$ pixels in $x$ to 0
    $n \leftarrow n+1$
    $p_n^c \leftarrow f(x)$
  $del \leftarrow \text{AreaUnderCurve}(p_n^c \text{vs.} i/n, \forall i = 0, \dots n)$
  return $del$
```

1.3.3.4 Interpretable object localization

Following in line with the desire of saliency maps aligning with the object belonging to the class they are computed for, interpretability properties are evaluated using object localization. One approach considering this, is found within the *Broden* dataset (Bau et al. 2017). In this work, interpretability is measured at the unit level, where individual channels from a convolutional layer are compared with semantic concepts previously annotated within the dataset.

In addition to the evaluation procedure proposed in Broden, weakly supervised evaluation approaches are taken into consideration for this task.

Notation Given the saliency map S^c obtained from test image $\mathbf{x}$ for ground truth class c. We denote by $S^c_{\mathbf{p}}$ its value at pixel $\mathbf{p}$. We binarize the saliency map by thresholding at its average value, and we take the bounding box of the largest connected component of the resulting mask as the predicted bounding box B_p, represented as a set of pixels. This box is compared against the set of ground truth bounding boxes $\mathscr{B}$, which typically contains 1 or 2 boxes of the same class c, or with their union $U = \cup\mathscr{B}$, again represented as a set of pixels. We also compare the predicted class label c_p with the ground truth label c. All metrics take values in $[0,1]$ and are expressed as percentages, except SM (1.22), which is unbounded.

Unit Interpretability Defined in Broden, this metric is different from those defined in the coming weakly-supervised approaches; compares each individual feature map A^k with the localization mask L_c of class c for a given image x. To compute this comparison, the activation map A^k is upsampled into the original image resolution and then binarized generating the binary mask M^k. This mask is used following:

$$\text{IoU}_{k,c} := \frac{\sum \left| M^k(\mathbf{u}) \cap L_c(x) \right|}{\sum \left| M^k(\mathbf{u}) \cup L_c(x) \right|} \tag{1.15}$$

Official Metric (OM) measures the maximum overlap of the predicted bounding box with any ground truth bounding box, requiring that the predicted class label is correct:

$$\mathrm{OM} := 1 - \left(\max_{B \in \mathscr{B}} \mathrm{IoU}(B, B_p) \right) \mathbb{1}_{c_p = c}, \tag{1.16}$$

where IoU is intersection over union.

Localization Error (LE) is similar but ignores the predicted class label:

$$\mathrm{LE} := 1 - \max_{B \in \mathscr{B}} \mathrm{IoU}(B, B_p). \tag{1.17}$$

Pixel-wise F_1 score (F1) is defined as $F_1 = 2\frac{PR}{P+R}$, where *precision* P is the fraction of mass of the saliency map that is within the ground truth union:

$$P := \frac{\sum_{\mathbf{p} \in U} S^c_{\mathbf{p}}}{\sum_{\mathbf{p}} S^c_{\mathbf{p}}} \tag{1.18}$$

and *recall* R is the fraction of the ground truth union that is covered by the saliency map:

$$R := \frac{\sum_{\mathbf{p} \in U} S^c_{\mathbf{p}}}{|U|}. \tag{1.19}$$

Box Accuracy (BA) (Choe et al. 2020) Given threshold values η and δ, we find the bounding box B^{η}_{p} of the largest connected component of the binary mask $\{\mathbf{p} : S_{\mathbf{p}} > \eta\}$ and require that it overlaps by δ with at least one ground truth box:

$$\mathrm{BoxAcc}(\eta, \delta) := \max_{B \in \mathscr{B}} \mathbb{1}_{\mathrm{IoU}(B^{\eta}_{p}, B) \geq \delta}. \tag{1.20}$$

After averaging over the test images, we take the maximum of this measure over a set of values η and then the average over a set of values δ.

Standard Pointing game (SP)(J. Zhang et al. 2017) We find the pixel $\mathbf{p}^* := \arg\max_{\mathbf{p}} S^c_{\mathbf{p}}$ having the maximum saliency value and require that it lands in any of the ground truth bounding boxes:

$$\mathrm{SP} := \mathbb{1}_{\mathbf{p}^* \in U}. \tag{1.21}$$

Energy Pointing game (EP) (H. Wang, Du, et al. 2019) is equivalent to precision (1.18).

Saliency Metric (SM) (Dabkowski et al. 2017) penalizes the size of the predicted bounding box B_p relative to the image and the cross-entropy loss:

$$\mathrm{SM} := \log \max \left(0.05, \frac{|B_p|}{hw} \right) - \log p^c, \tag{1.22}$$

where $h \times w$ is the input image resolution and p^c is the predicted probability for ground truth class label c.

1.3.4 Discussion

Following the rapid development of image recognition models and the subsequent need to understand their behavior, interpretability has become a sought after task within the community. As a result, several studies have been introduced over time, as demonstrated in the previous section. Yet, the work proposed by Lipton is unique in questioning what interpretability truly entails. Instead, most interpretability approaches are designed to address specific needs of current image recognition models, ranging from simplifying a model, to describing its components, disentangling embeddings on feature space and ultimately explaining predictions by responses on the input space. We suggest that some recurring issues that interpretability presents is an outdated vision on some properties and a lack of consensus, either in definitions, and evaluation.

Starting with propositions, according to different researchers interpretability is described across various dimensions depending on the nature of the approach. In this thesis we chose to follow the original propositions of Lipton, describing interpretability according to the properties of *transparency* and the ability to provide *post-hoc interpretations*. We acknowledge these properties as base descriptors for interpretability. However, these properties are ill-fitted for describing current image recognition models according to their original definitions. In particular, transparency on its preliminary definition applies mostly to traditional machine learning proposals. Nevertheless, by aligning transparency with the active dimension proposed by Y. Zhang et al. 2021, the definition holds, and subsequent studies adhere it.

Regarding saliency, we also note that this study is-ill formulated. When we obtain a saliency map, *how do we define what is important in an image?*. On one hand, although computer vision draws heavy inspiration from by human vision, the reasoning process is different between human and machine. In particular, a human might identify salient parts to describe an object in particular, differently that a machine would. Conversely, when discussing an attribution, *who are we considering the explanation for?*. As we mentioned previously, saliency is not well aligned for humans and machines, especially when machines derive their knowledge from context. Consequently, when evaluating the interpretable properties of an attribution, we find that usually those that provide the best results in terms of metrics are not usually the ones that a human would consider best. This question remains open up to this day.

Regarding evaluation of interpretability methods, in subsection 1.3.3 we made mention of current evaluation methodologies, but, several questions arise from them. To begin with, and relating to the previous paragraph, it is safe to assume that the quantitative metrics presented are taking into consideration relevance towards the classifier. In particular *these methods are telling us which representation explains the best a given class for the model.* And yet, in this case we can also wonder about *which class should we inquire about; groundtruth or predictions?*. In most applications researchers extract attributions and provide explanations for groundtruth objects. However, in real-world applications we expect models to predict incorrectly some instances in the task that they are given; these instances then require explanations for the class being predicted. This is crucial given Lipton's desidedata of model interpretability. We usually care the most for the instances in which the model fails, and we have to provide accountability for the effects these inferences provide.

In addition to this selection of instances to provide explanations for, quantitative evaluation suffers from another drawback: a lack of homogeneity in evaluation procedures. Complementary to the qualitative evaluation of attribution maps via visual inspection and human criteria, quantitative evaluation ought to be more robust, replicable and homogeneous. Yet, we observe variations in evaluation procedures across proposals, and while an effort to provide fair comparisons, complete insight on the behavior of the proposed methodologies is not attained. To exemplify, with the introduction of *objective evaluation for object recognition* in Grad- CAM++ (Chattopadhay et al. 2018), the evaluation procedure only required generation of visualizations for the entirety of the validation set of ImageNet and Pascal VOC 2012, whereas to assess the performance of Score-CAM (H. Wang, Du, et al. 2019), a small subset of two thousand random images is chosen, thus negating replicability. This is done to circumvent the discussion of the high computational cost required to compote Score-CAM attributions, which is not clearly discussed in its article. Complementary to this, Chattopadhay et al. 2018 provides analysis for VGG-16, ResNet-50 and AlexNet, while Score-CAM presents results only for VGG. This is not the only occurrence of these phenomena, upon the introduction of methodologies such as Integrated Score-CAM (Naidu et al. 2020), Ablation-CAM (Desai et al. 2020) and Layer-CAM (Jiang et al. 2021), the evaluation procedures are found not to be standardized between approaches.

A last point we want to highlight is addressed towards claims of interpretability upon the proposal of models. To begin, while introducing image recognition models in section 1.2, we found several approaches such as Conformer (Peng et al. 2021), Scouter (L. Li et al. 2021) and LFI-CAM (K. H. Lee et al. 2021) claiming to produce high performance image recognition architectures with built-in interpretability properties. However, these claims are sustained only with qualitative results in the shape of attribution maps such as Class Activation Maps (CAM) or attention visualization. Taking into consideration all the points we have discussed so far in this section, this exemplifies the challenges when discussing interpretability. On one hand, visually assessing the quality of explanation maps implies the alignment with human reasoning towards describing what is important within an image. Conversely, attribution maps are often shown mostly for groundtruth classes, not predictions, leaving in turn open the question of what is relevant for these instances. Lastly, qualitative measurements are not meaningful to claim the performance of an attribution compared with another, in particular given the misalignment between human and machine recognition processes.

In this dissertation we investigate machine-centered interpretability. In particular, we want to understand the inference process of a model and the key factors producing a prediction. Additionally, we align ourselves with the standard practice of evaluating explanations for instances of correct prediction. Consequently, we acknowledge that failure cases require further investigation from our part. Regarding the evaluation of our methodologies, we conduct standardized experimentation in our approaches, unless stated otherwise. Finally, we validate our claims of improvements of interpretability properties with our evaluation protocols. We conduct extensive evaluation for each of our proposals.

2 Opti-CAM: Optimizing saliency maps for interpretability

Table of contents

2.1 Introduction 60
2.2 Motivation 61
2.3 Opti-CAM 62
2.4 Average Gain 64
2.5 Experiments 64
2.5.1 Implementation details 64
2.5.2 Datasets 65
2.5.3 Evaluation 65
2.6 Qualitative Evaluation 66
2.7 Quantitaive Evaluation 67
2.7.1 Image classification 67
2.7.2 Weakly Supervised Approach 71
2.8 Ablation study 73
2.9 Discussion 74
2.10 Conclusion 75

2.1 Introduction

Within existing attribution approaches for interpretable saliency map generation, the CAM (B. Zhou, Khosla, et al. 2016) based family of methods takes special research interest given its dependence of existing information and properties of a given model to generate explanations. In particular, following Equation 1.3, modifying computation of the weighting coefficient w_k^c results in a different attribution being generated. Moreover, this computation can be altered, for instance by relying on information found while performing the backward pass (Selvaraju et al. 2016, Chattopadhay et al. 2018, Fu et al. 2020, Smilkov et al. 2017) and the forward pass (H. Wang, Du, et al. 2019) of the model during inference. Nevertheless, we observe that among existing weighting coefficient computation proposals, none has been directed at maximizing the predicted probability of the generated saliency maps.

Complementary to CAM methods, we observe that within attribution methods based on extremal perturbations (R. Fong et al. 2019) or IBA (Schulz et al. 2020), their class scores are optimized via gradient descent. In this regard, it can be stated that these masks then become variables within input-feature space, and the aforementioned scores then become a function of said masking. However, it is important to point that optimizing these masks ultimately becomes an expensive process, as several constraints are needed to control the masking area.

Drawing inspiration from the aforementioned observations, we propose *Opti-CAM*, an attribution method that generates saliency maps with enhanced interpretability. In particular, we hypothesize that the weighting coefficient w_k^c can be optimized to attain this task. Moreover, we suggest that should the predicted probability of the attribution map be optimized, we can gain insight within the regions of the image that appear to be the most important for the classifier. We define our approach in section 2.3

In addition to the proposal of an attribution method in this chapter, we design a complementary interpretability evaluation metric of saliency maps. In particular, based on the remarks found in Fake-CAM (Poppi et al. 2021), we observe that existing metrics such as AD (1.10) and AI (1.11) can be manipulated. As a result of this, we argue that a complementary criterion is missing regarding *Objective Evaluation for Object Recognition*. In section 2.4 we define this novel measurement under the name Average Gain (AG). To support our approach, we demonstrate our generated saliency maps in section 2.6, and we evaluate them in section 2.7.

To sum up, with the observations previously mentioned, in this chapter we propose a CAM variant that generates saliency maps by optimizing the weighting coefficient w_k^c, while also introducing a novel metric to complement the existing evaluation of attribution methods.

2.2 Motivation

From the CAM methods defined in subsection 1.3.2, we take particular interest on Score-CAM. In particular, Score-CAM considers each feature map as a mask in isolation. But, *what about linear combinations?* Given a vector $\mathbf{w} \in \Re^{K_\ell}$ with w_k its k-th element, let

$$F(\mathbf{w}) := f\left(\mathbf{u} \odot n\left(\operatorname{up}\left(\sum_k w_k A_\ell^k\right)\right)\right)_c . \tag{2.1}$$

If we assume that $\mathbf{u}_b = \mathbf{0}$ in (1.6) and define $n(\mathbf{0}) := \mathbf{0}$ in (1.7), then we can rewrite the right-hand side of (1.6) as

$$\frac{F(\mathbf{w}_0 + \delta \mathbf{e}_k) - F(\mathbf{w}_0)}{\delta}, \tag{2.2}$$

where $\mathbf{w}_0 = \mathbf{0}$, $\delta = 1$ and $\mathbf{e}_k$ is the k-th standard basis vector of $\Re^{K_\ell}$. This resembles the numerical approximation of the derivative $\frac{\partial F}{\partial w_k}(\mathbf{w}_0)$, except that δ is not small as usual. One could compute derivatives efficiently by standard backpropagation instead. It is then possible to iteratively optimize F with respect to $\mathbf{w}$, starting at any $\mathbf{w}_0$.

As an alternative, consider masking-based methods relying on optimization in the input space, like *meaningful perturbations* (MP) (R. C. Fong et al. 2017) or *extremal perturbations* (R. Fong et al. 2019). In general, optimization takes the form

$$S^c(\mathbf{u}) := \arg\max_{\mathbf{m} \in \mathcal{M}} f(\mathbf{u} \odot n(\text{up}(\mathbf{m})))_c + \lambda R(\mathbf{m}). \tag{2.3}$$

Here, a mask $\mathbf{m}$ is directly optimized and does not rely on feature maps, hence the saliency map $S^x(\mathbf{u})$ is not connected to any layer ℓ. The mask is at the same or lower resolution than the input image. In the latter case, upsampling is still necessary.

In this approach, one indeed computes derivatives by backpropagation and iteratively optimizes $\mathbf{m}$. However, because $\mathbf{m}$ is high-dimensional, there are constraints expressed by $\mathbf{m} \in \mathcal{M}$, *e.g.* $\mathbf{m}$ has a certain norm, and regularizers like $R(\mathbf{m})$, *e.g.* $\mathbf{m}$ is smooth in a certain way. This makes optimization harder or more expensive and introduces more hyperparameters like λ. One could simply constrain $\mathbf{m}$ to lie in the linear span of $\{A_\ell^k\}_{k=1}^{K_\ell}$ instead, like all CAM-based methods.

2.3 Opti-CAM

As noted in section 2.2, we obtain a saliency map as a convex combination of feature maps by optimizing a given objective function with respect to the weights. In particular, following (H. Wang, Du, et al. 2019), we use channel weights $w_k := \text{softmax}(\mathbf{u})_k$, where $\mathbf{u} \in \Re^{K_\ell}$ is a variable. We then consider saliency map S_ℓ in layer ℓ as a function of both the input image $\mathbf{x}$ and variable $\mathbf{u}$:

$$S_\ell(\mathbf{x}; \mathbf{u}) := \sum_k \text{softmax}(\mathbf{u})_k A_\ell^k. \tag{2.4}$$

In comparison with (1.3), h is the identity mapping, because feature maps are non-negative and weights are positive.

Optimization Now, given a layer ℓ and a class of interest c, we find the vector $\mathbf{u}^*$ that maximizes the classifier confidence for class c, when the input image $\mathbf{x}$ is masked

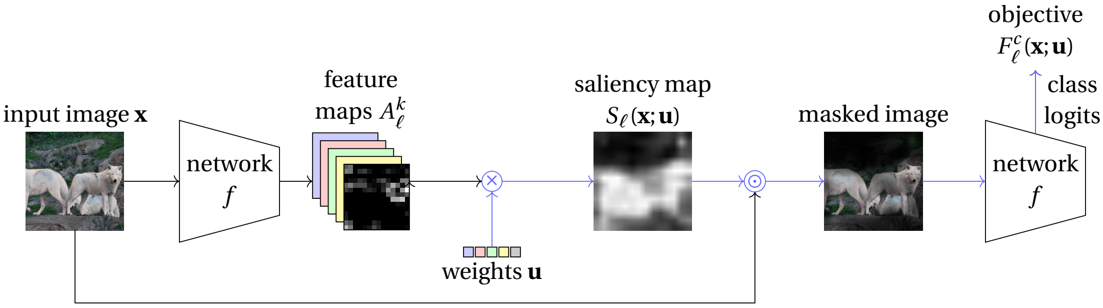


Figure 2.1: **Overview of Opti-CAM**. We are given an input image $\mathbf{x}$, a fixed network f, a target layer ℓ and a class of interest c. We extract the feature maps from layer ℓ and obtain a saliency map $S_\ell(\mathbf{x};\mathbf{u})$ by forming a convex combination of the feature maps ($\times$) with weights determined by a variable vector $\mathbf{u}$ (2.4). After upsampling and normalizing, we element-wise multiply ($\odot$) the saliency map with the input image to form a "masked" version of the input, which is fed to f. The objective function $F_\ell^c(\mathbf{x};\mathbf{u})$ measures the logit of class c for the masked image (2.6). We find the value of $\mathbf{u}^*$ that maximizes this logit by optimizing along the path highlighted in blue (2.5), as well as the corresponding optimal saliency map $S_\ell(\mathbf{x};\mathbf{u}^*)$ (2.7).

according to saliency map $S_\ell(\mathbf{x};\mathbf{u}^*)$:

$$\mathbf{u}^* := \arg\max_{\mathbf{u}} F_\ell^c(\mathbf{x};\mathbf{u}), \tag{2.5}$$

where we define the objective function:

$$F_\ell^c(\mathbf{x};\mathbf{u}) := g_c(f(\mathbf{x} \odot n({}^{(}S_\ell(\mathbf{x};\mathbf{u}))))). \tag{2.6}$$

Here, the saliency map $S_\ell(\mathbf{x};\mathbf{u})$ is adapted to $\mathbf{x}$ exactly as in (1.6) in terms of resolution and normalization. For *normalization function n*, the default is (1.7). The *selector function* g_c operates on the logit vector $\mathbf{y}$; the default is to select the logit of class c, *i.e.* $g_c(\mathbf{y}) := y_c$.

Putting everything together, we define:

$$S_\ell^c(\mathbf{x}) := S_\ell(\mathbf{x};\mathbf{u}^*) = S_\ell(\mathbf{x};\arg\max_{\mathbf{u}} F_\ell^c(\mathbf{x};\mathbf{u})), \tag{2.7}$$

where S_ℓ and F_ℓ^c are defined by (2.4) and (2.6) respectively. The objective function F_ℓ^c (2.6) depends on variable $\mathbf{u}$ through S_ℓ (2.4), where the feature maps $A_\ell^k = f_\ell^k(\mathbf{x})$ are fixed. Then, (2.6) involves masking and a forward pass through the network f, which is also fixed.

Figure 2.1 is an abstract illustration of our method, called Opti-CAM, without details like upsampling and normalization (2.6). Optimization takes place along the highlighted path from variable $\mathbf{u}$ to objective function F^c_ℓ. The saliency map is real-valued and the entire objective function is differentiable in $\mathbf{u}$. We use Adam optimizer (Kingma et al. 2015) to solve the optimization problem (2.5).

2.4 Average Gain

Continuing on with observations of CAM-based Saliency maps, we recall the observation made for *Fake-CAM* Poppi et al. 2021. In particular, we note that traditional interpretability measurements such as AD and AI can be deceiving; as perfect scores can be nearly achieved for AD by masking all but one pixel in the Saliency Map. This is used to motivate the definition of a number of metrics that are orthogonal to the task at hand, *i.e.* measuring the effect of masking to the classifier. By contrast, we address the problem by introducing a new metric to be paired with AD as a replacement of AI: *Average Gain*.

Average Gain (AG) quantifies how much predictive power, measured as class probability; is gained when we mask the image. We define this metric in the following manner, where higher is better:

$$\mathrm{AG}(\%) := \frac{1}{N}\sum_{i=1}^{N}\frac{[o^c_i - p^c_i]_+}{1-p^c_i}\cdot 100. \tag{2.8}$$

This definition is symmetric to the definition of average drop, in the sense that in absolute value, the numerator in the sum of AD, AG is the positive and negative part of $p^c_i - o^c_i$ respectively and the denominator is the maximum value that the numerator can get as a function of o^c_i, given that $0 < o^c_i < p^c_i$ and $p^c_i < o^c_i < 1$ respectively. The two metrics thus compete each other, in the sense that changing o^c_i to improve one leaves the other unchanged or harms it. As we shall see, an extreme example is Fake-CAM, which yields near-perfect AD but fails completely on AG.

2.5 Experiments

We evaluate Opti-CAM and compare it quantitatively and qualitatively against other state-of-the-art methods on a number of datasets and networks. We report classification metrics with execution times, and we provide visualizations, an ablation study and a study on the suitability of localization ground truth.

2.5.1 Implementation details

All input images are resized to 224 × 224 × 3. To optimize the saliency map with Opti-CAM (2.5), we use the Adam (Kingma et al. 2015) optimizer with learning rate 0.1 by default, setting the maximum number of iterations to 100 and stopping early when the change in loss is less than 10^{-10}. For VGG16, we generate the saliency map (2.4) from the feature maps of the last convolutional layer before max pooling by default, *i.e.* convolutional layer 3 of block 5. For ResNet50, we choose the last convolutional layer by default, *i.e.* convolutional layer 3 of bottleneck 2 of block 4. For ViT and DeiT, we choose the last self-attention block by default, *i.e.* layer normalization of self-attention block 12.

2.5.2 Datasets

ImageNet We use the validation set of ImageNet ILSVRC 2012 (Krizhevsky et al. 2012, Russakovsky et al. 2015), containing $50,000$ images evenly distributed over the $1,000$ categories. For the ablation study and for timing, we sample $1,000$ images from this set. Concerning the localization experiments, bounding boxes from the localization task of ILSVRC [1] are used on the same validation set.

Medical data We use two medical image datasets, namely *Chest X-ray* (Kermany et al. 2018) and *Kvasir* (Pogorelov et al. 2017).

Networks For all datasets, we use the pretrained ResNet50 (He, X. Zhang, et al. 2016) and VGG16 (Simonyan and Zisserman 2015) networks with batch normalization (Ioffe et al. 2015) from the Pytorch model zoo[2]. For ImageNet, we further use the pretrained ViT-B (16-224) (Dosovitskiy et al. 2020) and DeiT-B (16-224) (Touvron, Cord, Douze, et al. 2021) from Pytorch image models (timm)[3].

2.5.3 Evaluation

Metrics We use *AD* and *AI* (Chattopadhay et al. 2018) metrics, as well as the proposed *AG*, to measure the effect on classification performance of masking the input image by a saliency map. In addition, we report *Insertion (I)* and *Deletion (D)* (Vitali et al. 2018) and highlight their limitations. Using classification metrics, we show the limitations of using the localization ground truth for the evaluation of attribution methods. In subsection 2.7.2, we provide a number of localization metrics from the *weakly-supervised object localization* (WSOL) task of ILSVRC2014 [4].

Methods We compare against the following state-of-the-art methods: Grad-CAM (Selvaraju et al. 2016), Grad-CAM++Chattopadhay et al. 2018, Score-CAM (H. Wang, Du, et al. 2019), Ablation-CAM (et al. 2020), XGrad-CAM (Fu et al. 2020), Layer-CAM (Jiang et al. 2021), ExtremalPerturbation (R. Fong et al. 2019) and HiRes-CAM (Draelos et al. 2020). Implementations are obtained from the PyTorch CAM library[5] or

[1] https://www.image-net.org/challenges/LSVRC/2012/index.php
[2] https://pytorch.org/vision/0.8/models.html
[3] https://github.com/rwightman/pytorch-image-models
[4] https://www.image-net.org/challenges/LSVRC/2014/index#
[5] https://github.com/jacobgil/pytorch-grad-cam

TorchRay[6]. For transformer models, we also compare against raw attention (Dosovitskiy et al. 2020), rollout (Abnar et al. 2020) and TIBAV Chefer et al. 2021[7].

Image normalization It is standard that images are normalized before feeding them to a network. By doing so however, we cannot reproduce the results published for the baseline methods; rather, all results are improved dramatically. We can obtain results similar to published ones by *not* normalizing. We believe normalization is important, and we include it in all our experiments.

2.6 Qualitative Evaluation

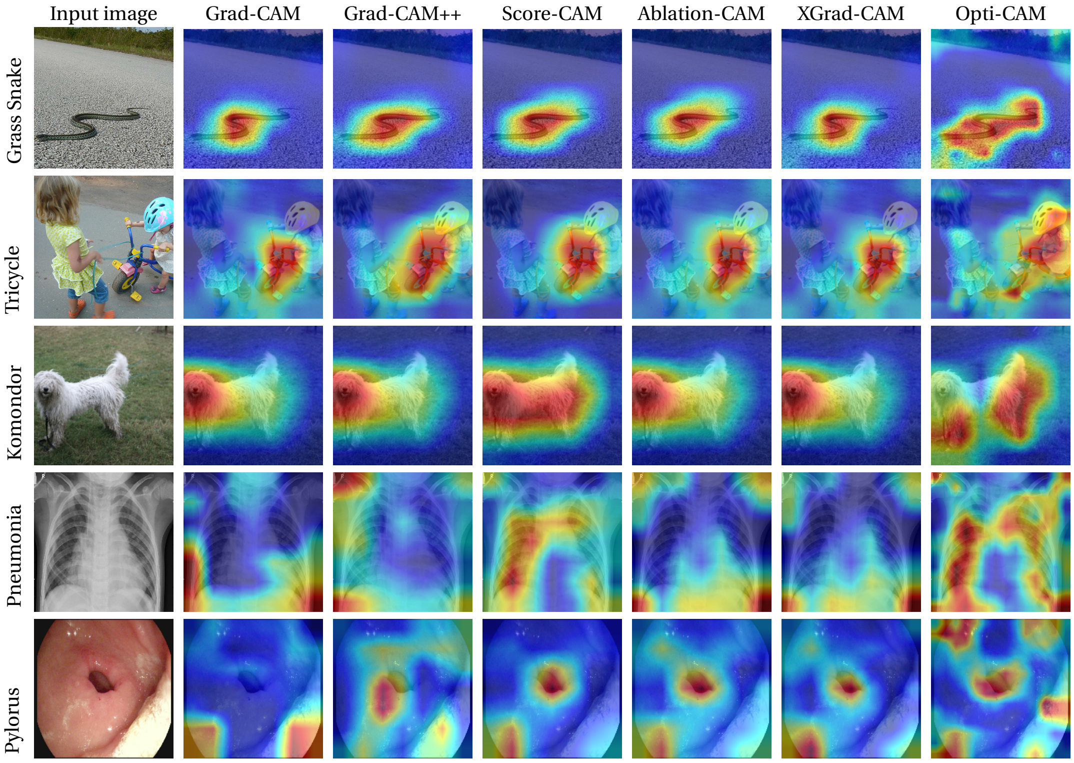


Figure 2.2: **Saliency maps obtained** by different methods for ImageNet (top two rows), Chest X-ray (row 3) and Kvasir (row 4) with VGG. Ground truth class shown on the left of the input image.

Figure 2.2 illustrates saliency map examples from ImageNet, Chest X-ray and Kvasir datasets. Opti-CAM saliency map is in general more spread out. This better highlights

[6] https://github.com/facebookresearch/TorchRay
[7] https://github.com/hila-chefer/Transformer-Explainability

full objects, multiple instances or background context, which may be taken into account by the model. On Chest X-ray, Opti-CAM and Score-CAM are the only methods that capture the chest, while all others focus on image corners.

2.7 Quantitaive Evaluation

2.7.1 Image classification

CNN Table 2.1 shows ImageNet classification metrics using VGG16 and ResNet50. Our Opti-CAM brings impressive performance in terms of Average Drop (AD) and Average Increase AI metrics. That is, not only impressive improvement over baselines, but near-perfect: near-zero AD and above 90% AI. Our new metric AG is lower, around 70% for Opti-CAM, but this is still several times higher than for all the other methods.

Method	ResNet50				VGG16			
	AD↓	AG↑	AI↑	T	AD↓	AG↑	AI↑	T
Fake-CAM	0.8	1.6	46.0	0.00	0.5	0.6	42.6	0.00
Grad-CAM	12.2	17.6	44.4	0.03	14.2	14.7	40.6	0.02
Grad-CAM++	12.9	16.0	42.1	0.03	17.1	10.2	33.4	0.02
Score-CAM	8.6	26.6	56.7	15.22	13.5	15.6	41.7	3.11
Ablation-CAM	12.5	16.4	42.8	18.26	15.5	12.6	36.9	2.98
XGrad-CAM	12.2	17.6	44.4	0.03	13.8	14.8	41.2	0.02
Layer-CAM	15.6	15.0	38.8	0.08	48.9	3.1	13.5	0.07
ExPerturbation	38.1	9.5	22.5	152.96	43.0	7.1	20.5	83.20
Opti-CAM	**1.5**	**68.8**	**92.8**	4.15	**1.3**	**71.2**	**92.7**	3.94

Table 2.1: **Classification metrics** on ImageNet validation set, using CNNs. AD/AI: average drop/increase (Chattopadhay et al. 2018); AG: average gain (ours); ↓ / ↑: lower / higher is better; T: Average time (sec) per batch of 8 images. Bold: best, excluding Fake-CAM.

Interestingly, Fake-CAM (Poppi et al. 2021) is the winner in terms of AD and second or third best in AI after Opti-CAM and Score-CAM, but fails completely AG. This is expected and makes Fake-CAM uninteresting as it should be: By only masking one pixel, the classification score can hardly drop (0.8% on ResNet50) and while it increases very often (on 46% of images), the gain is as little as the drop (0.7%). This makes the pair (AD, AG) sufficient as primary metrics and AI can be thought of as secondary, if important at all.

Table 2.1 also includes average execution time per image over the 1000-image ImageNet subset for all methods. Opti-CAM is slower than gradient-based methods that require only one pass through the network, but on par or faster than gradient-free methods. Indeed, we use a maximum of 100 iterations with one forward/backward pass per iteration, while Score-CAM and Ablation-CAM perform as many forward passes as channels. Hence, they are much slower on ResNet50 than VGG16. Extremal Perturbation does not depend on the number of channels but is very slow by performing a complex optimization in the image space.

Transformers Table 2.2 shows ImageNet classification metrics using ViT and DeiT. Unlike CAM-based methods that rely on a class-specific linear combination of feature maps, raw attention (Dosovitskiy et al. 2020) and rollout (Abnar et al. 2020) use the attention map of the [CLS] token from the last attention block and from all blocks respectively. This attention map depends only on the particular image and not on the target class, hence it is not really comparable. TIBAV (Chefer et al. 2021) uses both instance-specific and class-specific information.

Method	ViT-B				DeiT-B			
	AD↓	AG↑	AI↑	T	AD↓	AG↑	AI↑	T
Fake-CAM	0.3	0.4	48.3	0.00	0.6	0.3	44.6	0.00
Grad-CAM	69.4	2.5	12.4	0.14	33.5	1.7	12.5	0.11
Grad-CAM	86.3	1.5	1.0	0.15	50.7	0.9	7.2	0.13
Score-CAM	32.0	6.2	33.0	23.69	53.6	2.2	12.2	22.47
XGrad-CAM	88.1	0.4	4.3	0.13	80.5	0.3	4.1	0.12
Layer-CAM	82.0	0.2	2.9	0.24	88.9	0.4	2.6	0.24
ExPerturbation	28.8	6.2	24.4	133.52	60.9	2.0	8.5	129.12
RawAtt	92.6	0.2	2.8	0.02	95.3	0.0	1.8	0.02
Rollout	42.1	5.6	20.9	0.02	55.2	0.8	7.9	0.02
TIBAV	81.7	0.8	5.8	0.16	62.3	0.7	7.1	0.16
Opti-CAM	**0.6**	**18.0**	**90.1**	16.05	**0.9**	**26.0**	**83.5**	15.17

Table 2.2: **Classification metrics** on ImageNet validation set, using transformers. AD/AI: average drop/increase AG: average gain (ours); ↓ / ↑: lower / higher is better. T: Average time (sec) per batch of 8 images. Bold: best, excluding Fake-CAM.

Opti-CAM outperforms all other methods dramatically, reaching near-zero AD and AI above 80 or 90%. According to our new AG metric, Opti-CAM still works while all other methods fail, but AG is much more conservative than AI. On ViT-B for example, the classification score increases for 90.1% of the images by masking with Opti-CAM, but the gain is only 18.0% on average.

More metrics In this section, we show additional metrics including AOPC (Samek et al. 2016), Max-Sensitivity(Yeh et al. 2019) and ADCC (Poppi et al. 2021).

We use the code and suggested parameters of package Quantus[8] to measure AOPC and MS. In particular, patch size 14 and number of evaluation regions 10 for AOPC; lower bound 0.2 and number of samples 10 for MS. For ADCC, we use the official code[9].

We evaluate these metrics on ImageNet validation set using ResNet50 and VGG16. The results are reported in Table 2.3. Since AOPC shares the same philosophy as I/D, it is

[8] https://github.com/understandable-machine-intelligence-lab/Quantus
[9] https://github.com/aimagelab/ADCC

Method	ResNet50			VGG16		
	AOPC ↑	*MS* ↓	*ADCC* ↓	*AOPC* ↑	*MS* ↓	*ADCC* ↓
Grad-CAM	11.7	1.05	74.3	13.1	1.10	73.7
Grad-CAM++	11.6	1.04	73.6	11.6	1.09	74.6
Score-CAM	10.2	1.04	61.0	11.0	1.09	73.9
XGrad-CAM	11.9	1.05	74.3	13.1	1.10	73.9
Ablation-CAM	11.1	1.04	71.5	12.5	1.10	75.5
Layer-CAM	**13.0**	1.22	61.1	**13.3**	1.25	51.7
ExPerturbation	12.0	1.07	**26.0**	11.2	1.09	**42.8**
Opti-CAM (ours)	6.3	**1.03**	65.5	8.9	**1.06**	70.0

Table 2.3: **AOPC/MS/ADCC** scores on ImageNet validation set.

not a surprise that Opti-CAM has poor performance on AOPC. Opti-CAM achieves the best performance on MS. The experimental results are shown in Table 2.4 for CNNs and transformers. ExPerturbation (R. Fong et al. 2019) is expected to perform best in insertion because its optimization objective is very similar to this evaluation metric, using blurring for masked regions. However, ExPerturbation (ibid.) only performs best on ResNet50. TIBAV (Chefer et al. 2021), which is designed for transformers, outperforms the other methods on DeiT and ViT. According to the results of Insertion/Deletion, Opti-CAM has low performance, but there is no clear winner on either CNNs or transformers.

Method	ResNet50		VGG16		ViT-B		DeiT-B	
	I↑	D↓	I↑	D↓	I↑	D↓	I↑	D↓
Fake-CAM	50.7	28.1	46.1	26.9	57.4	33.3	57.5	34.2
Grad-CAM	66.3	14.7	**64.1**	11.6	62.9	19.8	61.8	17.5
Grad-CAM++	66.0	14.7	62.9	12.2	56.7	29.3	60.5	21.9
Score-CAM	65.7	16.3	62.5	12.1	**66.5**	15.1	60.6	24.4
XGrad-CAM	66.3	14.7	**64.1**	11.7	55.6	26.5	55.2	31.1
Layer-CAM	67.0	**14.2**	58.3	**6.4**	62.9	14.6	61.6	21.2
ExPerturbation	**70.7**	15.0	61.1	15.0	64.4	18.4	62.1	27.0
Ablation-CAM	65.9	14.6	63.8	11.4	-	-	-	-
RawAtt	-	-	-	-	62.2	17.9	56.3	29.3
Rollout	-	-	-	-	64.8	15.2	56.7	32.8
TIBAV	-	-	-	-	66.1	**14.1**	**63.7**	**16.3**
Opti-CAM (ours)	62.0	19.7	59.2	11.0	60.5	22.0	59.2	22.8

Table 2.4: **I/D: insertion/deletion** (Vitali et al. 2018) scores on ImageNet validation set; ↓ / ↑: lower / higher is better.

Insertion/Deletion include 224 steps of binarization, with a set of 224 pixels being inserted/deleted at each step. If these pixels are all inserted over a single small area, the effect on the classifier is more immediate than when sparsely inserting pixels over multiple areas. The same observation holds for deletion. By contrast, Opti-CAM attempts to find regions that contribute to the classification as a whole. There is no guarantee that those regions are effective when used in isolation.

To further understand the behavior of Opti-CAM, we investigate in Figure 2.3 examples where Score-CAM succeeds (insertion score greater than 90 and deletion score less than 10) and Opti-CAM fails (insertion score less than 70 and deletion score greater than 15). Compared with Score-CAM, the saliency maps obtained by Opti-CAM are more spread out and highlight several parts of the object and background context. In most of the cases, Opti-CAM fails I/D because it not only finds the object but also attaches importance to the background.

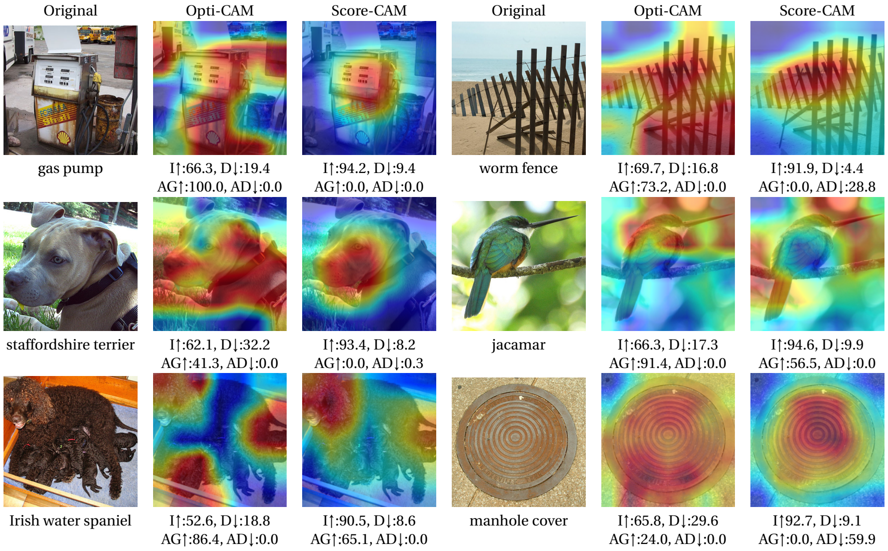


Figure 2.3: **Failure examples** of Opti-CAM regarding insertion/deletion.

We argue that this is not a failure. As we will see in our localization experiment in Table 2.5 indicates, the background is useful in discriminating a class. Often, the network recognizes the background better than the object itself. For example, a gas pump is likely to be seen with a truck, and a hare is often seen on grass. Several parts of the object are highlighted by Opti-CAM for the worm fence, terrier dog, hare, and manhole cover. Finally, several instances of spaniel dog are found by Opti-CAM.

Object localization Localization metrics are used to measure the precision of saliency maps relative to groundtruth bounding boxes of the foreground object of interest. These metrics originate from weakly supervised localization (WSOL). However, the objectives of WSOL and explaining the decision of a DNN are not necessarily aligned, since context may play an important role in the decision (Shetty et al. 2019, Rao et al. 2022).

To investigate the relative importance of the object and its context, we measure classification metrics when using the bounding box B itself as a saliency map as well as its complement $I \setminus B$, where I is the image. We also evaluate the intersection $B \cap S$ of the saliency map S with the bounding box and with its complement ($S \setminus B$).

As shown in Table 2.5, the ground truth region of the object is not the only one responsible for the network decision. For example, the bounding box fails both when used as a saliency map itself and when combined with any saliency map, by harming all classification metrics. Even the complement is more effective than the bounding box itself, either alone or when combined. These findings support the hypothesis that localization metrics based on the ground truth bounding box are not necessarily appropriate for evaluating explanations of network decisions. Classification metrics are clearly more appropriate in this sense.

METHOD	AD↓			AG↑			AI↑		
	S	$B \cap S$	$S \setminus B$	S	$B \cap S$	$S \setminus B$	S	$B \cap S$	$S \setminus B$
$S := B$	67.2	–	–	2.3	–	–	9.2	–	–
$S := I \setminus B$	44.0	–	–	2.8	–	–	16.3	–	–
Fake-CAM	0.5	67.2	44.1	0.7	2.3	2.8	42.0	9.2	18.9
Grad-CAM	15.0	72.6	52.1	15.3	1.8	6.0	40.4	8.4	19.4
Grad-CAM++	16.5	72.9	53.1	10.6	1.6	4.1	35.2	7.3	17.1
Score-CAM	12.5	71.5	50.5	16.1	2.2	6.3	42.5	8.6	20.8
Ablation-CAM	15.1	72.8	52.1	13.5	1.7	5.6	39.9	7.8	19.0
XGrad-CAM	14.3	72.6	51.4	15.1	1.8	6.0	42.1	8.0	20.1
Layer-CAM	49.2	84.2	74.4	2.7	0.4	1.2	12.7	4.4	7.3
ExPerturbation	43.8	81.6	71.0	7.1	1.4	3.2	18.9	5.6	11.1
Opti-CAM (ours)	**1.4**	**62.5**	**34.8**	**66.3**	**8.7**	**25.8**	**92.5**	**18.6**	**47.1**

Table 2.5: **Bounding box** study. Classification metrics on ImageNet validation set using VGG16. B: ground-truth box used by localization metrics; I: entire image; S: saliency map. AD/AI: average drop/increase (Chattopadhay et al. 2018); AG: average gain (ours); ↓ / ↑: lower / higher is better; bold: best, excluding Fake-CAM.

2.7.2 Weakly Supervised Approach

Several works measure the localization ability of saliency maps, using metrics from the *weakly -supervised object localization* (WSOL) task. While we show that localization of the object and classifier interpretability are not well aligned as tasks, we still provide localization results. We use the *official metric* (OM), *localization error* (LE), *pixel-wise* F_1 *score*, *box accuracy* (BoxAcc) (Choe et al. 2020), standard pointing game (SP) (J. Zhang et al. 2017), *energy pointing game* (EP) (H. Wang, Du, et al. 2019) and *saliency metric* (SM) (Dabkowski et al. 2017) on the ILSVRC2014[10] dataset. The goal of these metrics is to compare the saliency maps with bounding boxes around the object of interest.

We evaluate the localization ability of saliency maps obtained by our Opti-CAM and we compare with other attribution methods quantitatively. Table 2.6 and Table 2.7 report localization metrics on ImageNet. We observe different behavior in different

[10] https://www.image-net.org/challenges/LSVRC/2014/index#

metrics. In particular, Opti-CAM on ResNet and VGG performs best on OM and LE but poorly on the remaining metrics. On transformers, Opti-CAM performs best on OM, LE, F1, and SM.

METHOD	ResNet50							VGG16						
	OM↓	LE↓	F1↑	BA↑	SP↑	EP↑	SM↓	OM↓	LE↓	F1↑	BA↑	SP↑	EP↑	SM↓
Fake-CAM	63.6	54.0	57.7	47.9	99.8	28.5	0.98	64.7	54.0	57.7	47.9	99.8	28.5	1.07
Grad-CAM	72.9	65.8	49.8	**56.2**	69.8	33.3	1.30	71.1	62.3	42.0	54.2	64.8	32.0	1.39
Grad-CAM++	73.1	66.1	**50.4**	**56.2**	69.9	33.1	1.29	70.8	61.9	44.3	55.2	66.2	32.3	1.38
Score-CAM	**72.2**	64.9	49.6	54.5	68.7	32.4	**1.25**	71.2	62.5	**45.3**	**58.5**	**68.2**	33.4	1.40
Ablation-CAM	72.8	65.7	50.2	56.1	69.9	33.1	1.26	71.3	62.6	43.2	56.2	65.7	32.7	1.39
XGrad-CAM	72.9	65.8	49.8	**56.2**	69.8	33.3	1.30	70.8	62.0	41.9	53.5	64.4	31.6	1.41
Layer-CAM	73.1	66.0	50.1	55.5	**70.0**	33.0	1.29	70.5	61.5	28.0	54.7	65.0	32.4	1.45
ExPerturbation	73.6	66.6	37.5	44.2	64.8	**38.2**	1.59	74.1	66.4	37.8	43.3	62.7	**36.1**	1.74
Opti-CAM	**72.2**	**64.8**	47.3	49.2	59.4	30.5	1.34	**69.1**	**59.9**	44.1	51.2	61.4	30.7	**1.34**

Table 2.6: **Localization metrics** on ImageNet validation set. OM: *official metric*; LE: *localization error*; F1: *pixel-wise F_1 score*; BA: box accuracy; SP: standard pointing game; EP: energy pointing game; SM: *saliency metric*. ↓ / ↑: lower / higher is better. Bold: best, excluding Fake-CAM.

METHOD	ViT-B							DeiT-B						
	OM↓	LE↓	F1↑	BA↑	SP↑	EP↑	SM↓	OM↓	LE↓	F1↑	BA↑	SP↑	EP↑	SM↓
Fake-CAM	62.8	54.0	57.7	47.9	99.8	28.6	0.87	61.4	54.0	57.7	47.9	99.8	28.7	0.83
Grad-CAM	79.6	74.3	29.4	45.0	58.1	31.0	3.27	65.5	60.3	44.3	47.2	62.8	30.2	1.20
Grad-CAM++	84.2	80.6	14.8	23.8	51.4	27.3	4.15	70.6	67.2	34.3	43.6	57.7	30.3	2.14
Score-CAM	77.6	71.6	46.0	54.3	**66.1**	33.1	3.14	79.9	76.2	31.9	43.8	**63.4**	32.2	3.14
XGrad-CAM	82.0	76.9	19.6	41.3	52.8	28.5	3.31	82.0	78.4	19.5	44.1	53.4	28.8	3.03
Layer-CAM	70.7	63.9	20.6	50.5	60.7	32.6	1.44	80.2	77.3	17.6	50.8	62.7	35.1	3.15
ExPerturbation	71.5	64.9	35.9	44.6	62.3	**35.3**	1.34	69.9	64.3	36.2	44.2	63.1	**35.5**	1.16
RawAtt	72.4	64.8	18.5	50.4	55.4	31.6	1.68	73.5	68.2	5.9	**48.1**	46.5	27.3	1.91
Rollout	67.6	58.8	36.9	50.7	57.8	30.0	1.16	63.9	57.0	27.8	47.9	36.5	27.2	0.94
TIBAV	70.1	63.1	26.6	**58.8**	**66.1**	35.0	1.23	68.2	62.2	28.1	59.6	64.1	33.5	1.08
Opti-CAM (ours)	**64.4**	**54.6**	**54.5**	48.0	58.2	28.7	**0.98**	**62.3**	**55.1**	**53.9**	48.0	55.1	28.8	**0.84**

Table 2.7: **Localization metrics** with ViT and DeiT on ImageNet validation set. OM: *official metric*; LE: *localization error*; F1: *pixel-wise F_1 score*; BA: box accuracy; SP: standard pointing game; EP: energy pointing game; SM: *saliency metric*. ↓ / ↑: lower / higher is better. Bold: best, excluding Fake-CAM.

Metrics, where Opti-CAM does not perform well, are mostly the ones that penalize saliency maps that are more spread out. For example, SP and EP penalize saliency outside the ground truth bounding box of an object. This is not necessarily a weakness of Opti-CAM, because rather than weakly supervised object localization, the objective here is to explain how the classifier works.

2.8 Ablation study

We perform an ablation study of different choices of the objective function (2.6) and normalization (1.7) of the saliency map.

Normalization function For normalization function n (2.6), we investigate three choices:

$$\text{range}: \quad n(A) := \tfrac{A - \min A}{\max A - \min A} \tag{2.9}$$

$$\text{maximum}: \quad n(A) := \tfrac{A}{\max A} \tag{2.10}$$

$$\text{sigmoid}: \quad n(a_{ij}) := \frac{1}{1 + e^{-a_{ij}}}, \tag{2.11}$$

where a_{ij} is element (i, j) of matrix A. The default is (2.9), normalizing by the range of values in the saliency map, as in Score-CAM (1.7); while (2.10) normalizes by the maximum value and (2.11) by the sigmoid function element-wise.

Objective function We refer to the default definition of F_ℓ^c (2.6) as Mask because it maximizes the logit for the masked image. We also consider an alternative definition of objective function F_ℓ^c, which encourages the masked version to preserve the prediction of original image:

$$F_\ell^c(\mathbf{x};\hat{)} := -\left| g_c(f(\mathbf{x})) - g_c(f(\mathbf{x} \odot n({}^{(}S_\ell(\mathbf{x};\hat{)})))) \right|. \tag{2.12}$$

This function is named Diff as it minimizes the difference of logits between the masked and the original image.

Results Table 2.8 shows classification metrics for the different choices of Opti-CAM, as well as comparison to other methods for reference, for the small subset of ImageNet validation set.

We observe that the choice of normalization function has little effect overall and Sigmoid offers lower performance. Note that the minimum value of saliency maps is often zero or close to zero: *Saliency maps are non-negative as convex combinations of non-negative feature maps (2.4)*. In contrast, the choice of loss function has more impact on performance, and we observe that Mask (2.6) is superior on all cases.

METHOD	F^c_ℓ	n	AD↓	AG↑	AI↑
Fake-CAM			0.5	0.7	42.1
Grad-CAM			15.0	15.3	40.4
Grad-CAM++			16.5	10.6	35.2
Score-CAM			12.5	16.1	42.6
Ablation-CAM			15.1	13.5	39.9
XGrad-CAM			14.3	15.1	42.1
Layer-CAM			49.2	2.7	12.7
ExPerturbation			43.8	7.1	18.9
Opti-CAM	Mask (2.6)	Range (2.9)	**1.4**	**66.3**	**92.5**
	Diff (2.12)	Range (2.9)	7.1	18.5	54.9
Opti-CAM	Mask (2.6)	Max (2.10)	1.6	66.2	90.3
	Diff (2.12)	Max (2.10)	6.8	17.8	54.5
Opti-CAM	Mask (2.6)	Sigmoid (2.11)	5.0	18.3	57.5
	Diff (2.12)	Sigmoid (2.11)	6.5	10.0	45.3

Table 2.8: **Ablation study** using VGG16 on 1000 images of ImageNet validation set. AD/AI: average drop/increase (Chattopadhay et al. 2018); AG: average gain (ours); ↓ / ↑: lower / higher is better; bold: best, excluding Fake-CAM.

2.9 Discussion

Opti-CAM is constructed following the definition of *CAM-based* saliency maps. In particular, according to Equation 1.3, we optimize the variable w^c_k to construct a saliency map $S^c_\ell(\mathbf{u})$, maximizing the predicted probability of the explanation obtained by performing element-wise multiplication with an input image.

Classifier-Centric Explanations As a consequence of optimizing prediction probability of explanation maps, our approach highlights the salient regions in the image. We observe in CNNs, salient information is spread across the input image and not often centered within the object of interest. These models learn biases within data, and use context to construct a prediction. Studies have demonstrated instances in fine-grained image classification, where the model learns the background of images instead of the object of interest (Petryk et al. 2022). Classifier-centric explanations can demonstrate situations where this is the case. Moreover, classifier-centric explanations do not leave space for human interpretations about the inference process of a model. Thus removing one factor of bias towards interpretation.

Localization Properties A direct consequence of generating classifier-centric explanations, is a trade-off in localization properties. Compared to current attribution methods, Opti-CAM fails in this regard. Preliminary studies suggesting the evaluation of saliency maps based on localization, can be traced back to the work by (Shetty et al. 2019, B. Zhou, Bau, et al. 2019, Rao et al. 2022). This failure however is desired.

Furthermore, it highlights that context is important towards prediction of a model; and as such, localization is an ill-fitted requirement to assess interpretability.

Computational Complexity Opti-CAM generates an optimized saliency map for every image. However, in comparison to current high-performing *CAM* methods, the trade-off between complexity and performance favors our approach. In detail, methodologies such as *Ablation-CAM* and *Score-CAM* require as many forward passes as the number of channels in the target layer of interest. In contrast, our approach requires as much as a hundred optimization steps. Additionally, these optimization steps are not as complex as a complete forward pass through the network: our optimization objective requires forwarding the product of the optimization variable, with the feature maps from our target layer until the classifier. Therefore, Opti-CAM requires less memory resources; as well as being faster in running time.

Average Gain Current classification/recognition metrics are not robust and complete, to differentiate interpretability properties of different approaches. This is demonstrated in the work of (Poppi et al. 2021) where Fake-CAM achieves almost perfect AD but fails completely on AI. Average Gain is designed to address this remark. In particular, Average Gain acts as the complement of AD: we measure the positive impact that an explanation poses using Average Gain. Conversely, we observe that AI is a metric that on itself does not answer to anything in particular: on a real world application we do not care in how many instances the explanation map is better than an input image; we focus on the effect an explanation has over the classifier. This efficiently covered with AD and AG

2.10 Conclusion

In this chapter we propose Opti-CAM, a CAM-based methodology to generate saliency maps highlighting the most relevant image patches describing a classification according to a classifier. Our approach builds upon the definition of CAM attributions to optimize attribution map predicted probabilities.

Opti-CAM combines ideas of different saliency map generation methods, which are masking-based and CAM-based. Our method optimizes the saliency map at inference given a single input image. It does not require any additional data or training any other network, which would need interpretation too.

While Opti-CAM crafts a saliency map in the image space, it does not need any regularization. This is because the saliency map is expressed as a convex combination of feature maps, and we only optimize one vector over the feature dimensions. The underlying assumption is that of all CAM-based methods: feature maps contain activations at all regions that are of interest for the classes that are present. Opti-CAM is more expensive than non-iterative gradient-based methods but as fast or faster than gradient-free methods that require as many forward passes as channels.

Opti-CAM brings impressive performance improvement over the state of the art according to the most important classification metrics on several datasets. The saliency maps are more spread out compared with those of the competition, attending to larger parts of the object, multiple instances and background context, which may be helpful in classification.

Our new classification metric AG aims to be paired AD as a replacement of AI and resolves a long-standing problem in evaluating attribution methods, without further increasing the number of metrics. We provide strong evidence supporting that the use of ground-truth object bounding boxes for localization is not necessarily optimal in evaluating the quality of a saliency map, because the primary objective is to explain how a classifier works.

3 CA-Stream: Attention-based pooling for interpretable image recognition

Table of contents

3.1 Introduction 77
3.2 Cross Attention 78
3.3 Cross Attention Stream 80
3.4 Experimental Set-Up 81
3.5 Qualitative Evaluation 83
3.6 Quantitative Evaluation 84
3.6.1 Classification accuracy 86
3.6.2 Ablation 87
3.7 Discussion 90
3.8 Conclusion 91

3.1 Introduction

Another way to approach interpretability can be pointed towards the current advances in recognition in general. In particular, with the introduction of the transformer architecture (Vaswani et al. 2017) a switch in the paradigm occurred where the best performing architectures contain the self-attention module as a building block. Moreover, with the proposal of the Vision Transformer (Dosovitskiy et al. 2020) transformers were adopted into computer vision. This module gained prominence as it allowed models to push the boundaries in existing benchmarks. This led to an expansion with models such as Swin-T (Ze Liu et al. 2021), LeViT Graham et al. 2021. Conversely, hybrid architectures combining ideas from both CNN and transformers can be observed like Conformer (Peng et al. 2021), Patchconvnet (Touvron, Cord, El-Nouby, et al. 2021), while on another hand a modernization of CNNs in the shape of ConvNeXt (Zhuang Liu et al. 2022) drew inspiration from these models, whilst addressing their shortcomings in downstream tasks.

Although these models have pushed visual recognition to new frontiers, their interpretable properties still require further exploration. Conventional interpretability methodologies for CNNs do not translate properly into their domain, all the while the explanations obtained from these methods (Abnar et al. 2020) do not appear to have a proper evaluation protocol, resulting in research aimed at improved visualizations and their assesment (Chefer et al. 2021).

In this chapter we study the correlation between CAM and one such attention visualization proposal that is the raw attention found in the classification (CLS) token (Devlin et al. 2018). In particular, we note that self attention is defined for all patch tokens including CLS, however we can generate cross attention between this token and the feature maps found at any given depth of a CNN; this being expressed in via linear combination of feature maps with this token, ultimately resembling a class agnostic CAM. As an extension of this, we propose the inclusion of a cross-attention module used to train this token as a replacement of GAP (M. Lin et al. 2013), onto already trained models boosting both their interpretable properties, while maintaining recognition performance.

3.2 Cross Attention

Let matrix $F_\ell \in \Re^{p_\ell \times d_\ell}$ be a reshaping of feature tensor $\mathbf{F}_\ell$ at layer ℓ, where $p_\ell := w_\ell h_\ell$ is the number of patch tokens without CLS, and let $\mathbf{q}_\ell \in \Re^{d_\ell}$ be the CLS token embedding at layer ℓ. By focusing on the *cross attention* only between the CLS (query) token $\mathbf{q}_\ell$ and the patch (key) tokens F_ℓ and by ignoring projections W_Q, W_K, W_V for simplicity, attention A (1.1) is now a $1 \times p_\ell$ matrix that can be written as a vector $\mathbf{a} \in \Re^{p_\ell}$

$$\mathbf{a} = A^\top = \operatorname{softmax}\left(\frac{F_\ell \mathbf{q}_\ell}{\sqrt{d_\ell}}\right). \tag{3.1}$$

Here, $F_\ell \mathbf{q}_\ell$ expresses the pairwise similarities between the global CLS feature $\mathbf{q}_\ell$ and the local patch features F_ℓ. Now, by replacing $\mathbf{q}_\ell$ by an arbitrary vector $\mathbf{a}lpha \in \Re^{d_\ell}$ and by writing the feature matrix as $F_\ell = (\mathbf{f}_\ell^1 \dots \mathbf{f}_\ell^{d_\ell})$ where $\mathbf{f}_\ell^k = \operatorname{vec}(F_\ell^k) \in \Re^{p_\ell}$ for channel k, attention (3.1) becomes

$$\mathbf{a} = h_\ell(F_\ell \alpha) = h_\ell\left(\sum_k \alpha_k \mathbf{f}_\ell^k\right). \tag{3.2}$$

This takes the same form as (1.3), with feature maps F_ℓ^k being vectorized into $\mathbf{f}_\ell^k$ and the activation function is defined as $h_\ell(\mathbf{x}) = \operatorname{softmax}(\mathbf{x}/\sqrt{d_\ell})$. Eq. (3.2) is visualized in Figure 3.1. We thus observe the following.

> *Pairwise similarities between one query and all patch token embeddings in cross attention are the same as a linear combination of feature maps in CAM-based saliency maps, where the weights are determined by the elements of the query.*

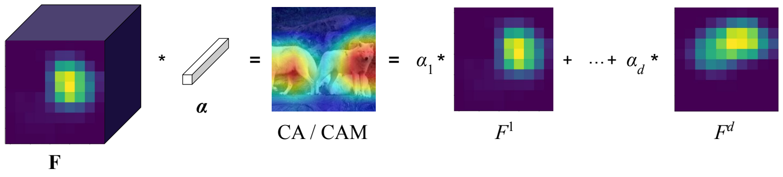


Figure 3.1: **Visualization of eq. (3.2).** On the left, a feature tensor $\mathbf{F} \in \Re^{w \times h \times d}$ is multiplied by the vector $\boldsymbol{\alpha} \in \Re^d$ in the channel dimension, like in 1×1 convolution, where $w \times h$ is the spatial resolution and d is the number of channels. This is *cross attention* (CA) (Dosovitskiy et al. 2020) between the query $\boldsymbol{\alpha}$ and the key $\mathbf{F}$. On the right, a linear combination of feature maps $F^1, \dots, F^d \in \Re^{w \times h}$ is taken with weights $\alpha_1, \dots, \alpha_d$. This is a *class activation mapping* (CAM) (B. Zhou, Khosla, et al. 2016) with class agnostic weights. Eq. (3.2) expresses the fact that these two quantities are the same, provided that $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d)$ and $\mathbf{F}$ is reshaped as $F = (\mathbf{f}^1 \dots \mathbf{f}^d) \in \Re^{p \times d}$, where $p = wh$ and $\mathbf{f}^k = \text{vec}(F^k) \in \Re^p$ is the vectorized feature map of channel k.

As it stands, one difference between (1.3) and (3.2) is that (3.2) is class agnostic, although it could be extended by using one query (weight) vector per class. For simplicity, we choose the class agnostic form in the following.

Pooling, or masking We are thus motivated to integrate an attention mechanism into any network such that making a prediction and explaining (localizing) it are inherently connected. In particular, considering cross attention only between CLS and patch tokens (3.1), equation (1.2) becomes

$$\text{CA}_\ell(\mathbf{q}_\ell, F_\ell) := F_\ell^\top \mathbf{a} = F_\ell^\top h_\ell(F_\ell \mathbf{q}_\ell) \in \Re^{d_\ell}. \tag{3.3}$$

By writing the transpose of feature matrix as $F_\ell^\top = (\boldsymbol{\phi}_\ell^1 \dots \boldsymbol{\phi}_\ell^{p_\ell})$ where $\boldsymbol{\phi}_\ell^i \in \Re^{d_\ell}$ is the feature of patch i, this is a weighted average of the local patch features $F_\ell^\top$ with attention vector $\mathbf{a} = (a_1, \dots, a_{p_\ell})$ expressing the weights:

$$\text{CA}_\ell(\mathbf{q}_\ell, F_\ell) := F_\ell^\top \mathbf{a} = \sum_i a_i \boldsymbol{\phi}_\ell^i. \tag{3.4}$$

We can think of it as a feature *reweighting* or *soft masking* in the feature space, followed by GAP.

Now, considering that $\mathbf{a}$ is obtained exactly as CAM-based saliency maps (3.2), this operation is similar to occlusion (masking)-based methods (Vitali et al. 2018; R. C. Fong et al. 2017; R. Fong et al. 2019; Schulz et al. 2020; Ribeiro et al. 2016; H. Wang, Du, et al. 2019; H. Zhang et al. 2023) and evaluation metrics (Chattopadhay et al. 2018;

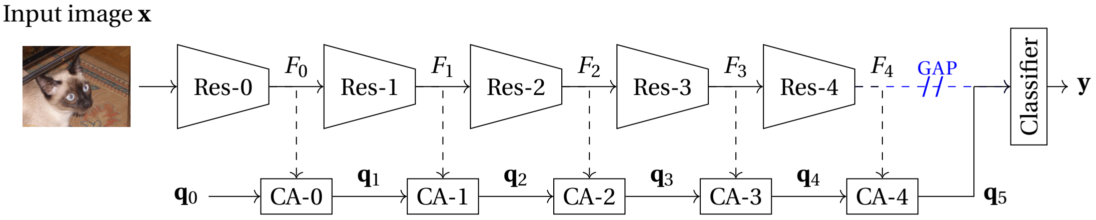


Figure 3.2: **Cross Attention Stream (CA-Stream) applied to ResNet-based architectures.** Given a network f, we replace global average pooling (GAP) by a learned, attention-based pooling mechanism implemented as a stream in parallel to f. The feature tensor $F_\ell \in \Re^{p_\ell \times d_\ell}$ (*key*) obtained by stage Res-ℓ of f interacts with a CLS token (*query*) embedding $\mathbf{q}_\ell \in \Re^{d_\ell}$ in block CA-ℓ, which contains cross attention (3.3) followed by a linear projection (3.7) to adapt to the dimension of $F_{\ell+1}$. Here, p_ℓ is the number of patches (spatial resolution) and d_ℓ the embedding dimension. The query is initialized by a learnable parameter $\mathbf{q}_0 \in \Re^{d_0}$, while the output $\mathbf{q}_5$ of the last cross attention block is used as a global image representation into the classifier.

Vitali et al. 2018), where a CAM-based saliency map is commonly used to mask the input image.

We thus observe the following:

> *Attention-based pooling is a form of feature reweighting or soft masking in the feature space followed by* GAP*, where the weights are given by a class agnostic CAM-based saliency map.*

3.3 Cross Attention Stream

Motivated by the observations above, we design a *Cross Attention Stream* (*CA-Stream*) in parallel to any network. It takes input features at key locations of the network and uses cross attention to build a global image representation and replace GAP before the classifier. An example is shown in Figure 3.2, applied to a ResNet-based architecture.

Architecture More formally, given a network f, we consider points between blocks of f where critical operations take place, such as change of spatial resolution or embedding dimension, *e.g.* between stages for ResNet. We decompose f at these points as

$$f = g \circ \text{GAP} \circ f_L \circ \cdots \circ f_0 \tag{3.5}$$

such that features $F_\ell \in \Re^{p_\ell \times d_\ell}$ of layer (stage) ℓ are initialized as $F_{-1} = \mathbf{x}$ and updated according to

$$F_\ell = f_\ell(F_{\ell-1}) \tag{3.6}$$

for $0 \leq \ell \leq L$. The last layer features F_L are followed by GAP and $g : \Re^{d_L} \to \Re^C$ is the classifier, mapping to the logit vector $\mathbf{y}$. As in (1.3), p_ℓ is the number of patch tokens and d_ℓ the embedding dimension of stage ℓ.

In parallel, we initialize a classification token embedding as a learnable parameter $\mathbf{q}_0 \in \Re^{d_0}$ and we build a sequence of updated embeddings $\mathbf{q}_\ell \in \Re^{d_\ell}$ along a stream that interacts with F_ℓ at each stage ℓ. Referring to the global representation $\mathbf{q}_\ell$ as *query* or CLS and to the local image features F_ℓ as *key* or patch embeddings, the interaction consists of cross attention followed by a linear projection $W_\ell \in \Re^{d_{\ell+1} \times d_\ell}$ to account for changes of embedding dimension between the corresponding stages of f:

$$\mathbf{q}_{\ell+1} = W_\ell \cdot \text{CA}_\ell(\mathbf{q}_\ell, F_\ell), \tag{3.7}$$

for $0 \leq \ell \leq L$, where CA_ℓ is defined as in (3.3).
Image features $F_0, \ldots, F_L$ do not change by injecting our CA-Stream into network f. However, the final global image representation and hence the prediction do change. In particular, at the last stage L, $\mathbf{q}_{L+1}$ is used as a global image representation for classification, replacing GAP over F_L. The final prediction is $g(\mathbf{q}_{L+1}) \in \Re^C$. Unlike GAP, the weights of different image patches in the linear combination are non-uniform, enhancing the contribution of relevant patches in the prediction.

Training In this sense, the network f is pretrained and remains frozen while we learn the parameters of our CA-Stream on the same training set as one used to train f. The classifier is kept frozen too. Referring to (3.5), $f_0, \ldots, f_L$ and g are fixed, while GAP is replaced by learned weighted averaging, with the weights obtained by the CA-Stream.

Inference As it stands, CA-Stream is a modification of the baseline architecture, *i.e.*, an attention-based pooling mechanism that replaces GAP during inference, to enhance the contribution of relevant image regions in the prediction. We are interested in investigating the interpretability properties of this modification. We therefore employ existing post-hoc, CAM-based interpretability methods to generate saliency maps with both baseline GAP and CA-Stream. Interpretability metrics are compared as well as classification accuracy.

3.4 Experimental Set-Up

We evaluate the interpretability and recognition capabilities of our approach. In particular, we generate explanations following current state-of-the art post-hoc interpretability methods derived from CAM (B. Zhou, Khosla, et al. 2016). We compare the properties of the backbone network f with and without our CA-Stream, where f is pretrained and fixed.

Training We train and evaluate our models on the ImageNet ILSVRC-2012 dataset (Russakovsky et al. 2015), on the training and validation splits respectively. Thus, we experiment with ResNet-based architectures (He, X. Zhang, et al. 2016) such as ResNet-18 and ResNet-50, and ConvNeXt based architectures (Zhuang Liu et al. 2022) such as ConvNeXt-Small and ConvNeXt-Base. We aim at learning our CA-Stream, generating a CLS token that interacts with feature maps at different stages of network f, to serve as an attention-based pooling mechanism in order to interpret the predictions of f. Therefore, we experiment with pretrained models[1], that we keep frozen while the parameters of the CA-Stream are optimized. Moreover, we present experiments on the bird dataset: CUB-200-2011 Wah et al. 2011 and on PASCAL VOC 2012 dataset Everingham, Eslami, et al. 2015. Here the ResNet-50 network is fine-tuned to these dataset as baseline. Then, our CA-Stream is learned as for ImageNet.

Implementation Details Following the training recipes from the pytorch models [2], we choose the ResNet protocol given its simplicity. Thus, we train over 90 epochs with SGD optimizer with momentum 0.9 and weight decay 10^{-4}. We start our training with a learning rate of 0.1 and decrease it every 30 epochs by a factor of 10. Our models are trained on 8 V100 GPUs with a batch size 32 per GPU, thus global batch size 256.

We follow the same protocol for both ResNet and ConvNeXt, though a different protocol might lead to improvements on ConvNeXt.

Evaluation We employ existing post-hoc interpretability methods to generate saliency maps with and without CA-Stream and compare interpretability metrics as well as classification accuracy. Regarding interpretability methods, we use Grad-CAM (Selvaraju et al. 2016), Grad-CAM++ (Chattopadhay et al. 2018) and ScoreCAM (H. Wang, Du, et al. 2019). We note that the evaluation is performed on the entire validation set, unlike the previous approaches.

Following Opti-CAM (Chapter 2), we use a number of classification metrics for interpretability. In particular, we consider the changes in predictive power measured by *average drop* (AD) (Chattopadhay et al. 2018) and *average gain* (AG) section 2.4, the proportion of better explanations measured by *average increase* (AI) (ibid.) and the impact of different extent of masking measured by *insertion* (I) and *deletion* (D) (Vitali et al. 2018).

[1] https://pytorch.org/vision/0.8/models.html

[2] https://github.com/pytorch/vision/tree/main/references/classification

3.5 Qualitative Evaluation

We show saliency maps obtained by different interpretability methods using either GAP or CA-Stream, as well as the class-agnostic raw attention coming from our CA-Stream, see Figure 3.3.

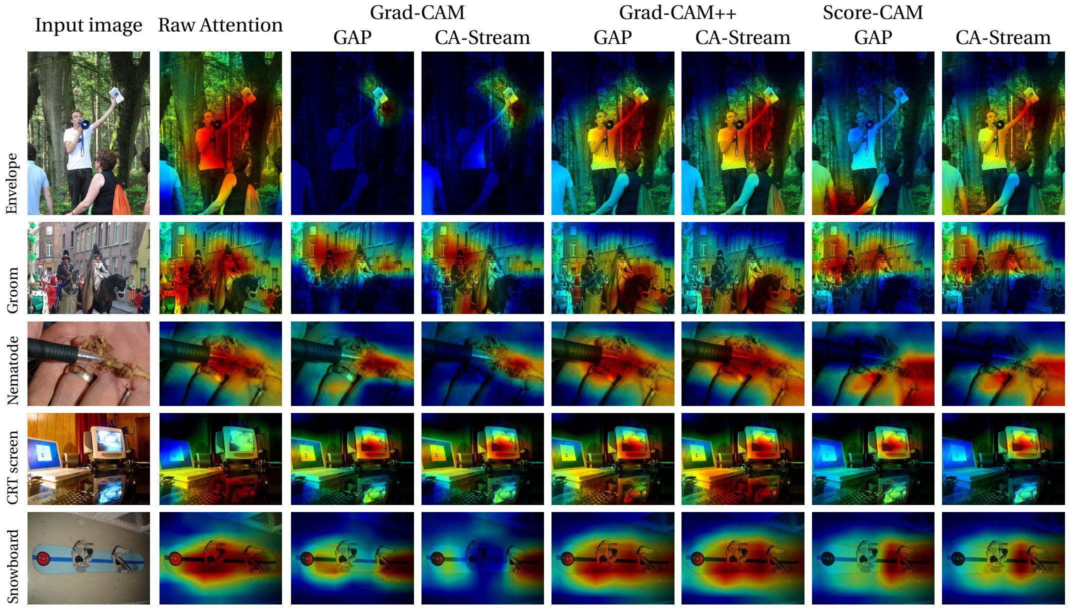


Figure 3.3: **Comparison of saliency maps** generated by different CAM-based methods, using GAP and our CA-Stream, on ImageNet images. The raw attention is the one used for pooling by CA-Stream.

We observe that the raw attention focuses on objects of interest in the images. In general, saliency maps obtained with CA-Stream are similar but tend to cover larger regions of the object or more instances compared with GAP.

Indeed, the differences in saliency maps should not be large, as both methods share the same features maps F_ℓ^k and only the weight coefficients α_k^c differ. Despite the small differences, the following quantitative results show that CA-Stream has a significant impact on the interpretability metrics.

In addition, Figure 3.4 shows examples of images from the MIT 67 Scenes dataset (Quattoni et al. 2009) along with raw attention maps obtained by CA-Stream. These images come from four classes that do not exist in ImageNet and the network sees them at inference for the first time. Nevertheless, the attention maps focus on objects of interest in general.

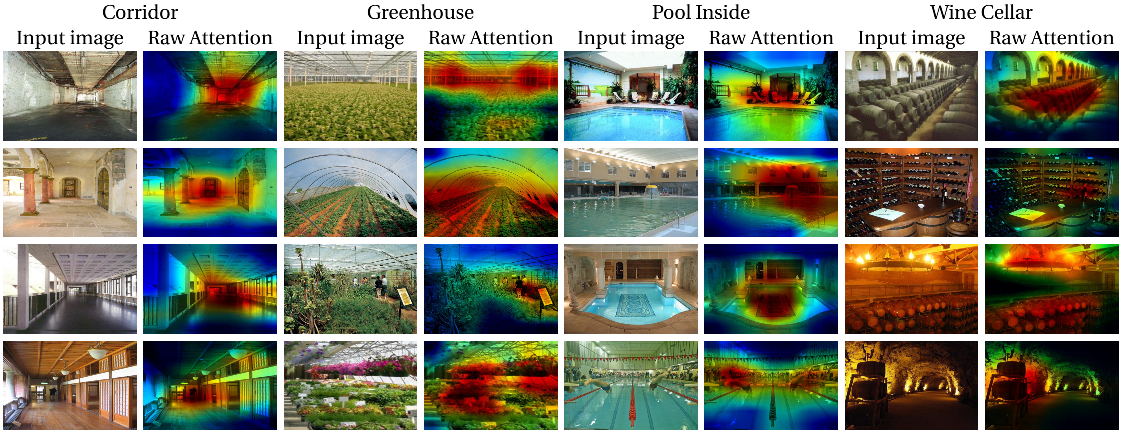


Figure 3.4: **Raw attention maps** obtained from our CA-Stream on images of the MIT 67 Scenes dataset (Quattoni et al. 2009) on classes that do not exist in ImageNet. The network sees them at inference for the first time.

3.6 Quantitative Evaluation

Here we measure the effect of employing our CA-Stream approach to pool features *vs.* the baseline GAP on the faithfulness of explanations, using classification metrics for interpretability. Results are reported in Table 3.1 for ImageNet and Table 3.2 for CUB and Pascal VOC.

Network	Method	Pool	AD↓	AG↑	AI↑	I↑	D↓
ResNet-18	Grad-CAM	GAP	17.64	12.73	41.21	63.13	**10.66**
		CA	**16.99**	**17.22**	**44.95**	**65.94**	10.68
	Grad-CAM++	GAP	19.05	11.16	37.99	62.80	**10.75**
		CA	**19.02**	**14.76**	**40.82**	**65.53**	10.82
	Score-CAM	GAP	13.64	12.98	44.53	62.56	**11.37**
		CA	**11.53**	**18.12**	**50.32**	**65.33**	11.51
ResNet-50	Grad-CAM	GAP	13.04	17.56	44.47	72.57	**13.24**
		CA	**12.54**	**22.67**	**48.56**	**75.53**	13.50
	Grad-CAM++	GAP	**13.79**	15.87	42.08	72.32	**13.33**
		CA	13.99	**19.29**	**44.60**	**75.21**	13.78
	Score-CAM	GAP	8.83	17.97	48.46	71.99	**14.31**
		CA	**7.09**	**23.65**	**54.20**	**74.91**	14.68
ConvNeXt-S	Grad-CAM	GAP	42.99	1.69	12.60	48.42	**30.12**
		CA	**22.09**	**14.91**	**32.65**	**84.82**	43.02
	Grad-CAM++	GAP	56.42	1.32	10.35	48.28	**33.41**
		CA	**51.87**	**9.40**	**20.55**	**84.28**	52.58
	Score-CAM	GAP	74.79	1.29	10.10	47.40	**38.21**
		CA	**64.21**	**8.81**	**18.96**	**82.92**	57.46
ConvNeXt-B	Grad-CAM	GAP	33.72	2.43	15.25	52.85	**29.57**
		CA	**19.45**	**13.96**	**32.89**	**86.38**	45.29
	Grad-CAM++	GAP	**34.01**	2.37	15.60	52.83	**29.17**
		CA	36.69	**8.00**	**21.95**	**85.39**	53.42
	Score-CAM	GAP	43.55	2.23	15.67	50.96	**39.49**
		CA	**23.51**	**11.04**	**27.35**	**83.41**	60.53

Table 3.1: **Interpretability metrics** of CA-Stream *vs.* baseline GAP for different networks and interpretability methods on ImageNet.

Table 3.1 shows that for different networks, CAM-based interpretability methods and dataset, CA-Stream provides consistent improvements over GAP in terms of AD, AG, AI and I metrics, while performing lower on D.

Deletion has raised concerns in previous works (section 2, Chefer et al. 2021). Indeed, it gradually replaces pixels by black, unlike insertion which starts from a blurred image. This poses the problem of *out-of-distribution* (OOD) data (Gomez et al. 2022, Hase et al. 2021, Qiu et al. 2021), possibly introducing bias related to the shape of black regions (Rong et al. 2022). Moreover, non-spread saliency maps tend to perform better (as seen on section 2), which is likely the reason for lower performance.

		Pascal VOC 2012					CUB-200-2011				
		Pooling				mAP↑	Pooling				Acc↑
		GAP				78.32	GAP				76.96
		CA				78.35	CA				75.90
		Interpretability Metrics									
Method	Pooling	AD↓	AG↑	AI↑	I↑	D↓	AD↓	AG↑	AI↑	I↑	D↓
Grad-CAM	GAP	**12.61**	9.68	27.88	**89.10**	59.39	10.87	10.29	45.81	65.71	**6.17**
	CA	12.77	**15.46**	**34.53**	88.53	**59.16**	**10.44**	**17.61**	**53.54**	**74.60**	6.56
Grad-CAM++	GAP	**12.25**	9.68	27.62	**89.34**	54.23	11.35	9.68	44.32	65.64	**5.92**
	CA	12.28	**16.76**	**34.87**	89.02	**53.34**	**11.01**	**16.50**	**51.63**	**74.64**	6.21
Score-CAM	GAP	14.8	6.76	36.41	71.10	**39.95**	9.05	10.62	48.90	65.58	5.94
	CA	**10.96**	**21.35**	**43.82**	**89.21**	51.44	**6.37**	**19.50**	**60.41**	**74.22**	**2.14**

Table 3.2: **Recognition and Interpretability metrics:** Evaluation between CA-Stream *vs.* baseline GAP for ResNet-50 on CUB and Pascal dataset.

Results on CUB in Table 3.2 show that our CA-Stream consistently provides improvements when the model is finetuned on a smaller fine-grained dataset.

Regarding Pascal VOC, the results for Score-CAM are similar to the ones on ImageNet and CUB, with consistent improvements on all metrics but Deletion. However, Grad-CAM and Grad-CAM++ only provide improvements on Average Gain and Average Increase. Average Drop, Insertion and Deletion are very similar. In fact, Pascal VOC is a multi-class dataset and our CA-Stream is class agnostic. Thus, the attention-based pooling is the same for different class for a given image, which reduces the benefit of our CA-Stream.

It is also interesting to observe the performance of Score-CAM, as it computes channel weights α_k^c in (1.3) without using gradients. In gradient-based methods, channel weights are modified by CA-Stream due to modified backward gradient flow to features through cross attention blocks rather than GAP. In Score-CAM however, channel weights are only modified in the forward class probabilities computation, due to attention.

3.6.1 Classification accuracy

Classification accuracy, number of parameters and GFLOPs for both our CA-Stream and the baselines are reported in Table 3.3.

Accuracy and Parameters					
Network	Pool	GFLOPs	#Param	Param%	Acc↑
ResNet-18	GAP CA	3.648 3.652	11.69M 12.13M	3.71	67.28 67.54
ResNet-50	GAP CA	8.268 8.288	25.56M 32.53M	27.27	74.55 74.70
ConvNeXt-S	GAP CA	17.395 17.400	50.22M 51.20M	1.95	83.26 83.14
ConvNeXt-B	GAP CA	30.747 30.753	88.59M 90.33M	1.96	83.72 83.51

Table 3.3: **Accuracy and parameters** of CA-Stream *vs.* baseline GAP for different networks and interpretability methods on ImageNet. #Param: total parameters; Param%: percentage of CA-Stream parameters relative to backbone.

By adding our CA-Stream to the network, classification remains on par with the baseline. Importantly, the network including the classifier remains frozen and the features used for the global image representation remain fixed, meaning that any change in accuracy is due to the attention-based pooling mechanism.

We further report the number of GFLOPs for one forward pass and the parameters count of both methods. Our CA-Stream has little computation cost and the parameter overhead depends on the embedding dimension because of projection W_ℓ in (3.7) and is small in general, except for ResNet-50. Thus, with small overhead in resources, CA-Stream achieves superior explanations of the classifier predictions, while maintaining accuracy.

3.6.2 Ablation

We conduct ablation experiments on ResNet50 because of its modularity and ease of modification. We investigate the effect of the cross attention block design, the placement of the CA-Stream relative to the backbone network.

Cross attention block design Following transformers (Vaswani et al. 2017, Dosovitskiy et al. 2020), it is possible to add more layers in the cross attention block. We consider a variant referred to as PROJ→CA, which uses linear projections $W_\ell^K, W_\ell^V \in \Re^{d_\ell \times d_\ell}$ on the key and value

$$\textsc{ca}_\ell(\mathbf{q}_\ell, F_\ell) := (F_\ell W_\ell^V)^\top h_\ell(F_\ell W_\ell^K \mathbf{q}_\ell) \in \Re^{d_\ell}, \tag{3.8}$$

while equation (3.7) remains.

Results are reported in Table 3.4. We observe that the stream made of vanilla CA blocks (3.3) offers slightly better accuracy than projections (3.8), while having less parameters. We also note that most of the computation takes place in the last residual stages, where the channel dimension is the largest. To keep our design simple, we choose the vanilla solution without projections (3.3) by default.

BLOCK TYPE	#PARAMS	ACCURACY
CA	6.96M	74.70
PROJ→CA	18.13M	74.41

Table 3.4: **Different cross attention block design for CA-Stream.** Classification accuracy and parameters using ResNet-50 on ImageNet. #PARAM: parameters of CA-Stream only.

CA-Stream placement To validate the design of CA-Stream, we measure the effect of its depth on its performance *vs.* the baseline GAP in terms of both classification accuracy / number of parameters and classification metrics for interpretability. In particular, we place the stream in parallel to the network f, starting at stage ℓ and running through stage L, the last stage of f, where $0 \leq \ell \leq L$. Results are reported in Table 3.5.

From the interpretability metrics as well as accuracy, we observe that stream configurations that allow for iterative interaction with the network features obtain the best performance, although the effect of stream placement is small in general. In many cases, the lightest stream of only one cross attention block ($S_4 - S_4$) is inferior to options allowing for more interaction. Since starting the stream at early stages has little effect on the number of parameters and performance is stable, we choose to start the stream in the first stage ($S_0 - S_4$) by default.

Class-specific CLS As discussed in section 3.2, the formulation of single-query cross attention as a CAM-based saliency map (1.1) is class agnostic (single channel weights α_k), whereas the original CAM formulation (1.3) is class specific (channel weights α_k^c for given class of interest c). Here we consider a class specific extension of CA-Stream using one query vector per class. In particular, the stream is initialized by one learnable parameter $\mathbf{q}_0^c$ per class c, but only one query (CLS token) embedding is forwarded along the stream. At training, c is chosen according to the target class label, while at inference, the class predicted by the baseline classifier is used instead.

Results are reported in Table 3.6. We observe that the class specific representation for CA-Stream provides no improvement over the class agnostic representation, despite the additional complexity and parameters. We thus choose the class agnostic representation by default. The class specific approach is similar to (Touvron, Cord, El-Nouby, et al. 2021) in being able to generate class specific attention maps, although no fine-tuning is required in our case.

Accuracy and Parameters				
	Placement	CLS dim	#Param	Acc↑
	$S_0 - S_4$	64	6.96M	**74.70**
	$S_1 - S_4$	256	6.95M	74.67
	$S_2 - S_4$	512	6.82M	74.67
	$S_3 - S_4$	1024	6.29M	74.67
	$S_4 - S_4$	2048	4.20M	74.63

Interpretability Metrics						
Method	Placement	AD↓	AG↑	AI↑	I↑	D↓
Grad-CAM	$S_0 - S_4$	**12.54**	**22.67**	48.56	75.53	13.50
	$S_1 - S_4$	12.69	22.65	48.31	75.53	13.41
	$S_2 - S_4$	**12.54**	21.67	**48.58**	75.54	13.50
	$S_3 - S_4$	12.69	22.28	47.89	**75.55**	13.40
	$S_4 - S_4$	12.77	20.65	47.14	74.32	**13.37**
Grad-CAM++	$S_0 - S_4$	13.99	19.29	44.60	75.21	13.78
	$S_1 - S_4$	13.99	19.29	44.62	75.21	13.78
	$S_2 - S_4$	13.71	**19.90**	**45.43**	75.34	13.50
	$S_3 - S_4$	13.69	19.61	45.04	**75.36**	13.50
	$S_4 - S_4$	**13.67**	18.36	44.40	74.19	**13.30**
Score-CAM	$S_0 - S_4$	**7.09**	23.65	54.20	74.91	14.68
	$S_1 - S_4$	**7.09**	23.65	54.20	74.92	14.68
	$S_2 - S_4$	**7.09**	**23.66**	**54.21**	74.91	14.68
	$S_3 - S_4$	7.74	23.03	52.92	**74.97**	14.65
	$S_4 - S_4$	7.52	19.45	50.45	74.19	**14.46**

Table 3.5: **Effect of stream placement** on accuracy, parameters and interpretability metrics for ResNet-50 on ImageNet. $S_\ell - S_L$: CA-Stream runs from stage ℓ to L (last); #Param: parameters of CA-Stream only.

Accuracy and Parameters			
	Representation	#Param	Acc↑
	Class agnostic	32.53M	74.70
	Class specific	32.59M	74.68

Interpretability Metrics						
Method	Representation	AD↓	AG↑	AI↑	I↑	D↓
Grad-CAM	Class agnostic	12.54	22.67	48.56	75.53	13.50
	Class specific	12.53	22.66	48.58	75.54	13.50
Grad-CAM++	Class agnostic	13.99	19.29	44.60	75.21	13.78
	Class specific	13.99	19.28	44.62	75.20	13.78
Score-CAM	Class agnostic	7.09	23.65	54.20	74.91	14.68
	Class specific	7.08	23.64	54.15	74.99	14.53

Table 3.6: **Effect of class agnostic *vs.* class specific representation** on accuracy, parameters and interpretability metrics of CA-Stream for ResNet-50 and different interpretability methods on ImageNet. #Param: parameters of CA-Stream only.

3.7 Discussion

Class Agnostic Pooling On section 3.2 the groundwork relating to Cross Attention was established. In particular, the [CLS] token is able to capture class specific information, or the information relating to the learned class. In our approach, we opt to utilize the class agnostic representation, we specifically want the information leading to top-1 predictions. Nevertheless, the implementation and usage of the class-specific [CLS] token is still interesting, it allows for the visualization of multiple instances of different classes in one image, without any extensive computational requirements such as forward-backward passes, and learning.

Computational Complexity Additionally, our approach is designed to be efficient in terms of computational cost. Its inclusion does not result in a significant increase in parameter count for most models, excluding ResNet-50. This results from an increase in *Block Expansion* values. This parameter acts as a multiplier controlling the size of filter depth across different stages of this architecture. In detail, when we switch from ResNet-18 to ResNet 50; this value increases from 1 in the former, to 4 in the latter. Our approach introduces cross attention layers on stages where a change in feature map depth occurs; therefore a sharp increase of Block Expansion value leads directly to an increase of Cross-Attention Stream size. We acknowledge that in deeper ResNet variants this overhead is similar to that encountered in ConvNeXt architectures.

Saliency Maps Visualization Our approach is not designed as a novel attribution method, instead our [CLS] representation improves predictive power. In particular, when we generate attributions, the computation procedure is mostly the same between GAP and [CLS], with the biggest difference coming from the pooling method used. Since we obtain fairly similar recognition values, we argue that these two representations are quite similar, thus attributions ought to be similar too.

Differences on Interpretability Metrics In contrast to the lack of significant differences on visualization, we observe that our approach performs better than GAP on interpretability evaluation. On one hand, since we use a training recipe special for ResNet, we obtain a better accuracy reading with our approach. Since interpretability metrics are related to the recognition capabilities of the classifier, this improvement on explainability is expected. On the other hand, our approach achieves worse accuracy results in ConvNeXt than baseline. However, its interpretability metrics are better. Relating to the prior observation, we argue that if accuracy is low, then prediction probability must be higher: the classifier must be more confident. This is sufficiently demonstrated with the differences of average drop and average gain for all models.

3.8 Conclusion

In this chapter we observe that attention-based pooling in transformers is the same as forming a class agnostic CAM-based saliency map. This map is used to mask the features before global average pooling, much like we mask inputs to confirm that the prediction is due to a certain object. This observation establishes that transformers have a built-in CAM-based interpretability mechanism and allows us to design a similar mechanism for convolutional networks. Masking in feature space is much more efficient than in the input space as it requires only one forward pass, although of course it is not equivalent because of interactions within the network.

Although the saliency maps obtained with our CA-Stream are not very different from those obtained with GAP, our approach improves a number of CAM-based interpretability methods on a number of convolutional networks according to most interpretability metrics, while preserving classification accuracy. By doing so, it also enhances the differences in performance between interpretability methods, facilitating their evaluation. Further study may be needed to improve the differentiation of saliency maps themselves, to possibly make a class specific representation more competitive and to apply the approach to more architectures, including transformers.

4 A learning paradigm for interpretable gradients

Table of contents

4.1 Introduction 92
4.2 Interpretable Gradients 93
4.3 Experiments 95
4.3.1 Experimental Set-up 96
4.3.2 Qualitative Evaluation 96
4.3.3 Quantitative Evaluation 97
4.3.4 Ablation Experiments 98
4.4 Discussion 99
4.5 Conclusion 100

4.1 Introduction

A recurring issue faced by both neural networks and transformers is their inherent lack of interpretability. These models are primarily optimized for high performance in their designated tasks. Yet, reflecting upon the information that can be drawn out of a model without too much effort; we observe that the gradient of deep models displays the response of its parameters, to a given input. Many current interpretability methods are constructed based on this observation.

However, the effective utilization of gradients in interpretability methods remains a pressing question. *How can we leverage gradient better?*, previous interpretability approaches have relied on the stand alone gradient information such as Guided Backpropagation (Springenberg et al. 2014) and Smoothgrad (Smilkov et al. 2017). On another hand as seen in previous chapters some CAM variants are based on gradient, like Grad-CAM (Selvaraju et al. 2016), Grad-CAM++ (Chattopadhay et al. 2018) and Axiom-CAM (Fu et al. 2020). Nevertheless, it is worth reflecting that gradient plays a more prominent role during the training phase of a model, particularly as a cornerstone in this process, we can not help but reflect upon *how can we leverage upon the gradient to improve interpretability during training?*.

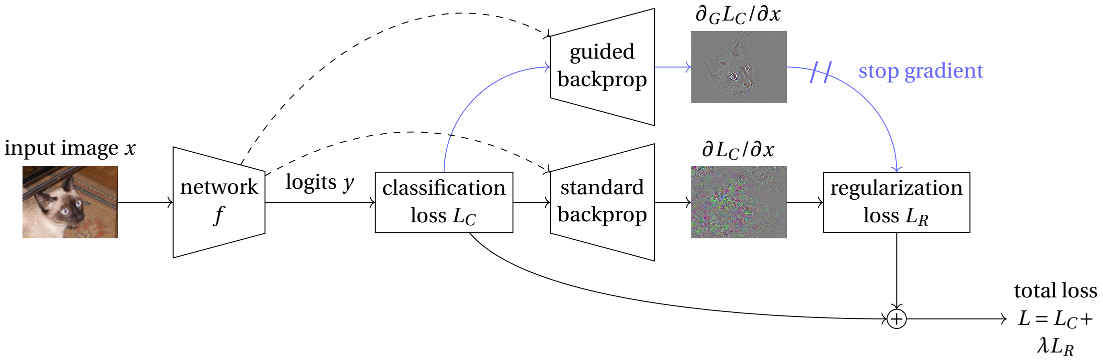

input image x
network f
logits y
classification loss L_C
guided backprop
standard backprop
$\partial_G L_C / \partial x$
$\partial L_C / \partial x$
stop gradient
regularization loss L_R
total loss $L = L_C + \lambda L_R$

Figure 4.1: **Interpretable gradient learning**. For an input image x, we obtain the logit vector $y = f(x;\theta)$ by a forward pass through the network f with parameters θ. We compute the classification loss L_C by softmax and cross-entropy (4.1), (4.2). We obtain the standard gradient $\partial L_C / \partial x$ and guided gradient $\partial_G L_C / \partial x$ by two backward passes (dashed) and compute the regularization loss L_R as the error between the two (4.3),(4.5)-(4.7). The total loss is $L = L_C + \lambda L_R$ (4.4). Learning is based on $\partial L / \partial \theta$, which involves differentiation of the entire computational graph except the guided backpropagation branch (blue).

In this chapter, we propose a modification to the training process of deep models by introducing of a regularization term to the error function. This term constrains the gradient, aligning it with guided backpropagation in the input space.

4.2 Interpretable Gradients

Preliminaries We consider an image classification network f with parameters θ, which maps an input image x to a vector of class logits $y = f(x;\theta)$. At inference, we predict the class label of maximum confidence $\arg\max_j y_j$, where y_j is the logit of class i. At training, given training images $X = \{x_i\}_{i=1}^n$ and target labels $T = \{t_i\}_{i=1}^n$, we compute the *classification loss*

$$L_C(X, \theta, T) = \frac{1}{n} \sum_{i=1}^{n} (f(x_i;\theta), t_i), \tag{4.1}$$

where CE is softmax followed by cross-entropy:

$$(y, t) = -\log \frac{e^{y_t}}{\sum_i e^{y_i}} = -y_t + \sum_i e^{y_i}. \tag{4.2}$$

Updates of parameters θ are then performed by an optimizer, based on the standard partial derivative (gradient) $\partial L_C / \partial \theta$ of the classification loss L_C with respect to θ, obtained by standard back-propagation.

However, due to non-linearities like ReLU activations and downsampling like max-pooling or convolution stride > 1, the standard gradient is noisy (Smilkov et al. 2017). This is shown by visualizing the gradient $\partial L_C / \partial x$ with respect to an input image x. By contrast, the guided gradient $\partial_G L_C / \partial x$ (Springenberg et al. 2014) does not suffer much from noise and preserves sharp details. The difference of the two gradients is illustrated in Figure 4.1.

Regularization The main idea of this work is to introduce a regularization term during training, which will make the standard gradient $\partial L_C / \partial x$ behave similarly to the corresponding guided gradient $\partial_G L_C / \partial x$, while maintaining the predictive power of the classifier f. We hypothesize that, if possible, this will improve the quality of all gradients with respect to intermediate activations and therefore the quality of saliency maps obtained by CAM-based methods (B. Zhou, Khosla, et al. 2016, Selvaraju et al. 2016, Chattopadhay et al. 2018, H. Wang, Du, et al. 2019) and the interpretability of network f. The effect may be similar to that of SmoothGrad (Smilkov et al. 2017), but without the need for several forward passes at inference.
In particular, given an input image x, we perform a forward pass through f and compute the logit vectors $y_i = f(x_i, \theta)$ and the classification loss $L_C(X, \theta, T)$ (4.1). We then obtain the standard gradients $\delta x_i = \partial L_C / \partial x_i$ and the guided gradients $\delta_G x_i = \partial_G L_C / \partial x_i$ with respect to the input images x_i by two separate backward passes. Since the whole process is differentiable (w.r.t. θ) at training, we stop the gradient computation of the latter, so that it only serves as a "teacher". We define the *regularization loss*

$$L_R(X, \theta, T) = \frac{1}{n} \sum_{i=1}^{n} E(\delta x_i, \delta_G x_i), \tag{4.3}$$

where E is an error function between the two gradient images, considered below.

Finally, the total loss is defined as

$$L(x, \theta, t) = L_C(x, \theta, t) + \lambda L_R(x, \theta, t), \tag{4.4}$$

where λ is a hyperparameter determining the regularization coefficient. λ should be large enough to smooth the gradient without decreasing the classification accuracy or hurting the training process. Updates of the network parameters θ are now based on the gradient $\partial L / \partial \theta$ w.r.t. the total loss, using any optimizer. At inference, one may use any interpretability method to obtain a saliency map at any layer.

Algorithm Our method is summarized in Algorithm 3 and illustrated in 4.1.It is interesting to note that the entire computational graph depicted in 4.1 involves one forward and two backward passes. This graph is then differentiated again to compute

$\partial L / \partial \theta$, which involves one more forward and backward pass, since the guided back-propagation branch is excluded. Thus, each training iteration requires five passes through f instead of two in a standard training.

Algorithm 3: Interpretable gradient loss

```
**Input:** network $f$, parameters $\theta$
**Input:** input images $X = \{x_i\}_{i=1}^{n}$
**Input:** target labels $T = \{t_i\}_{i=1}^{n}$
**Output:** loss $L$
$L_C \leftarrow \frac{1}{n}\sum_i (f(x_i;\theta), t_i)$                    ▷ class. loss (4.1)
**foreach** $i \in \{1, \dots, n\}$ **do**
    $\delta x_i \leftarrow \partial L_C / \partial x_i$                      ▷ standard grad
    $\delta_G x_i \leftarrow \partial_G L_C / \partial x_i$                  ▷ guided grad
    DETACH($\delta_G x_i$)                                ▷ detach from graph
$L_R \leftarrow \frac{1}{n}\sum_{i=1}^{n} E(\delta x_i, \delta_G x_i)$              ▷ reg. loss (4.3)
$L \leftarrow L_C + \lambda L_R$                                ▷ total loss (4.4)
```

Error function Given two gradient images δ, δ' consisting of p pixels each, we consider the following error functions E to compute the regularization loss (4.3).

1. Mean absolute error:
$$E_{\mathrm{MAE}}(\delta, \delta') = \frac{1}{p}\left\|\delta - \delta'\right\|_1. \tag{4.5}$$

2. Mean squared error:
$$E_{\mathrm{MSE}}(\delta, \delta') = \frac{1}{p}\left\|\delta - \delta'\right\|_2^2. \tag{4.6}$$

We also consider the following two similarity functions, with a negative sign.

3. Cosine similarity:
$$E_{\cos}(\delta, \delta') = -\frac{\langle \delta, \delta' \rangle}{\|\delta\|_2 \|\delta'\|_2}, \tag{4.7}$$

4. Histogram intersection:
$$E_{\mathrm{HI}}(\delta, \delta') = -\sum_{i=0}^{p} \frac{\min(|\delta_i|, |\delta'_i|)}{\|\delta\|_1 \|\delta'\|_1}. \tag{4.8}$$

where $\langle , \rangle$ denotes inner product.

4.3 Experiments

This section presents the experimental settings, our evaluation metrics and results.

4.3.1 Experimental Set-up

In the following sections, we evaluate recognition properties and interpretability capabilities of our approach. Specifically, we generate explanations following popular attribution methods derived from CAM (B. Zhou, Khosla, et al. 2016) from the **pytorch-grad-cam** library from Jacob Gildenblat[1].

Dataset We train and evaluate our models on CIFAR-100 (Krizhevsky 2009). This dataset contains 60,000 images of 100 categories, split in 50,000 for training and 10,000 for testing. Each image has a resolution of 32 × 32 pixels. This dataset is chosen because of its ease of usage and prototyping properties.

Settings To obtain competitive performance and ensure the replicability of our method, we follow the methodology found in the repository by weiaicunzai [2]. Thus, we train each model following the same training procedure. We perform 200 epochs, with a starting learning rate of 10^{-1}, a batch-size of 128 images, SGD optimizer and a learning rate policy updating said parameter by division over 5 on epochs 60, 120 and 160.

4.3.2 Qualitative Evaluation

We visualize the effect of our approach on saliency maps and gradients, obtained for the baseline model *vs.* the one trained with our approach.

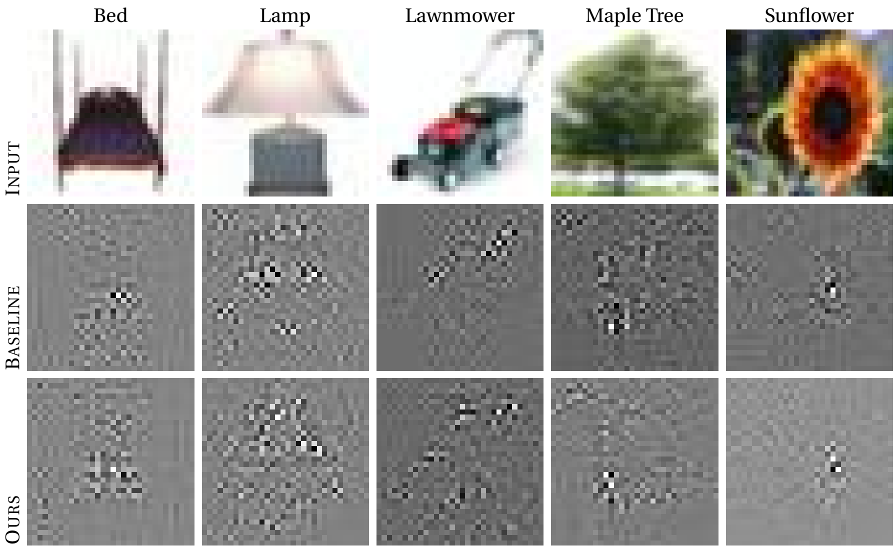


Figure 4.2: **Gradient comparison** of standard *vs.* our training on CIFAR-100 examples.

[1] https://github.com/jacobgil/pytorch-grad-cam
[2] https://github.com/weiaicunzai/pytorch-cifar100

Figure 4.2 shows gradients. We observe slightly less noise with our method, while the object of interest is better covered by gradient activations.

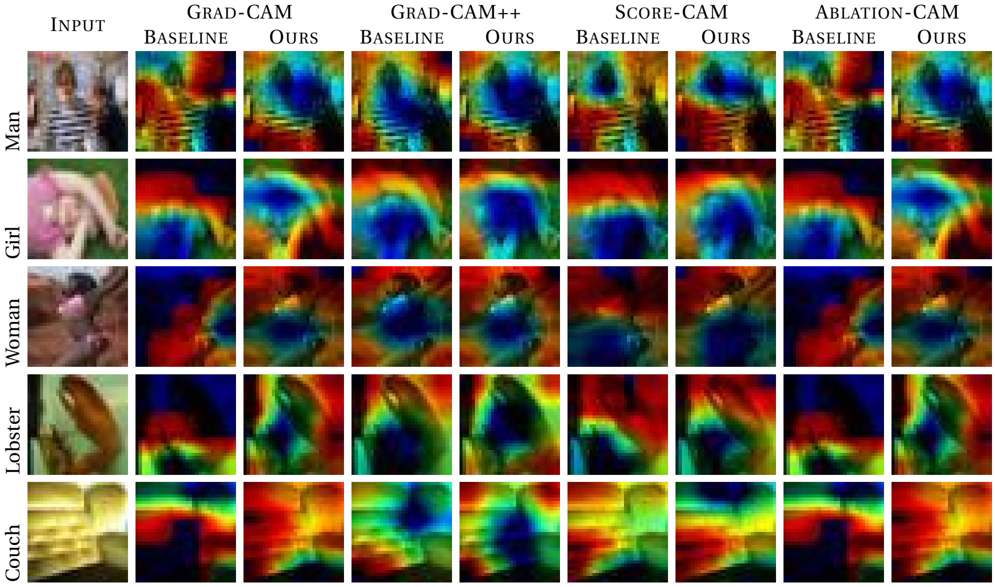


Figure 4.3: **Saliency map comparison** of standard *vs.* our training using different CAM-based methods on CIFAR-100 examples.

Figure 4.3 shows saliency maps. We observe the differences brought by our training method. The differences are particularly important for Grad-CAM, which directly averages the gradient to weigh feature maps. Interestingly, the differences are smaller for Score-CAM, which is not gradient-based but only obtains changes of predicted probabilities.

4.3.3 Quantitative Evaluation

We evaluate the effect of training a given model using our proposed approach with *Faithfulness* and *Causality*. Results are reported in Table 4.1. We observe that our method offers a consistent improvement in terms of interpretability metrics. Specifically, we obtain improvements on both networks and systematically on five out of six metrics. The improvements are higher for AD, AG, and AI. Insertion gets a smaller but consistent improvement and Deletion is almost always worse with our method, but with a very small difference. This decrease in performance of Deletion may be due to some limitations of the metrics as reported in Chapter 2. It is interesting to note that improvements on Score-CAM means that our training not only improves gradient for interpretability, but also builds better activation maps.

Recognition Metrics

Model	Error	λ	Acc↑
ResNet-18	-	-	**73.42**
	Cosine	7.5×10^{-3}	72.86
MobileNet-V2	-	-	59.43
	Cosine	1×10^{-3}	**62.36**

Interpretable Recognition Metrics

ResNet-18

Method	Error	AD↓	AG↑	AI↑	Ins↑	Del↓
Grad-CAM	-	30.16	15.23	29.99	58.47	**17.47**
	Cosine	**28.09**	**16.19**	**31.53**	**58.76**	17.57
Grad-CAM++	-	31.40	14.17	28.47	58.61	**17.05**
	Cosine	**29.78**	**15.07**	**29.60**	**58.90**	17.22
Score-CAM	-	26.49	18.62	33.84	58.42	**18.31**
	Cosine	**24.82**	**19.49**	**35.51**	**59.11**	18.34
Ablation-CAM	-	31.96	14.02	28.33	58.36	**17.14**
	Cosine	**29.90**	**15.03**	**29.61**	**58.70**	17.37
Axiom-CAM	-	30.16	15.23	29.98	58.47	**17.47**
	Cosine	**28.09**	**16.20**	**31.53**	**58.76**	17.57

MobileNet-V2

Method	Error	AD↓	AG↑	AI↑	Ins↑	Del↓
Grad-CAM	-	44.64	6.57	25.62	44.64	**14.34**
	Cosine	**40.89**	**7.31**	**27.08**	**45.57**	15.20
Grad-CAM++	-	45.98	6.12	24.10	44.72	**14.76**
	Cosine	**40.76**	**6.85**	**26.46**	**45.51**	14.92
Score-CAM	-	40.55	7.85	28.57	45.62	**14.52**
	Cosine	**36.34**	**9.09**	**30.50**	**46.35**	14.72
Ablation-CAM	-	45.15	6.38	25.32	44.62	**15.03**
	Cosine	**41.13**	**7.03**	**26.10**	**45.38**	15.12
Axiom-CAM	-	44.65	6.57	25.62	44.64	15.27
	Cosine	**40.89**	**7.31**	**27.08**	**45.57**	**15.20**

Table 4.1: **Cosine Regularization Experiments:** on CIFAR-100 with ResNet-18 and MobileNet-V2. Accuracy and interpretability metrics are reported.

4.3.4 Ablation Experiments

We conduct ablation experiments using ResNet18. In these experiments we analyze the different regularization proposals mentioned in Section 4.2 and the impact of the regularization coefficient.

Regularization proposals To validate our selection of regularization function, we train several models following the same training regime while varying the error function. To evaluate these approaches, we focus solely on Grad-CAM attributions. Results are reported in Table 4.2

Regularization Selection						
Regularizer	Acc	AD↓	AG↑	AI↑	Ins↑	Del↓
-	73.42	30.16	15.23	29.99	58.47	17.47
Cosine	72.86	**28.09**	**16.19**	**31.53**	58.76	17.57
Histogram	73.88	30.39	14.78	29.38	58.52	**17.35**
MAE	**73.41**	30.33	15.06	29.61	58.13	17.95
MSE	73.86	29.64	15.19	30.11	**59.05**	18.02

Table 4.2: **Regularization selection:** Evaluation of the four proposed regularization with ResNet-18 on CIFAR-100.

Following these results, we observe a consistent improvement on most metrics for all regularizer options. We note that the accuracy remains stable within half a percent of the original model. However, we note that most options struggle regarding deletion. Cosine Similarity however manages to provide improvements in most metrics, while maintaining deletion performance.

Regularization coefficient Finally, we study the behavior of the regularization coefficient λ in 4.4. We train multiple models with *Cosine Similarity* and a range of values for λ, see Table 4.3.

Regularization Selection						
λ	Acc	AD↓	AG↑	AI↑	Ins↑	Del↓
-	73.42	30.16	15.23	29.99	58.47	17.47
1×10^{-3}	**73.71**	29.52	15.17	30.03	59.23	**17.45**
2.5×10^{-3}	72.99	30.53	15.82	30.56	59.04	17.96
5×10^{-3}	72.46	30.10	16.06	30.67	57.47	17.80
7.5×10^{-3}	72.86	**28.09**	**16.20**	**31.53**	58.76	17.57
1×10^{-2}	73.28	28.97	15.75	31.16	58.99	17.50
1×10^{-1}	73.00	28.93	16.13	31.55	**59.66**	17.95
1	73.30	28.44	16.02	31.31	58.64	17.48
10	73.04	29.28	15.23	30.47	58.74	17.47

Table 4.3: **Regularization coefficient:** Evaluation of the regularization coefficient λ, using ResNet-18 with *Cosine Similarity* on CIFAR-100.

We observe that our method is not very sensible to the regularization coefficient and that the value of $7.5e^{-3}$ offers better performances and is thus selected as the default value for λ.

4.4 Discussion

Guided Backpropagation and Smoothgrad Guided backpropagation is not the sole gradient explanation we can generate for a specific model. However, it is important to highlight efficiency requirements. On one hand, guided backpropagation only requires two passes through the network: one forward pass and one backward pass. On the other hand, smoothgrad requires several passes throgh the network. By default, this approach involves five forward-backward passes; this would mean a noticeable increase of training time.

Training efficiency Computational complexity and optimized training are the main challenges regarding the scaling of our approach. In particular, in section 4.3, we mention that *each training iteration requires five passes through f instead of two in a standard training*. We compute the first forward-backward pass to generate the guided gradient in the input space. The second forward-backward pass generates the standard gradient. Finally, we do a final backward pass taking into consideration cross entropy and the regularization.

Still, *why can't it be done with fewer passes?* In theory, guided backpropagation is calculated modifying the behavior of activation functions, such as ReLU. In practice, activation functions work as class objects within pytorch. Introducing changes into these objects inherently increases their complexity. For example, a modification to ReLU to account for *Guided ReLU* could be achieved with the introduction of an *if-else* case: one condition for standard ReLU and one for Guided ReLU. In this scenario, the amount of activations in the model would lead to bottlenecks in running time evaluating because of the condition controlling the gradient behavior.

Saliency Map Visualization Upon saliency map visualization we observe a high degree of sparsity covering the input images. After the hierarchical nature of CNNs and the ensuing reduction of feature map sizes, deep representations present small spatial dimensions. Furthermore, since CIFAR-100 contains images with 32×32 spatial resolution, intermediary activations are chosen to avoid generating attributions using a 1×1 patch.

Pure gradient evaluation Pure gradient-based interpretability approaches do not display quantitative measurements to validate interpretability claims. Conversely, we hypothesize that since other attribution proposals rely on this information, denoising gradients leads to explainability improvements. This is ultimately confirmed with our evaluation procedure.

4.5 Conclusion

In this chapter, we propose a transparency methodology that denoises gradients in the input level, improving interpretability properties of CNNs. Our approach aligns the standard gradient with the guided gradient, regularizing the training of neural networks.

We validate our claims on improvement of interpretability properties using post-hoc interpretability evaluation. Our approach displays improvements in these properties and on image recognition. However, an optimized version of this study better suited for datasets and more complex models. Additional experimentation is required to address these limitations. An optimization of our training paradigm reducing its computational would make the method scalable.

Conclusion

Across this thesis we studied explainable deep learning proposals, to understand image recognition models. Explainable Artificial Intelligence and Interpretability are blooming fields within the research community. In particular, current high performing models are being steadily assimilated within society and their prominence in human life is increasing. Thus, it is important to understand the processes prompting a prediction in such models. Furthermore, these fields are being studied following a plethora of axis of research. Still, the work presented by Lipton (Lipton 2018) and Zhang (Y. Zhang et al. 2021) lay the foundation for our work.

We structure these conclusions of our work in the following manner. First on section 4.1 we comment on the thesis objectives and the manner they were addressed across each chapter. On section 4.2 we address future work regarding this topic.

4.1 General remarks

Thesis Objectives This thesis was conducted to study image recognition models, building novel explainability approaches to further understand them. In detail, our major objective was to improve both recognition and interpretability properties. To that end, we identified three areas of work: high computational cost, lack of consensus between evaluation procedures, and a mismatch between human and model interpretability.

To begin with, regarding the improvement of image recognition, this work introduced two different approaches that address this requirement. In particular, Chapter 3 showcased how the addition of cross attention can enhance recognition characteristics. Furthermore, Chapter 4 presented a novel training paradigm that can potentially yield better recognition predictions. Complementary to this, interpretability measurements are improved consistently across this thesis. In particular, Chapter 2 presented *Opti-CAM*, a methodology that consistently improves upon these capabilities, evidenced across different datasets and evaluation modes: recognition and localization.

In line with the specific objectives, our thesis follows a standardized evaluation procedure. Across each chapter, we compare baselines with our approaches under the exact same settings. Conversely, we observed limitations in the details of the evaluation of xAI methodologies. We hope that our contributions will encourage the community towards a set of good practices in this domain.

Lastly, the differences between human and model interpretations were discussed in detail in Chapter 2. Specifically, we observed that context matters in the formulation of saliency maps: the most important regions describing a category are spread across the image. This is highlighted with the failure on interpretable object localization, and the success on interpretable object recognition achieved by Opti-CAM: human centric explanations expect the most salient areas to be found mostly over the object of interest.

Background In Chapter 1, we introduced and described the evolution of image recognition models. We started with models based on traditional machine learning algorithms, to current high performance architectures based on attention computation. In relation to this modelling evolution, we demonstrated how the improvement of image recognition models consequently benefits the development of related Computer Vision fundamental tasks. Thus, further development of image recognition models is acknowledged as a major task in Computer Vision, enhancing adjacent tasks within the discipline.

Complementary to the introduction of these models, we highlighted the necessity of providing explanations to current image recognition models. We mentioned the proposition of the European AI act to regulate Artificial Intelligence technologies, as well as in the Mythos of Model Interpretability by Lipton. In particular, following Lipton's work we revised the properties proposed therein, as well as illustrated how they can be adapted to explain current state-of-the-art models. Furthermore, we demonstrated our interest on CAM methods to produce explanations. A thorough description of their computation and different proposals is established to lay the foundation for the following studies.

Finally, we also introduced evaluation methods to assess the effect of the attribution methods mentioned. Regarding these evaluation procedures, we grouped them according to the reasoning of the measurement provided, as well as highlighted the positive and negative points of each procedure. Notably, Interpretable Object Recognition and Causal Analysis are observed to best assess interpertability properties of a model. On one hand, it is observed that Interpretable Object Localization implies that model interpretations should be aligned with human interpretations. On the other hand, pure human measurements are completely aligned with what individuals deem salient on images, which is often biased and not replicable on experimental settings. Pure classifier centric evaluation ultimately addresses these shortcomings, removing implicit bias produced by human reasoning, although not from supervision.

Opti-CAM Chapter 2 presented our first contribution: Opti-CAM, a post-hoc interpretability method, constructed following the principles of CAM and evaluation procedures. Specifically, Opti-CAM produces a saliency map that maximizes predictive probability of images masked by it. Additionally, issues regarding quantitative evaluation are displayed, most importantly the incompleteness of **Interpretable Object Recognition**. To address these shortcomings, we proposed Average Gain, a complementary metric to Average Drop, measuring the predictive gains obtained while considering explanation maps as input images.

On one hand, we observed that true to its design, Opti-CAM outperforms contemporary CAM attribution methods in most quantitative measurements. In particular, this methodology performs the best in **Interpretable Image Recognition** and **Causal Analysis**, but fails in **Object Localization**. We made sense of these observations aided with visualizations. In contrast to current CAM methods, Opti-CAM generates a saliency map that is spread across the input image. From this we infer that context matters describing an explanation. Consequently, since context is necessary to explain a prediction, the requirement of saliency maps covering the object of interest, is counter-intuitive and does not hold.

Lastly, regarding Average Gain our experimental results demonstrated its complementary behavior to Interpretable Image Recognition Evaluation. In particular, this metric efficiently demonstrates how Fake-CAM fails as an attribution method: although it attains almost perfect Average Drop; its Average Gain measurement fails entirely, effectively complementing the shortcomings instated by this CAM method. Still, a complete benchmark comparing most attribution methods, as well as explanations for predicted labels, would provide a reality check on the evaluation of interpretability

Cross-Attention Stream Chapter 3 presented our second contribution, the Cross-Attention Stream. This addition inspired by pure attention architectures, computes an abstract representation of classes, via interaction of a class token and feature maps across different depths of a model. Additionally, this approach was validated in common image recognition models studied on interpretability such as ResNet, as well as in a family of models not often studied in this fashion: ConvNeXt.

In this chapter, we set the stage for quantitative interpretability measurements for transparency based approaches. We trained the stream similarly to prior transparency approaches, and we evaluated its properties using CAM, a post-hoc method. Moreover, we observed that our saliency maps do not differ much from the baseline ones. However, this result was expected as this approach does not modify existing parameters within the network, nor changes the computation of attributions. Instead, our representation conveys information differently to the classifier, enhancing predicted probability of groundtruth classes.

Gradient Denoising Lastly, Chapter 4 introduced a learning paradigm for interpretable gradients. In this approach, the guided backpropagated gradient of the network, observed in the input space is used to regularize the network gradient during training, enhancing interpretability properties. Continuing with the evaluation of transparency methods seen on Chapter 3, we evaluate these properties using CAM methods.

From the family of pure gradient based attribution methods, guided backpropagation required less computation to function. In stark contrast with approaches such as smooth gradients and integrated gradients, guided backpropagation maintains the requirement of one forward pass and one backward through the model to generate an output.

Lastly, we observed that pure guided backpropagation training is not plausible. During the prototyping phase of this chapter, we experimented using this setting, and we found that the training was unstable leading to gradients pushing towards infinity. We hypothesize that gradient information produced by responses to negative gradients, regularizes neural network training.

4.2 Future Work

We set the foundation for our future work in three axes. First, a discussion on the future for interpretable image recognition. Then, we iterate over how our proposals can be improved upon in the future. Finally, exploratory directions beyond the scope of this thesis.

Interpretable Image Recognition The development of image recognition models within computer vision and deep learning benefits from ongoing advancements. With the recent emergence of CNNs and transformers, new architectures are expected to continue appearing. Evaluating the impact of these models through testing and proposing methodologies is essential for future progress. While CNNs have been extensively studied over the past decade, the properties and functioning of transformers remain an open field. Despite transformers being newer, their impact and performance are significant, necessitating further study. However, research on CNNs should also continue.

Additionally, standardization of interpretability study and evaluation is another area with potential for future work. A thorough differentiation between model interpretability and human interpretability studies should be established. A preliminary study regarding this topic is conducted in this thesis, still a widespread adoption within the community is mostly desired. To achieve this, a more thorough survey describing these comparisons, as well as the failure of current interpretability evaluation methods is one manner to address these requirements.

On the chapters of this thesis Regarding Opti-CAM, we observe one possibility for future work. CAM-based attributions struggle to explain transformer-based architectures. These saliency maps are often sparse and do not provide sufficient clarity when compared to raw attention. A different attribution method could provide improved representations, possibly calculated using the class token. Accounting for the fixation of saliency maps on transformers as demonstrated on Simpool (Psomas et al. 2023) and Registers (Darcet et al. 2024) would allow for updates of this approach or a novel attribution method.

On the topic of Cross-Attention, future work aims at optimizing the architecture and broadening its scope to different models. In particular, the parameter count can be reduced by shortening the stream length, focusing only on layers where semantic information is prominent. Additionally, expanding beyond the usage of ResNet and ConvNeXt, and presenting an optimized training paradigm for this approach, would enable its application to more architectures, enhancing its coverage of image recognition tasks.

Lastly, the gradient denoising paradigm was showcased in a constrained setting: small datasets and low-parameter networks. This limitation is due to the high computational cost of the approach. However, the promising results suggest that addressing this complexity could allow for scaling to large-scale image datasets and more complex models.

Beyond the scope of this thesis Currently, computer vision is one open field, thriving with possibilities for further research and industry developments. In particular, during the development of this thesis several technologies were unveiled, addressing different areas of study for artificial intelligence. For instance, NLP is currently a prominent field where technologies like Large Language Models have taken the spotlight. However, such kind of developments require heavy computational infrastructure, limiting their development to bigger research groups. Optimizing such methodologies, as well as producing competitive, yet more simplified alternatives is one path where research could be focused as well.

In contrast to developing large models and mainstream tasks, future work could focus on updating particular applications. Concepts from general image recognition and interpretability are applied to fields such as medical diagnosis and industry, requiring highly specific models. Adapting and modifying state-of-the-art models for these fine-grained applications is a key direction for future developments.

Lastly, on a personal level, future work comprises on setting the stage for continuing a scientific career. To achieve this, I aim to continue with a post-doctoral position on topics aligned with my interests: image recognition, explainable AI and foundational models. Moreover, given my focus on academia, pursuing a research engineer position is a direction that would also allow to advance for my career.

Bibliography

[AZ20] Samira Abnar and Willem Zuidema. “Quantifying attention flow in transformers”. In: *arXiv preprint arXiv:2005.00928* (2020) (cit. on pp. 41, 46, 66, 68, 78).

[Ade+18] Julius Adebayo, Justin Gilmer, Ian J. Goodfellow, et al. “Local Explanation Methods for Deep Neural Networks Lack Sensitivity to Parameter Values”. In: *ICLR Workshop* (2018) (cit. on p. 45).

[AS20] Mark Andrejevic and Neil Selwyn. “Facial recognition technology in schools: Critical questions and concerns”. In: *Learning, Media and Technology* 45.2 (2020), pp. 115–128 (cit. on p. 25).

[Bac+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, et al. “On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation”. In: *PloS one* (2015) (cit. on pp. 19, 41, 45, 46).

[Bae+10] David Baehrens, Timon Schroeter, Stefan Harmeling, et al. “How to Explain Individual Classification Decisions”. In: *J. MLR* (2010) (cit. on pp. 41, 45).

[Bau+17] David Bau, Bolei Zhou, Aditya Khosla, et al. “Network dissection: Quantifying interpretability of deep visual representations”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6541–6549 (cit. on pp. 19, 41, 52, 55).

[BDF10] Alex Berg, Jia Deng, and L Fei-Fei. *Large scale visual recognition challenge 2010*. 2010 (cit. on p. 30).

[Bod+21] Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, et al. “Benchmarking and Survey of Explanation Methods for Black Box Models”. In: *CoRR* abs/2102.13076 (2021). arXiv: 2102.13076 (cit. on p. 18).

[BFS22] Moritz Böhle, Mario Fritz, and Bernt Schiele. “B-cos networks: Alignment is all we need for interpretability”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10329–10338 (cit. on pp. 41, 43).

[Böh+24] Moritz Böhle, Navdeeppal Singh, Mario Fritz, et al. “B-cos Alignment for Inherently Interpretable CNNs and Vision Transformers”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024) (cit. on pp. 41, 43).

[Cha+19] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, et al. "Explaining Image Classifiers by Counterfactual Generation". In: *ICLR* (2019) (cit. on pp. 41, 45).

[Cha+09] Jonathan Chang, Sean Gerrish, Chong Wang, et al. "Reading tea leaves: How humans interpret topic models". In: *Advances in neural information processing systems* 22 (2009) (cit. on p. 44).

[Cha+18] A. Chattopadhay, A. Sarkar, P. Howlader, et al. "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks". In: *WACV*. 2018 (cit. on pp. 20, 41, 46, 48, 52, 58, 60, 65, 67, 71, 74, 79, 82, 92, 94).

[CGW21] Hila Chefer, Shir Gur, and Lior Wolf. "Transformer interpretability beyond attention visualization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 782–791 (cit. on pp. 41, 46, 66, 68, 69, 78, 85).

[Che+19] Chaofan Chen, Oscar Li, Daniel Tao, et al. "This looks like that: deep learning for interpretable image recognition". In: *Advances in neural information processing systems* 32 (2019) (cit. on pp. 41, 43).

[Cho+20] Junsuk Choe, Seong Joon Oh, Seungho Lee, et al. "Evaluating Weakly Supervised Object Localization Methods Right". In: *CVPR*. 2020 (cit. on pp. 52, 56, 71).

[CV95] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20 (1995), pp. 273–297 (cit. on p. 27).

[CH67] Thomas Cover and Peter Hart. "Nearest neighbor pattern classification". In: *IEEE transactions on information theory* 13.1 (1967), pp. 21–27 (cit. on p. 27).

[Cru+12] Javier Cruz-Mota, Iva Bogdanova, Benoît Paquier, et al. "Scale invariant feature transform on the sphere: Theory and applications". In: *International journal of computer vision* 98 (2012), pp. 217–241 (cit. on p. 27).

[Csu+04] Gabriella Csurka, Christopher Dance, Lixin Fan, et al. "Visual categorization with bags of keypoints". In: *Workshop on statistical learning in computer vision, ECCV*. Vol. 1. 1-22. Prague. 2004, pp. 1–2 (cit. on p. 27).

[DG17] Piotr Dabkowski and Yarin Gal. "Real Time Image Saliency for Black Box Classifiers". In: *NIPS* (2017). Ed. by I. Guyon, U. V. Luxburg, S. Bengio, et al. (cit. on pp. 41, 45, 52, 56, 71).

[DT05] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893 (cit. on p. 24).

[Dar+24] Timothée Darcet, Maxime Oquab, Julien Mairal, et al. "Vision Transformers Need Registers". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=2dnO3LLiJ1 (cit. on p. 106).

[DR20] Saurabh Desai and Harish G. Ramaswamy. "Ablation-CAM: Visual Explanations for Deep Convolutional Network via Gradient-free Localization". In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020, pp. 972–980. DOI: 10.1109/WACV45572.2020.9093360 (cit. on pp. 41, 46, 50, 58).

[Dev+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018) (cit. on pp. 34, 78).

[DCN16] Vinayak V Dixit, Sai Chand, and Divya J Nair. "Autonomous vehicles: disengagements, accidents and reaction times". In: *PLoS one* 11.12 (2016), e0168054 (cit. on p. 25).

[Dol+12] Piotr Dollar, Christian Wojek, Bernt Schiele, et al. "Pedestrian Detection: An Evaluation of the State of the Art". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.4 (2012), pp. 743–761. DOI: 10.1109/TPAMI.2011.155 (cit. on p. 24).

[DWA15] Finale Doshi-Velez, Byron Wallace, and Ryan Adams. "Graph-sparse lda: a topic model with structured sparsity". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1. 2015 (cit. on p. 44).

[Dos+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020) (cit. on pp. 35, 65, 66, 68, 77, 79, 87).

[DC20] Rachel Lea Draelos and Lawrence Carin. "Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks". In: *arXiv preprint arXiv:2011.08891* (2020) (cit. on p. 65).

[DR18] Fábio Duarte and Carlo Ratti. "The impact of autonomous vehicles on cities: A review". In: *Journal of Urban Technology* 25.4 (2018), pp. 3–18 (cit. on p. 25).

[DH72] Richard O Duda and Peter E Hart. "Use of the Hough transformation to detect lines and curves in pictures". In: *Communications of the ACM* 15.1 (1972), pp. 11–15 (cit. on p. 27).

[Esc+19] María Escobar, Cristina González, Felipe Torres, et al. "Hand Pose Estimation for Pediatric Bone Age Assessment". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019 (cit. on p. 25).

[Eve+15] M. Everingham, S. M. A. Eslami, L. Van Gool, et al. “The Pascal Visual Object Classes Challenge: A Retrospective”. In: *International Journal of Computer Vision* 111.1 (Jan. 2015), pp. 98–136 (cit. on p. 82).

[Eve+] M. Everingham, L. Van Gool, C. K. I. Williams, et al. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.* http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html (cit. on p. 24).

[Fin01] Stanley Finger. *Origins of neuroscience: a history of explorations into brain function.* Oxford University Press, 2001 (cit. on p. 23).

[FH89] Evelyn Fix and Joseph Lawson Hodges. “Discriminatory analysis. Non-parametric discrimination: Consistency properties”. In: *International Statistical Review/Revue Internationale de Statistique* 57.3 (1989), pp. 238–247 (cit. on p. 27).

[FV17] Ruth C. Fong and Andrea Vedaldi. “Interpretable Explanations of Black Boxes by Meaningful Perturbation”. In: *ICCV.* 2017 (cit. on pp. 41, 46, 54, 62, 79).

[FPV19] 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 (cit. on pp. 41, 46, 54, 61, 62, 65, 69, 79).

[Fu+20] Ruigang Fu, Qingyong Hu, Xiaohu Dong, et al. “Axiom-based Grad-CAM: Towards Accurate Visualization and Explanation of CNNs”. In: *BMVC* (2020) (cit. on pp. 41, 46, 50, 60, 65, 92).

[Fuk69] Kunihiko Fukushima. “Visual feature extraction by a multilayered network of analog threshold elements”. In: *IEEE Transactions on Systems Science and Cybernetics* 5.4 (1969), pp. 322–333 (cit. on p. 29).

[Fuk75] Kunihiko Fukushima. “Cognitron: A self-organizing multilayered neural network”. In: *Biological cybernetics* 20.3-4 (1975), pp. 121–136 (cit. on pp. 28, 40).

[Gha+19] Reza Ghaeini, Xiaoli Z Fern, Hamed Shahbazi, et al. “Saliency learning: Teaching the model where to pay attention”. In: *NAACL* (2019) (cit. on pp. 41, 43).

[GFM22] Tristan Gomez, Thomas Fréour, and Harold Mouchère. “Metrics for saliency map evaluation of deep learning explanation methods”. In: *International Conference on Pattern Recognition and Artificial Intelligence.* Springer. 2022, pp. 84–95 (cit. on p. 85).

[Gra+21] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, et al. “Levit: a vision transformer in convnet’s clothing for faster inference”. In: *Proceedings of the IEEE/CVF international conference on computer vision.* 2021, pp. 12259–12269 (cit. on pp. 36, 77).

[Gui+18] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, et al. "A Survey of Methods for Explaining Black Box Models". In: *ACM Comput. Surv.* 51.5 (2018). ISSN: 0360-0300 (cit. on pp. 18, 40, 45).

[Gul09] A Gullstrand. "Hemholtz's Handbuch der Physiologischen Optik". In: *English translation ed. JP Southall.) Oxford: Butterworth-Heinemann* (1909) (cit. on p. 23).

[Har54] Zellig S Harris. "Distributional structure". In: *Word* 10.2-3 (1954), pp. 146–162 (cit. on p. 27).

[HXB21] Peter Hase, Harry Xie, and Mohit Bansal. *The Out-of-Distribution Problem in Explainability and Search Methods for Feature Importance Explanations*. 2021. arXiv: `2106.00786 [cs.LG]` (cit. on p. 85).

[He+17] Kaiming He, Georgia Gkioxari, Piotr Dollár, et al. "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969 (cit. on p. 32).

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on pp. 25, 31, 65, 82).

[Heo+21] Byeongho Heo, Sangdoo Yun, Dongyoon Han, et al. "Rethinking spatial dimensions of vision transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 11936–11945 (cit. on p. 35).

[Ho95] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282 (cit. on p. 27).

[HSS08] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. "Kernel methods in machine learning". In: (2008) (cit. on p. 27).

[HT85] John J Hopfield and David W Tank. ""Neural" computation of decisions in optimization problems". In: *Biological cybernetics* 52.3 (1985), pp. 141–152 (cit. on p. 40).

[HSS18] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141 (cit. on p. 32).

[Hua+17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708 (cit. on p. 32).

[Hua+21] Shigao Huang, Jie Yang, Simon Fong, et al. "Artificial intelligence in the diagnosis of COVID-19: challenges and perspectives". In: *International journal of biological sciences* 17.6 (2021), p. 1581 (cit. on p. 25).

[HW59] David H Hubel and Torsten N Wiesel. “Receptive fields of single neurones in the cat’s striate cortex”. In: *The Journal of physiology* 148.3 (1959), p. 574 (cit. on pp. 23, 28, 47).

[IS15] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *ICML*. 2015 (cit. on p. 65).

[ICF21] Aya Abdelsalam Ismail, Hector Corrada Bravo, and Soheil Feizi. “Improving deep learning interpretability by saliency guided training”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 26726–26739 (cit. on pp. 41, 43, 44).

[Jia+21] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, et al. “Layer-CAM: Exploring hierarchical class activation maps for localization”. In: *IEEE Transactions on Image Processing* 30 (2021), pp. 5875–5888 (cit. on pp. 41, 46, 51, 58, 65).

[KZG+18] Daniel Kermany, Kang Zhang, Michael Goldbaum, et al. “Labeled optical coherence tomography (oct) and chest X-ray images for classification”. In: *Mendeley data* 2.2 (2018) (cit. on p. 65).

[KRS14] Been Kim, Cynthia Rudin, and Julie A Shah. “The bayesian case model: A generative approach for case-based reasoning and prototype classification”. In: *Advances in neural information processing systems* 27 (2014) (cit. on p. 44).

[KB15] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *ICLR* (2015). Ed. by Yoshua Bengio and Yann LeCun (cit. on p. 64).

[Kri09] Alex Krizhevsky. “Learning Multiple Layers of Features from Tiny Images”. In: (2009), pp. 32–33. URL: `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf` (cit. on p. 96).

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012) (cit. on pp. 29, 65).

[LeC+98] Yann LeCun, Léon Bottou, Yoshua Bengio, et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cit. on pp. 24, 28).

[Lee+09] Honglak Lee, Roger Grosse, Rajesh Ranganath, et al. “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations”. In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 609–616 (cit. on p. 47).

[Lee+21] Kwang Hee Lee, Chaewon Park, Junghyun Oh, et al. “LFI-CAM: Learning feature importance for better visual explanation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 1355–1363 (cit. on pp. 20, 41, 43, 44, 58).

[Li+21] Liangzhi Li, Bowen Wang, Manisha Verma, et al. “SCOUTER: Slot attention-based classifier for explainable image recognition”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 1046–1055 (cit. on p. 58).

[Li+18] Oscar Li, Hao Liu, Chaofan Chen, et al. “Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018 (cit. on pp. 18, 39, 41, 43).

[LCY13] Min Lin, Qiang Chen, and Shuicheng Yan. “Network in Network”. In: *arXiv preprint arXiv:1312.4400* (2013) (cit. on pp. 29–31, 78).

[Lin+17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, et al. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988 (cit. on p. 32).

[Lin+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. “Microsoft coco: Common objects in context”. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755 (cit. on p. 24).

[Lip18] 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 (cit. on pp. 19, 25, 41, 102).

[Liu+21] Ze Liu, Yutong Lin, Yue Cao, et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022 (cit. on pp. 36, 77).

[Liu+22] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, et al. “A convnet for the 2020s”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11976–11986 (cit. on pp. 33, 77, 82).

[LCG12] Yin Lou, Rich Caruana, and Johannes Gehrke. “Intelligible models for classification and regression”. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2012, pp. 150–158 (cit. on p. 42).

[Low99] David G Lowe. “Object recognition from local scale-invariant features”. In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157 (cit. on p. 27).

[Mac+67] James MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297 (cit. on p. 28).

[Mad21] Tambiama Madiega. “Artificial intelligence act”. In: *European Parliament: European Parliamentary Research Service* (2021) (cit. on p. 26).

[MV15] Aravindh Mahendran and Andrea Vedaldi. “Understanding deep image representations by inverting them”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 5188–5196 (cit. on p. 44).

[Mal+16] Jitendra Malik, Pablo Arbeláez, Joao Carreira, et al. “The three R’s of computer vision: Recognition, reconstruction and reorganization”. In: *Pattern Recognition Letters* 72 (2016), pp. 4–14 (cit. on pp. 18, 38).

[Mar10] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010 (cit. on pp. 23, 27).

[Mar+01] D. Martin, C. Fowlkes, D. Tal, et al. “A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics”. In: *Proc. 8th Int’l Conf. Computer Vision*. Vol. 2. June 2001, pp. 416–423 (cit. on p. 24).

[ML13] Julian McAuley and Jure Leskovec. “Hidden factors and hidden topics: understanding rating dimensions with review text”. In: *Proceedings of the 7th ACM conference on Recommender systems*. 2013, pp. 165–172 (cit. on p. 44).

[Mil18] Adam Millard-Ball. “Pedestrians, autonomous vehicles, and cities”. In: *Journal of planning education and research* 38.1 (2018), pp. 6–12 (cit. on p. 25).

[MBL20] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. “A metric learning reality check”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*. Springer. 2020, pp. 681–699 (cit. on p. 38).

[Nai+20] Rakshit Naidu, Ankita Ghosh, Yash Maurya, et al. “IS-CAM: Integrated Score-CAM for axiomatic-based explanations”. In: *arXiv preprint arXiv:2010.03023* (2020) (cit. on pp. 20, 58).

[Nak+13] Ryoichi Nakashima, Kazufumi Kobayashi, Eriko Maeda, et al. “Visual search of experts in medical image reading: the effect of training, target prevalence, and expert knowledge”. In: *Frontiers in psychology* 4 (2013), p. 166 (cit. on p. 25).

[Ope18] OpenAI. *AI and Compute*. URL: https://openai.com/research/ai-and-compute. 2018 (cit. on pp. 19, 40).

[PMB13] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks". In: *International conference on machine learning*. Pmlr. 2013, pp. 1310–1318 (cit. on p. 31).

[Pen+21] Zhiliang Peng, Wei Huang, Shanzhi Gu, et al. "Conformer: Local features coupling global representations for visual recognition". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 367–376 (cit. on pp. 37, 58, 77).

[PDN22] Jeremy Petch, Shuang Di, and Walter Nelson. "Opening the black box: the promise and limitations of explainable machine learning in cardiology". In: *Canadian Journal of Cardiology* 38.2 (2022), pp. 204–213 (cit. on p. 39).

[Pet+22] Suzanne Petryk, Lisa Dunlap, Keyan Nasseri, et al. "On guiding visual attention with language specification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18092–18102 (cit. on p. 74).

[PPG20] Jason Phang, Jungkyu Park, and Krzysztof J Geras. "Investigating and Simplifying Masking-based Saliency Methods for Model Interpretability". In: *arXiv preprint arXiv:2010.09750* (2020) (cit. on pp. 41, 45).

[Pog81] Tomaso Poggio. "Marr's computational approach to vision". In: *Trends in neurosciences* 4 (1981), pp. 258–262 (cit. on pp. 23, 27).

[Pog+17] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, et al. "Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection". In: *Multimedia Systems Conf.* 2017 (cit. on p. 65).

[Pop+21] Samuele Poppi, Marcella Cornia, Lorenzo Baraldi, et al. "Revisiting the evaluation of class activation mapping for explainability: A novel metric and experimental analysis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2299–2304 (cit. on pp. 52, 53, 61, 64, 67, 68, 75).

[Pso+23] Bill Psomas, Ioannis Kakogeorgiou, Konstantinos Karantzalos, et al. "Keep It SimPool: Who Said Supervised Transformers Suffer from Attention Deficit?" In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2023, pp. 5350–5360 (cit. on p. 106).

[Qiu+21] Luyu Qiu, Yi Yang, Caleb Chen Cao, et al. *Resisting Out-of-Distribution Data Problem in Perturbation of XAI*. 2021. arXiv: `2107.14000 [cs.AI]` (cit. on p. 85).

[QT09] Ariadna Quattoni and Antonio Torralba. "Recognizing indoor scenes". In: *2009 IEEE conference on computer vision and pattern recognition*. 2009 (cit. on pp. 83, 84).

[RBS22] Sukrut Rao, Moritz Böhle, and Bernt Schiele. "Towards Better Understanding Attribution Methods". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10223–10232 (cit. on pp. 70, 74).

[Ren+15] Shaoqing Ren, Kaiming He, Ross Girshick, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015) (cit. on p. 32).

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *SIGKDD*. KDD '16. San Francisco, California, USA, 2016. ISBN: 9781450342322 (cit. on pp. 19, 41, 46, 52, 79).

[REO20] Robin Rombach, Patrick Esser, and Björn Ommer. "Making sense of cnns: Interpreting deep representations and their invariances with inns". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*. Springer. 2020, pp. 647–664 (cit. on p. 44).

[Ron+22] Yao Rong, Tobias Leemann, Vadim Borisov, et al. *A Consistent and Efficient Evaluation Strategy for Attribution Methods*. 2022. arXiv: 2202.00449 [cs.CV] (cit. on p. 85).

[RHD17] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. "Right for the right reasons: Training differentiable models by constraining their explanations". In: *IJCAI* (2017) (cit. on pp. 41, 43).

[Rus+15] Olga Russakovsky, Jia Deng, Hao Su, et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y (cit. on pp. 24, 29, 65, 82).

[Rym+22] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, et al. "Interpretable image classification with differentiable prototypes assignment". In: *European Conference on Computer Vision*. Springer. 2022, pp. 351–368 (cit. on pp. 41, 43).

[Sam+16] Wojciech Samek, Alexander Binder, Grégoire Montavon, et al. "Evaluating the visualization of what a deep neural network has learned". In: *IEEE transactions on neural networks and learning systems* 28.11 (2016), pp. 2660–2673 (cit. on p. 68).

[SP11] Jorge Sánchez and Florent Perronnin. "High-dimensional signature compression for large-scale image classification". In: *CVPR 2011*. IEEE. 2011, pp. 1665–1672 (cit. on p. 30).

[R20] saurabh desai saurabh and Harish Guruprasad Ramaswamy. "Ablation-CAM: Visual Explanations for Deep Convolutional Network via Gradient-free Localization". In: *WACV*. 2020 (cit. on p. 65).

[Sch+20] Karl Schulz, Leon Sixt, Federico Tombari, et al. “Restricting the flow: Information bottlenecks for attribution”. In: *arXiv preprint arXiv:2001.00396* (2020) (cit. on pp. 41, 45, 61, 79).

[SH20] Evan Selinger and Woodrow Hartzog. “The inconsentability of facial surveillance”. In: *Loy. L. Rev.* 66 (2020), p. 33 (cit. on p. 25).

[Sel+16] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, et al. “Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization”. In: *CoRR* abs/1610.02391 (2016). arXiv: 1610.02391. URL: http://arxiv.org/abs/1610.02391 (cit. on pp. 41, 46, 48, 52, 60, 65, 82, 92, 94).

[SSF19] Rakshith Shetty, Bernt Schiele, and Mario Fritz. “Not Using the Car to See the Sidewalk–Quantifying and Controlling the Effects of Context in Classification and Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8218–8226 (cit. on pp. 70, 74).

[SVZ14] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”. In: *ICLR Workshop* (2014) (cit. on pp. 41, 45).

[SZ15] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV] (cit. on pp. 31, 65).

[Smi+17] Daniel Smilkov, Nikhil Thorat, Been Kim, et al. “Smoothgrad: removing noise by adding noise”. In: *arXiv preprint arXiv:1706.03825* (2017) (cit. on pp. 41, 45, 60, 92, 94).

[Spr+14] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, et al. “Striving for simplicity: The all convolutional net”. In: *arXiv preprint arXiv:1412.6806* (2014) (cit. on pp. 41, 45, 48, 92, 94).

[Sun+17] Chen Sun, Abhinav Shrivastava, Saurabh Singh, et al. “Revisiting unreasonable effectiveness of data in deep learning era”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 843–852 (cit. on p. 35).

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. “Axiomatic Attribution for Deep Networks”. In: *ICML*. 2017 (cit. on pp. 41, 45).

[Swe10] Rivka Swenson. “Optics, Gender, and the Eighteenth-Century Gaze: Looking at Eliza Haywood’s Anti-Pamela”. In: *The Eighteenth Century* 51.1 (2010), pp. 27–43 (cit. on p. 23).

[Sze+15] Christian Szegedy, Wei Liu, Yangqing Jia, et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9 (cit. on p. 31).

[TL19] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114 (cit. on p. 32).

[Tou+21a] Hugo Touvron, Matthieu Cord, Matthijs Douze, et al. “Training data-efficient image transformers & distillation through attention”. In: *International Conference on Machine Learning*. Vol. 139. July 2021, pp. 10347–10357 (cit. on p. 65).

[Tou+21b] Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, et al. “Augmenting convolutional networks with attention-based aggregation”. In: *arXiv preprint arXiv:2112.13692* (2021) (cit. on pp. 36, 77, 88).

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017) (cit. on pp. 33, 77, 87).

[VAK18] Petsiuk Vitali, Das Abir, and Saenko Kate. “RISE: Randomized Input Sampling for Explanation of Black-box Models”. In: *BMVC* (2018) (cit. on pp. 41, 46, 52, 54, 65, 69, 79, 80, 82).

[Wag+12] Johan Wagemans, James H Elder, Michael Kubovy, et al. “A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure–ground organization.” In: *Psychological bulletin* 138.6 (2012), p. 1172 (cit. on p. 23).

[Wah+11] C. Wah, S. Branson, P. Welinder, et al. *Caltech-UCSD Birds-200-2011*. Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011 (cit. on p. 82).

[Wan+19] Haofan Wang, Mengnan Du, Fan Yang, et al. “Score-CAM: Improved Visual Explanations Via Score-Weighted Class Activation Mapping”. In: *CoRR* abs/1910.01279 (2019). arXiv: `1910.01279`. URL: `http://arxiv.org/abs/1910.01279` (cit. on pp. 20, 41, 46, 49, 50, 52, 56, 58, 60, 62, 65, 71, 79, 82, 94).

[Wan+20] Haofan Wang, Rakshit Naidu, Joy Michael, et al. “SS-CAM: Smoothed Score-CAM for sharper visual feature localization”. In: *arXiv preprint arXiv:2006.14255* (2020) (cit. on p. 20).

[Wan+18] Xiaolong Wang, Ross Girshick, Abhinav Gupta, et al. “Non-local neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7794–7803 (cit. on p. 32).

[WTJ21] Ross Wightman, Hugo Touvron, and Hervé Jégou. “ResNet strikes back: An improved training procedure in timm”. In: *arXiv preprint arXiv:2110.00476* (2021) (cit. on pp. 30, 32, 38).

[Wu+18] Mike Wu, Michael Hughes, Sonali Parbhoo, et al. “Beyond sparsity: Tree regularization of deep models for interpretability”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 2018 (cit. on pp. 19, 41, 43).

[Wu+20] Mike Wu, Sonali Parbhoo, Michael Hughes, et al. “Regional tree regularization for interpretability in deep neural networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 2020, pp. 6413–6421 (cit. on pp. 41, 43).

[Xia+21] Tete Xiao, Mannat Singh, Eric Mintun, et al. “Early convolutions help transformers see better”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 30392–30400 (cit. on p. 37).

[Yeh+19] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, et al. “On the (in) fidelity and sensitivity of explanations”. In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 68).

[Yu+16] Lequan Yu, Hao Chen, Qi Dou, et al. “Automated melanoma recognition in dermoscopy images via very deep residual networks”. In: *IEEE transactions on medical imaging* 36.4 (2016), pp. 994–1004 (cit. on p. 25).

[Zha+23] Hanwei Zhang, Felipe Torres, Ronan Sicre, et al. “Opti-CAM: Optimizing saliency maps for interpretability”. In: *arXiv preprint arXiv:2301.07002* (2023) (cit. on p. 79).

[Zha+17] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, et al. “Top-Down Neural Attention by Excitation Backprop”. In: *IJCV* 126 (2017), pp. 1084–1102 (cit. on pp. 52, 56, 71).

[ZWZ18] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. “Interpretable convolutional neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8827–8836 (cit. on pp. 41, 43).

[Zha+21] Yu Zhang, Peter Tiňo, Aleš Leonardis, et al. “A survey on neural network interpretability”. In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5 (2021), pp. 726–742 (cit. on pp. 19, 40–42, 45, 57, 102).

[Zho+19] Bolei Zhou, David Bau, Aude Oliva, et al. “Interpreting Deep Visual Representations via Network Dissection”. In: *Trans. PAMI* 41.9 (2019), pp. 2131–2145 (cit. on pp. 43, 74).

[Zho+16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, et al. “Learning Deep Features for Discriminative Localization”. In: *CVPR*. 2016 (cit. on pp. 19, 41, 46, 48, 52, 60, 79, 81, 94, 96).

[Zho+22] Hao Zhou, Keyang Cheng, Yu Si, et al. "Improving Interpretability by Information Bottleneck Saliency Guided Localization". In: *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. URL: https://bmvc2022.mpi-inf.mpg.de/0605.pdf (cit. on pp. 41, 43).

[ZGC20] Konrad Zolna, Krzysztof J. Geras, and Kyunghyun Cho. "Classifier-agnostic saliency map extraction". In: *CVIU* 196 (2020), p. 102969. ISSN: 1077-3142 (cit. on pp. 41, 45).

[Zop+18] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, et al. "Learning transferable architectures for scalable image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8697–8710 (cit. on p. 32).