UCLouvain

Bone X-Ray Analysis with Explainable AI and Multimodal Self-Supervised Learning

Alexandre Englebert

Thesis submitted in partial fulfillment
of the requirements for the degree of
Ph.D. in biomedical and pharmaceutical sciences

Dissertation committee:

Prof. Christophe De Vleeschouwer (UCLouvain)
Prof. Olivier Cornu (UCLouvain)
Prof. Frédéric Lecouvet (UCLouvain)
Prof. John Lee (UCLouvain)
Prof. Thomas Schubert (UCLouvain)
Prof. Thomas Kirchgesner (UCLouvain)
Prof. Benoît Macq (UCLouvain)
Prof. Jean-Philippe Thiran (EPFL, Switzerland)
Dr. Maxime Taquet (University of Oxford, United Kingdom)

Version of November 6, 2024.

Nothing is impossible, the word itself says "I'm possible"! — Audrey Hepburn

Abstract

The rapid development of artificial intelligence (AI) in medicine promises to transform diagnosis, treatment, research, and medical education. AI-powered systems have demonstrated remarkable capabilities in image recognition, natural language processing, and predictive analytics, improving the accuracy and efficiency of various applications. However, deploying AI models in medicine faces significant challenges, including the need for more model transparency and high-quality annotated datasets .

Explainability: To address the "black-box" nature of AI models, we developed explainability methods to improve transparency and trust in AI diagnoses. We made Poly-CAM, a method for generating high-resolution class activation maps (CAMs) for Convolutional Neural Networks (CNNs) without relying on gradient backpropagation. We demonstrated Poly-CAM on bone radiographs to identify potential biases in model predictions. Additionally, we introduced Transformer Input Sampling (TIS), enhancing explainability for vision transformers by sampling tokens.

Self-Supervision and Vision-Language Models: Given the scarcity of annotated medical data, we decided to explore self-supervised learning techniques to reduce the need for manual annotation while maintaining robust model performance. Due to the lack of available datasets, we also created a dataset from raw bone radiographic images and French reports at Cliniques Universitaires Saint Luc, which enabled us to explore self-supervised multimodal techniques. We demonstrated the effectiveness of self-supervised techniques and pseudo-labels for enhancing downstream tasks. Additionally, we preprocessed the dataset to produce training data for future vision-language models aimed at automating medical report generation and visual question answering (VQA).

Overall, this research contributes to a more transparent and reliable healthcare system where AI supports medical professionals, and opens doors for future research.

Foreword

While the segmentation of science into disciplines is practically useful (it would be unnecessary for every doctor to be trained in astrophysics), I envision the borders between disciplines more like the Schengen Area than the Korean Demilitarized Zone. Therefore, I was not content with just a "Tourist visa" in artificial intelligence; I aimed for full residency. Consequently, this thesis is likely atypical for a medical doctor, more technical than medical, but I hope it is no less valuable.

My initial interest was in predicting pseudarthrosis, but limited data availability led me to pivot. Instead, I created a novel dataset of bone X-ray images and their corresponding French-language radiology reports, which I then used to develop and apply self-supervised learning (SSL) techniques. In doing so, I laid the groundwork for future research, creating the resource I had initially wished I had at the beginning of my thesis.

The objective of this thesis evolved multiple times, reflecting both my growing knowledge and rapid advancements in the field. The fast-paced developments often made the work feel like it was built on quicksand. For instance, the state of the art in self-supervision shifted from auto-encoders to Generative adversarial networks, and then to contrastive methods. If I were to start over, I would approach it very differently, knowing that in another six months, the landscape would shift again. Even large language models like GPT4 or Llama 3.1, which are now unavoidable, seemed almost impossible at the time.

This journey has underscored the importance of adaptability and the continuous integration of emerging technologies. Each pivot in focus not only challenged my understanding but also deepened my appreciation for the fluidity and interconnectedness of modern scientific disciplines. Through this dynamic process, I aimed to create a thesis that reduce the gap between technical innovation and medical application, providing a foundation for future explorations at this interdisciplinary frontier.

Remerciements

Alors que ce travail touche à sa fin, il est pour moi temps d'exprimer toute la gratitude que je ressens envers ceux qui m'ont accompagné et soutenu dans cette aventure.

Je précise que la liste des personnes mentionnées ici n'est pas exhaustive, et ma plus grande crainte est d'en oublier. Si c'est le cas, je leur adresse mes remerciements les plus sincères.

Mes premiers remerciements vont naturellement à mes deux (co-)promoteurs, les Professeurs Christophe De Vleeschouwer et Olivier Cornu. J'ai rencontré Olivier Cornu il y a plus de dix ans, lors de mes premiers pas en tant qu'étudiant-chercheur au laboratoire CARS (Computer Assisted and Robotic Surgery). Depuis, il a toujours pris le temps de répondre à mes questions et de partager ses conseils, malgré un emploi du temps chargé. Il a soutenu mon projet de thèse, a joué un rôle clé dans mon parcours et a toujours cru en moi, même lorsque moi-même doutais. Pour cela, et pour bien d'autres raisons, un immense merci.

Ce travail aurait été bien différent sans Christophe. Je suis arrivé dans son bureau en tant que médecin, déterminé à m'investir dans un projet touchant au domaine des ingénieurs, une ambition bien utopiste. Pourtant, il ne s'est pas découragé : il m'a non seulement donné une chance, mais aussi accueilli dans son équipe, avec la même attention que pour n'importe quel autre doctorant en ingénierie. Nos discussions du lundi matin, où j'arrivais avec mes problèmes du moment et repartais avec des idées nouvelles, m'ont énormément appris. Je n'ai pas assez de mots pour lui exprimer ma gratitude, alors je dirai simplement : merci.

Je tiens également à remercier les membres de mon jury. Tout d'abord, mes membres externes, le Docteur Maxime Taquet et le Professeur Jean-Philippe Thiran, qui, malgré l'absence de lien préalable, ont accepté de consacrer un temps précieux à mon travail et m'ont fait des retours des plus pertinents. Un immense merci également au Professeur Frédéric Lecouvet, pour sa bienveillance en tant que président du jury, et au Professeur Thomas Kirchgesner, dont l'expertise radiologique était indispensable pour une thèse portant sur l'analyse de radiographies. Merci également au Professeur Benoit Macq, toujours débordant de pro-

jets et d'un optimisme communicatif, au Professeur John Lee, pour nos échanges aussi plaisants que variés, et au Professeur Thomas Schubert, avec qui je partage de nombreux centres d'intérêt, de l'impression 3D aux canons rayés. Merci à tous pour votre contribution et votre soutien.

Mes remerciements s'étendent également à mes collègues, car j'ai eu la chance d'avoir de nombreux collègues formidables, ayant travaillé dans divers laboratoires et avec des personnes exceptionnelles.

À Louvain-la-Neuve, où j'ai passé le plus de temps, mes collègues du laboratoire ELEN m'ont toujours accueilli avec bienveillance, même avec mes excentricités : Antoine 1 et 2 (ou l'inverse), Anne-Sophie, Niels, Victor, Gabriel, Abolfalz, Baptiste, Vladimir, Benoît, Gilles, Dany, Eleonor, Simon, Sarah, Clément, Isabelle, Patricia et tant d'autres. Vous avez illuminé mes journées, et j'ai été heureux de partager ce parcours avec vous. Pardonnez mes blagues douteuses, même si je n'en regrette pas une seule… sauf peut-être celle où j'ai failli avoir des ennuis avec la boîte d'Antoine pour un prétendu "piratage" (en y repensant, je ne regrette pas vraiment non plus, mais sur le moment, je n'en menais pas large !).

Au département INGI, merci aussi à Amaury pour toutes nos discussions sur le NLP, ainsi qu'à son promoteur, le Professeur Sébastien Jodoigne.

Merci également à l'équipe du CISM, qui a rendu possibles mes expériences en gérant le cluster que j'ai souvent mis à contribution.

Sur Woluwe, je tiens à remercier tout particulièrement Julie et Robin, qui ont été comme une sœur et un frère d'armes dans ce parcours. J'ai été très heureux de collaborer sur vos thèses et de partager du temps précieux avec vous, autant professionnellement qu'en dehors. Et merci d'avoir soutenu vos thèses avant moi, ce qui m'a permis de profiter de toutes vos astuces pratiques pour préparer ma propre défense.

Dans les laboratoires NMSK et CHEX, je remercie Nicolas, "Pong", Julien, Hervé, Jean-Louis, Louise, Alexandre, Tim, Alain, Randy, Marine, Christine, Philippe, Lise, Daela, Julia, Gwen, Pascale, et bien d'autres que j'ai peut-être moins croisés mais qui ont tout autant compté.

Merci à Sedick à l'UMons, avec qui j'ai eu beaucoup de plaisir à collaborer, notamment sur les projets TRAIL et TIS, ainsi qu'à Otman, qui a rejoint l'équipe.

Je remercie également Olivier Cartiaux, mon promoteur de mémoire, qui a accompagné mes premiers pas dans le monde de la recherche.

Un merci spécial à l'équipe de Sciense, avec qui de beaux projets sont encore en gestation : Sami, Alexandre, Salim, Mejdi.

Ce travail n'aurait pas été possible sans le soutien financier du Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) et du FRIA (Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture). Merci sincèrement pour votre confiance.

Enfin, je tiens à remercier ceux qui m'ont soutenu au-delà du cadre professionnel.

À mes amis, qui m'ont aidé à garder un équilibre : les soirées, les vacances partagées, et ces moments où je pouvais vous confier mes craintes, mes doutes et mes joies. Merci d'être toujours présents dans ma vie.

À ma famille élargie, un immense merci pour votre soutien, et un clin d'œil spécial à Ghislain, dont les conseils et l'expérience de thèse m'ont aidé à prendre la décision de me lancer dans cette aventure.

Peut-être plus atypique, mais je souhaite également remercier mes deux chiens, Rox et Rocky. Leur présence a été d'un réconfort inestimable, et c'est avec eux, couchés à mes côtés, que j'écris ces derniers mots.

Pour conclure, il est de coutume dans les publications scientifiques de reconnaître le premier auteur comme principal contributeur et le dernier auteur comme superviseur principal. Dans cet esprit, la dernière personne que je tiens à remercier est Sarah, ma fiancée, qui partage ma vie depuis plus de dix ans. Si une personne mérite tout particulièrement ma gratitude, c'est bien toi. Pour ton soutien, tes encouragements, et pour m'avoir poussé lorsque cela était nécessaire. Merci, ma chérie, je t'aime.

Author's publication list

Peer-Reviewed Publications

Englebert, A., Cornu, O. & De Vleeschouwer, C. (2022, August). Backward recursive class activation map refinement for high resolution saliency map. In *2022 26th International Conference on Pattern Recognition (ICPR)* (pp. 2444-2450). IEEE.

Englebert, A., Stassin, S., Nanfack, G., Mahmoudi, S. A., Siebert, X., Cornu, O. & De Vleeschouwer, C. (2023). Explaining through Transformer Input Sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop* (pp. 806-815).

Stassin, S., Englebert, A., Albert, J., Nanfack, G., Versbraegen, N., Frénay, B., Peiffer G., Doh M., Riche N. & De Vleeschouwer, C. (2023). An Experimental Investigation into the Evaluation of Explainability Methods for Computer Vision. In *Proceedings of Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases Workshop (ECML PKDD).*

Englebert, A., Cornu, O. & De Vleeschouwer, C. (2024). Poly-CAM: High resolution class activation map for convolutional neural networks. *Machine Vision and Applications*, 35(4), 89.

Manon, J., Englebert, A. (Joint first authors), Evrard, R., Schubert, T. & Cornu, O. (2024). FixThePig: a custom 3D printed femoral intramedullary nailing for preclinical research applications. Frontiers in Bioengineering and Biotechnology, 12, 1478676.

Correspondence

Niset, A., El Hadwe, S., Englebert, A., Barrit, S. (2024). AI in emergency medicine: Building literacy or castles in the air. *The American Journal of Emergency Medicine.*

Manuscripts Under Review

Englebert, A., Collin A.-S., Cornu, O. & De Vleeschouwer, C. (2024). Using vision-language self-supervision to turn raw local hospital data into image analysis models: a bone radiographs case study. *Under review*

Englebert, A., De Vleeschouwer, C., Lecouvet, F. & Cornu, O. (2024). Bone radiography analysis and Deep learning biases. *Under review*

Englebert, A., Evrard, R. (Joint first authors), Cornu, O. & Schubert, T. (2024). 3D Printed Design of a Custom Bioreactor for Large Bone Graft Recellularization. *Under review*

Lambricht, N., Englebert, A., Nguyen, A. P., Pitance, L., Fisette, P., Detrembleur, C. (2024). Accuracy and clothing effects in smartphone-based 2D joint kinematics assessment during running using OpenPifPaf. *Under review*

Contents

Conclusion

List of Figures

1 Introduction

1.1 The AI revolution

The Fourth Industrial Revolution, as coined by Klaus Schwab, founder and executive chairman of the World Economic Forum, is transforming many aspects of our lives, including healthcare [138]. This revolution, characterized by the widespread adoption of artificial intelligence (AI), 3D printing, and the internet of things, is happening at an unprecedented pace, the fastest change humanity has ever seen. At the heart of this revolution is AI, which has been dubbed the "new electricity" by Andrew Ng [115]. Just as electricity transformed industries and revolutionized the way we live and work, AI is poised to have a similar impact on many sectors, including healthcare. According to a 2023 report by the McKinsey Global Institute, about 30% of hours currently worked across the US economy could be automated [39].

As illustrated in Figure 1.1, AI systems have already demonstrated impressive capabilities relative to human performance in various areas, highlighting the potential for significant impact across industries.

Healthcare is no exception, and the impact of AI is also likely to be significant, building on the progress of previous industrial revolutions. It's probably impossible to draw up an exhaustive list of the practical impacts that technology has already had, and continues to have, on human life. But a simple figure that can represent the importance of this impact is life expectancy, which has increased for almost two decades as never before in history, as shown in figure 1.2. Today, AI is driving the next wave of innovation in healthcare. Just as modern surgery is unimaginable without respirators, monitoring systems, electric scalpels, and advanced lighting, future healthcare will be transformed by AI-powered technolo-

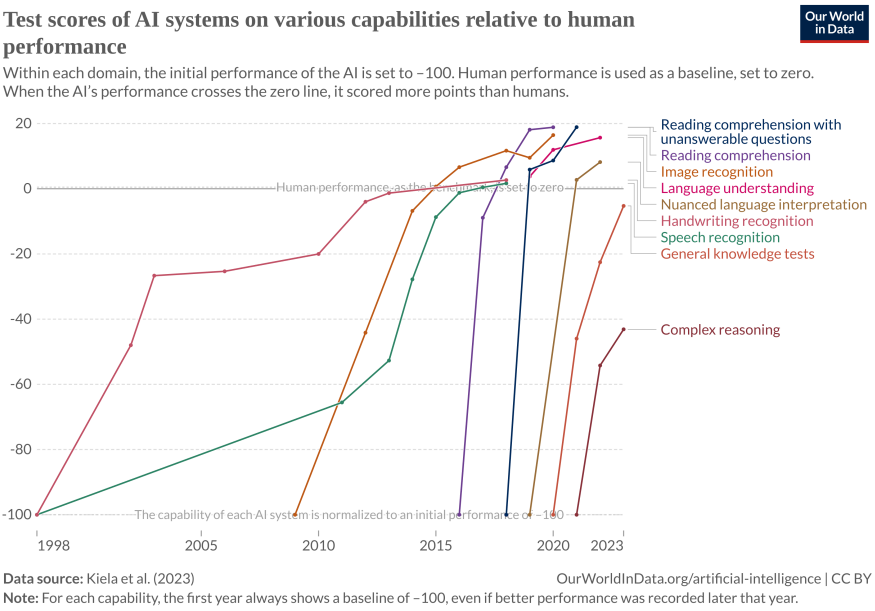


Fig. 1.1 Test scores of AI systems on various capabilities relative to human performance. source: `https://ourworldindata.org/grapher/test-scores-ai-capabilities-relative-human-performance`.

gies that improve patient outcomes, streamline clinical workflows, and enhance the overall quality of care.

1.2 AI and medicine

The rapid development of artificial intelligence (AI) in medicine has already affected the healthcare landscape [98], promising to transform not only diagnosis [54, 133, 140] and treatment [175, 151], but also in research [72], drug discovery [13, 16] and medical education [74]. In recent years, AI-powered systems have demonstrated remarkable capabilities in image recognition, natural language processing and predictive analytics, improving the accuracy and efficiency of a variety of applications. However, the development and deployment of AI models in medicine is often hampered by significant challenges, including the need for high-quality annotated datasets and in-depth domain expertise.

Unlike the development of AI in other fields, which has been facilitated by the availability of large, publicly accessible datasets such as ImageNet [136] or COCO [94] for image analysis, or web crawl based datasets [169] for large language models, medical data is often sparse, fragmented and protected by confidentiality rules [109, 154]. This disparity highlights the need to find innovative

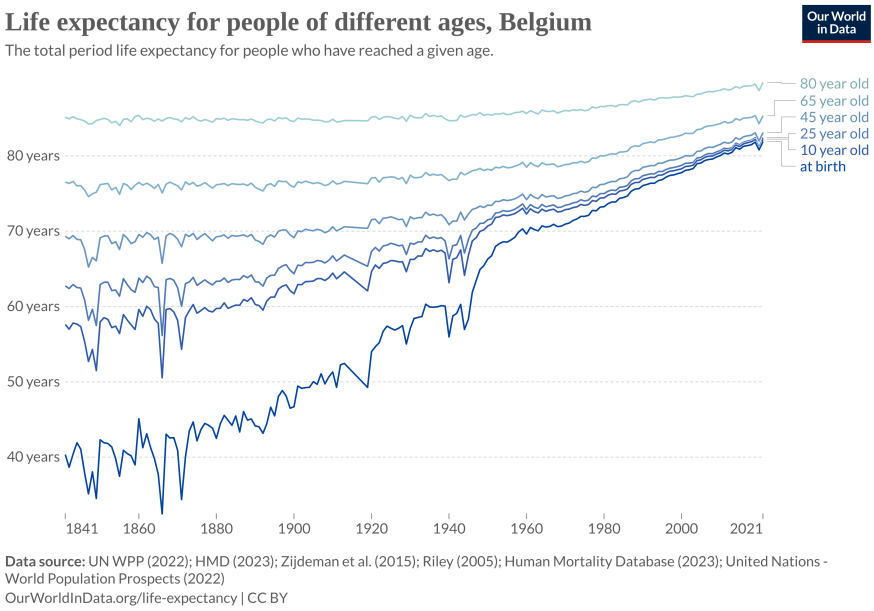


Fig. 1.2 Life expectancy at different ages, Belgium. source: `https://ourworldindata.org/grapher/life-expectancy-at-different-ages?country=~BEL`.

solutions to the unique challenges of AI development in medicine.

Furthermore, the increasing complexity of AI models has raised concerns about their transparency and interpretability [3, 111], making it difficult for healthcare professionals to understand the decision-making processes behind AI-driven diagnoses and recommendations [160, 66]. In contrast, classical statistical models, such as linear regression and decision trees, have traditionally been more transparent and interpretable, allowing healthcare professionals to understand the relationships between variables and the underlying assumptions of the models. Nevertheless, these classical models are unable to perform the full range of tasks that more advanced models can, and cannot learn the complex relationships that these newer models can grasp.

Uncertainty estimation techniques, such as test-time augmentation, Monte Carlo (MC) dropout, and ensembling, have been proposed to address this issue. Test-time augmentation applies transformations to the input data, helping reflect prediction robustness [164]. MC dropout approximates Bayesian inference by dropping units at inference, generating multiple predictions to estimate uncertainty [46]. Ensemble methods produce multiple model outputs, using the variance among them as an uncertainty measure [78]. These techniques provide healthcare professionals with a sense of confidence in model outputs, which can improve decision-making. However, they fall short of fully addressing the need

for interpretability.

A lack of interpretability can hinder the identification and correction of biases in AI models [79]. This is particularly problematic in healthcare, where biased or poorly understood models can lead to serious consequences, such as misdiagnosis or delayed treatment. Explainable AI (XAI) approaches are thus essential for bridging these gaps, enabling clinicians to better understand AI-driven insights and fostering a collaborative relationship between human expertise and machine intelligence. By enhancing interpretability alongside uncertainty estimation, AI can support more accurate, transparent, and efficient medical decision-making.

In the context of bone radiography, the challenges are even more pronounced. The availability of large annotated datasets for bone radiography is limited compared to other radiographic modalities, such as chest X-rays [127, 1, 62, 70]. This data scarcity hinders the development of accurate and robust AI models.

In addition, developing AI models capable of accurately analyzing reports written in languages other than English is a major challenge [30]. While most AI models are trained on English datasets, they may not generalize well to other languages, including French, which is the language used in hospitals in the Walloon region, where this thesis takes place. The development of AI models capable of accurately analyzing reports written in French is essential for widespread adoption in French-speaking countries.

To address these challenges, there is a growing need for innovative solutions that can facilitate the development and deployment of transparent, trustworthy, and robust AI models in medicine. This thesis aims to contribute to this effort by exploring two critical axes: explainability and vison-language self-supervision. By developing novel explainability methods and adapting self-supervision techniques to bone radiographic data coupled with French reports, this research seeks to improve the transparency and robustness of AI models in medicine, ultimately reducing the gap between technical innovation and medical application.

1.3 Research objectives

This thesis is guided by two main lines of research, which attempt to address the constraints of the medical world.

1.3.1 Explainability

Explainability refers to the ability of an artificial intelligence (AI) model to make its decision-making process transparent and understandable to humans, who may be patients, doctors or the people developing the AI.

This research line aims to improve the transparency and trustworthiness of AI models in medicine, enabling a medical doctor to understand and adapt to the evidence-based decisions made by these models.

Research Question #1

How can we develop explainability methods that provide insights into the decision-making processes of artificial intelligence models ?

To answer this question, a literature review is first carried out in Chapter 3, followed by the exploration and development of the Poly-CAM method for convolutional neural networks in Chapter 4 and the Transformer Input Sampling method in Chapter 5.

Research Question #2

Can explainable methods effectively uncover and help to mitigate biases in artificial intelligence model training ?

This question is explored in Chapter 6 by applying the Poly-CAM methods developed in this thesis to a public bone radiograph dataset to gain more insight into the inner workings of the models and explore the existing biases and potential spurious correlations acquired during training.

1.3.2 Self-Supervision and Vision-Language

This research line seeks to explore the potential of self-supervision in addressing the scarcity of annotated medical data, particularly in the context of bone radiographic examinations and French radiologic reports.

Research Question #3

Can self-supervision techniques be adapted to utilize the inherent supervision within bone radiographic data and associated French reports ?

A literature review is first performed in Chapter 7, followed by the exploration in Chapter 8 of vision-language pretraining using various text encoders more adapted to French than classical models used in the literature for English-based medical reports.

Research Question #4

How can these methods be optimized to reduce the need for costly annotations in medical imaging ?

Chapter 8 not only explores vision-language pretraining, but also investigates the automatic generation of pseudo-labels to reduce the need for annotations. Chapter 9 further develops this idea by generating standardized re-

port and question-answer pairs, laying the groundwork for training a vision-language model for reports generation and visual question answering on bone radiographs.

1.4 Outline

This thesis is organized as follows. Chapter 2 introduces the deep learning background. The remaining of the work is split in two main parts for explainability and vision-language self-supervised learning.

The first part focuses on explainability and is composed of the following chapters: Chapter 3, which introduces the fundamentals of explainable AI, Chapter 4, which proposes our Poly-CAM method to explain convolutional neural networks, Chapter 5, which provides the Transformer Input Sampling method for transformer models, and Chapter 6, which explores the usage of explainable methods on bone radiographs.

The second part explores self-supervised vision-language learning, with Chapter 7 introducing the fundamentals, Chapter 8 explaining the construction of our dataset and the vision-language pretraining, while Chapter 9 introduces preliminary explorations to continue this work on report generation and visual question answering.

Research Question #4

How can these methods be optimized to reduce the need for costly annotations in medical imaging ?

Chapter 8 not only explores vision-language pretraining, but also investigates the automatic generation of pseudo-labels to reduce the need for annotations. Chapter 9 further develops this idea by generating standardized report and question-answer pairs, laying the groundwork for training a vision-language model for reports generation and visual question answering on bone radiographs.

1.5 Outline

Finally, Chapter 10 concludes this work.

2

Deep learning background

This section will introduce the general concepts necessary for a thorough understanding of the remaining manuscript. Concepts specific to explainable AI and self-supervision will be introduced in their respective chapters.

2.1 What is deep learning?

This thesis will make extensive use of deep learning, so it is essential to start by introducing what deep learning is. We can begin by stating that deep learning is a machine learning method. And that machine learning is a type of artificial intelligence, so deep learning is also a type of artificial intelligence, as shown in Figure 2.1. As the reader may feel confused, let's describe this further.

Definition 2.1. Artificial Intelligence
The science of making computers do things that human beings can do[1].

This definition is quite vague and does not explain how this is achieved or what the "things" are that human beings can do. Artificial intelligence therefore encompasses a very broad field: a system based on predefined rules can easily be implemented in a form that qualifies as artificial intelligence. The Logic Theorist [114], a program designed in 1956 to mimic human problem-solving, is often considered the first artificial intelligence program. A more concrete example for the reader could be the old GPS navigation system left lying in your basement, since even the first rule-based GPS navigation system introduced by Honda in 1990 [55] is already an example of artificial intelligence, although far from the kind of AI used in this thesis.

[1] https://dictionary.cambridge.org/dictionary/english-french/artificial-intelligence

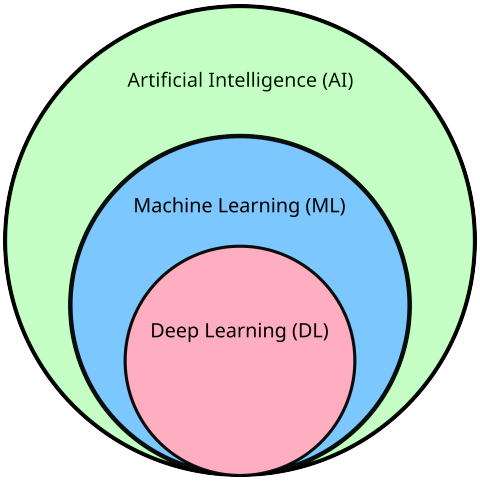


Fig. 2.1 From Artificial Intelligence to Deep Learning: nomenclature.

Definition 2.2. Machine Learning
The process of computers improving their own ability to carry out tasks by analyzing new data, without a human needing to give instructions in the form of a program, or the study of creating and using computer systems that can do this.[2]

This takes us a step further, as we don't need to manually plan rules for all the possible cases the software would encounter, but rather design an algorithm to learn these rules based on data. These methods are often based on statistics, and a classical linear regression can be seen as a kind of simple machine learning algorithm. Many methods have been developed, such as Bayesian inference, Support-vector machines (SVMs), Random forest, k-NN, and of course, neural networks. We will not develop all available machine learning methods, otherwise, this work would require a painkiller prescription. We will instead focus on neural networks, and more specifically on deep neural networks, which will bring us to our next definition.

Definition 2.3. Deep Learning
a type of machine learning (= the process of computers improving their own ability to perform tasks by analyzing new data) that uses many layers of data processing.[3]

Many methods have been developed in the field of machine learning, and deep learning is one of them. It is often used for tasks such as image recognition, speech recognition, and natural language processing. Deep learning typically involves the use of neural networks, which are composed of multiple layers

[2]https://dictionary.cambridge.org/dictionary/english/machine-learning
[3]https://dictionary.cambridge.org/dictionary/english/deep-learning

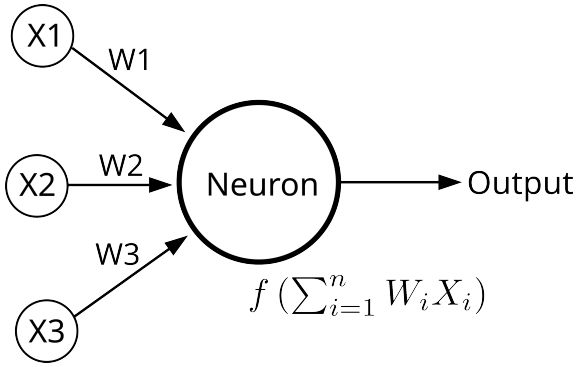


Fig. 2.2 Logical neuron.

of interconnected nodes (neurons) that process and transmit information. These neural networks are designed to recognize patterns in data and learn from it, allowing them to perform complex tasks such as image recognition and natural language processing.

The term "deep" in deep learning refers to the use of multiple layers in these neural networks. The more layers, the deeper the network. This allows the network to learn more abstract and sophisticated representations of the data, enabling it to perform tasks such as image recognition, natural language processing, and speech recognition.

The following section will delve deeper into the workings of neural networks and deep learning, providing the reader with a more detailed understanding of how these complex systems operate.

2.2 Neural Networks

Neural networks are a fundamental component of deep learning. They are composed of multiple layers of interconnected nodes, or neurons, that process and transmit information. Each neuron receives one or more inputs, performs a computation on those inputs, and then sends the output to other neurons. This process allows the network to learn and represent complex patterns in data.

The concept of neural networks dates back to the 1940s, when Warren McCulloch and Walter Pitts introduced the first artificial neural network model, laying the foundation for the development of neural networks [106]. Building on this work, the perceptron, a single-layer neural network, was first introduced by Frank Rosenblatt in the 1950s [134], illustrated in Figure 2.2. However, it was not until the 1986 paper by David Rumelhart, Geoffrey Hinton, and Ronald Williams that the concept of multi-layer perceptron (MLP), illustrated in Figure 2.3, and the backpropagation algorithm for training them were introduced [135]. This

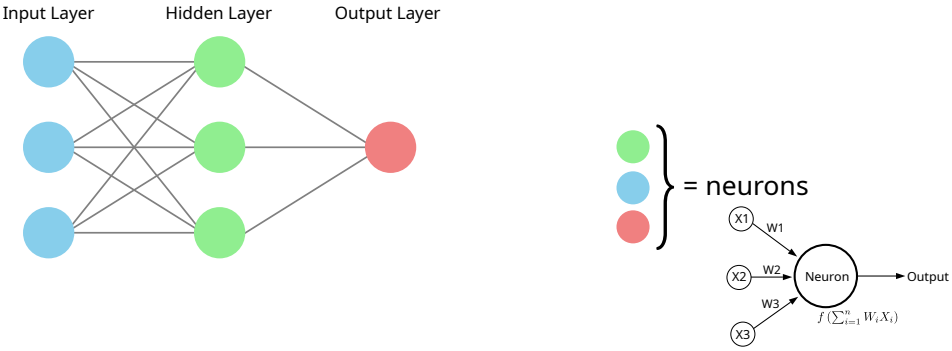


Fig. 2.3 Multilayer perceptron.

breakthrough enabled the training of deeper and more complex neural networks, which has since become a cornerstone of modern deep learning techniques.

Over the years, the development of neural networks has been marked by significant milestones, including the introduction of convolutional neural networks (CNNs) [45], recurrent neural networks (RNNs) [58] and Long short-term memory networks (LSTMs) [56], and lately, transformers [161], among many others architectures. These advancements, combined with the availability of large amounts of data and advances in computing power, have enabled the training of increasingly complex and powerful neural networks. Today, neural networks are a fundamental component of many state-of-the-art machine learning systems, with applications in computer vision, natural language processing, speech recognition, and more.

The next sections will dive into the details of deep learning models. To allow less technical readers to skip the details while still grasping the big picture, a simplified summary will be provided in boxes called *In Simple Terms* at the beginning of each section.

2.3 Activation Functions

In Simple Terms

An activation function in deep learning is like a decision-making tool for a neuron that helps it decide whether to "fire" or stay inactive, similar to how our brain decides to act on certain signals. It enables the model to learn and understand complex patterns by adding non-linearity, making it capable of solving difficult tasks.

Activation functions introduce non-linearity into the model, allowing it to learn more complex patterns. Without activation functions, a neural network,

regardless of the number of layers, would behave like a single-layer perceptron.

For illustration purpose, some common activation functions include:

Sigmoid Maps any real-valued number into the range (0, 1), useful for binary classification tasks.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

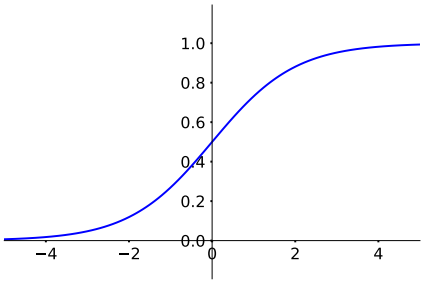


Fig. 2.4 Sigmoid function.

Tanh Maps any real-valued number into the range (-1, 1). It is a scaled version of the sigmoid function.

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

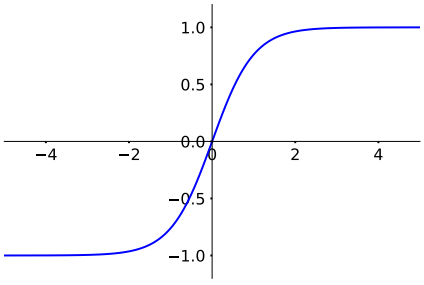


Fig. 2.5 Tanh function.

ReLU (Rectified Linear Unit) [44] One of the most commonly used activation functions in deep learning because of its effectiveness and simplicity, defined as:

$$\text{ReLU}(x) = \max(0, x)$$

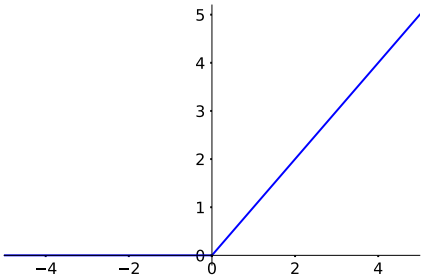


Fig. 2.6 ReLU function.

Leaky ReLU [103] An extension of ReLU that allows a small gradient when the input is negative:

$$\text{Leaky ReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases}$$

where α is a small constant.

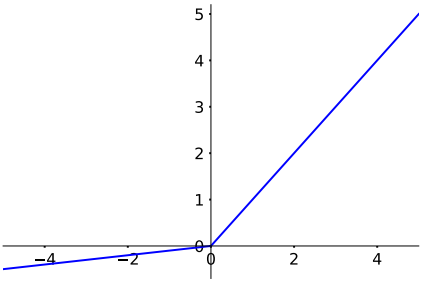


Fig. 2.7 Leaky ReLU function.

Other variations exist, such as GELU [53], ELU [28] or SILU [38], but developing them in greater detail would be of little benefit to the rest of this work.

2.4 Backpropagation and Loss Functions

In this section, we will explore the mechanisms that enable neural networks to learn from data: backpropagation and loss functions. Understanding these concepts is essential to grasp how deep learning models are trained and optimized.

2.4.1 Backpropagation

In Simple Terms

Backpropagation is a learning process where a deep learning model adjusts its internal settings (weights) to improve its accuracy. It works by comparing the model's prediction to the actual answer, calculating the difference (error) using the loss function, and then tweaking the settings to reduce this error for better future predictions through an optimization algorithm.

Backpropagation, short for "backward propagation of errors", is a fundamental algorithm for training neural networks [135]. Backpropagation efficiently computes gradients that are used to update the model parameters during the training process.

The primary goal of backpropagation is to minimize a loss function, which quantifies the difference between the network's predictions and the actual target values. The algorithm works by propagating the error backward through the layers of the network, allowing the computation of gradients for each parameter.

The backpropagation algorithm consists of four main steps:

1. **Forward Pass:** Compute the output of the network for a given input by passing the input through each layer. This involves applying weights, biases, and activation functions at each layer to obtain the final output.
2. **Compute Loss:** Evaluate the loss function by comparing the predicted output with the actual target value. This step quantifies how well the network is performing.
3. **Backward Pass:** Compute the gradient of the loss function with respect to each parameter in the network. This is done by applying the chain rule of calculus, which involves calculating the partial derivatives of the loss function with respect to each parameter and propagating these gradients backward through the network.
4. **Update Parameters:** Adjust the network parameters (weights and biases) using the computed gradients and an optimization algorithm (e.g., stochastic gradient descent). The goal is to reduce the loss function by moving the parameters in the direction that decreases the loss.

The backpropagation algorithm iteratively repeats these steps for multiple epochs (passes through the entire training dataset) until the loss converges to a minimum value.

To make it simple, backpropagation fine-tunes the model's parameters step by step, helping it learn from mistakes and make more accurate predictions.

2.4.2 Loss Functions

In Simple Terms

The loss function measures how far off the model's predictions are from the correct answers. It's like a scorecard that tells the model how well it's doing, and the goal is to minimize this score by making more accurate predictions over time.

A loss function, also known as a cost function or objective function, measures the difference between the predicted output of a neural network and the true target values. The choice of loss function depends on the type of task, such as regression or classification. A crucial requirement for a loss function in deep learning is that it be differentiable. This differentiability makes it possible to calculate the gradients essential for the backpropagation algorithm to update the model parameters. Here are some common loss functions used in deep learning:

Mean Squared Error (MSE) / L2 Loss Used primarily for regression tasks, the MSE loss function measures the average squared difference between predicted and actual values.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and N is the number of samples.

Mean Absolute Error (MAE) / L1 Loss Another common loss function for regression tasks, MAE measures the average absolute difference between predicted and actual values.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and N is the number of samples.

Cross-Entropy Loss Commonly used for classification tasks, cross-entropy loss measures the difference between the true probability distribution and the predicted probability distribution. This concept is rooted in information theory [142].

$$\text{Cross-Entropy Loss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

where y_i is the true binary label and $\hat{y}_i$ is the predicted probability.

There are many other loss functions, each tailored to specific use cases and data types. Choosing the appropriate loss function is crucial for the successful training of a neural network, as it directly impacts the optimization process and the model's ability to generalize to new data.

In other words, loss functions help the neural network understand how close its predictions are to the actual results. By measuring the error, the loss function guide the model to adjust and improve its predictions.

2.4.3 Optimization Algorithms

In Simple Terms

An optimization algorithm is a method that helps the model find the best settings (weights) to reduce errors and make more accurate predictions. It works by repeatedly adjusting the model's settings in small steps, guided by the loss function, until the model performs as well as possible.

Optimization algorithms are methods used to adjust the weights and biases in neural networks to minimize the loss function. Here, we'll cover some of the most commonly used optimization techniques in a straightforward manner.

Stochastic Gradient Descent (SGD) [132] SGD is a simple and widely used method. It updates network parameters step by step for each training example:

$$\theta = \theta - \eta \cdot \frac{\partial L}{\partial \theta}$$

Here, θ represents the model parameters (weights and biases), η is the learning rate (a small number that controls how big the steps are), and L is the loss function. Instead of computing the gradient over the entire dataset, SGD updates the parameters after looking at each individual data point, making it faster but potentially noisier.

Mini-Batch Gradient Descent Mini-Batch Gradient Descent is a middle ground between using all data points (batch gradient descent) and one data point at a time (SGD). It splits the dataset into small batches and updates the parameters using each batch. This approach balances speed and noise. However, in the literature, the term SGD is often used interchangeably with mini-batch gradient descent.

Momentum [121] Momentum improves SGD by keeping track of past updates and adding a fraction of the previous update to the current update. This helps accelerate learning and dampen oscillations:

$$v_t = \gamma v_{t-1} + \eta \cdot \frac{\partial L}{\partial \theta}$$
$$\theta = \theta - v_t$$

Where v_t is the momentum term at time t, and γ is a parameter that controls how much of the previous update is used.

Adaptive Learning Rate Methods These methods adjust the learning rate for each parameter independently, helping the network to converge faster and more efficiently.

1. **AdaGrad** [37]: This method adjusts the learning rate for each parameter based on how frequently it is updated. Parameters that are updated often get smaller learning rates.

   $$\theta = \theta - \frac{\eta}{\sqrt{G_t} + \epsilon} \cdot \frac{\partial L}{\partial \theta}$$

   Where G_t is the sum of the squares of past gradients, and ϵ is a small number to prevent division by zero.

2. **RMSprop** [155]: Similar to AdaGrad, but it uses a moving average of squared gradients to adjust the learning rate, making adjustments more stable over time.

   $$G_t = \beta G_{t-1} + (1 - \beta) \left(\frac{\partial L}{\partial \theta} \right)^2$$
   $$\theta = \theta - \frac{\eta}{\sqrt{G_t} + \epsilon} \cdot \frac{\partial L}{\partial \theta}$$

   Where β controls the moving average.

3. **Adam (Adaptive Moment Estimation)** [73]: Adam combines the benefits of both AdaGrad and Momentum. It calculates adaptive learning rates for each parameter and takes an average of past gradients and squared gradients.

   $$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \frac{\partial L}{\partial \theta}$$
   $$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left(\frac{\partial L}{\partial \theta} \right)^2$$
   $$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta = \theta - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t$$

Where β_1 and β_2 control the decay rates of the moving averages.

LION (Layer-wise Optimizer for Neural Networks) [26] LION is a recent optimizer that builds on the Evolved Sign Momentum (ESM) approach, designed to be efficient and scalable for large models. Instead of relying directly on gradient values, LION updates parameters using the sign of the moving average of past gradients, which helps reduce noise and stabilize updates. This method is particularly useful for deep learning tasks where traditional gradient-based methods might struggle with oscillations or instability.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \cdot \frac{\partial L}{\partial \theta}$$
$$\theta_t = \theta_{t-1} - \eta \cdot \text{sign}(m_t)$$

Where m_t is the moving average of past gradients, η is the learning rate, and β_1 controls the momentum. The use of $\text{sign}(m_t)$ allows the optimizer to make consistent updates while mitigating oscillations, contributing to faster and more stable convergence.

Choosing the right optimization algorithm is essential for efficiently training a neural network. The right optimizer can help the network learn faster and perform better on new, unseen data.

2.5 Particular Architectures

Several specialized neural network architectures have been developed to tackle specific types of data and tasks more effectively. In this section, we'll cover some of the architectures that are relevant to this thesis.

2.5.1 Convolutional Neural Networks (CNNs)

In Simple Terms

A Convolutional Neural Network (CNN) is a type of deep learning model designed to process and recognize patterns in visual data, like images. It uses special layers called convolutional layers to automatically detect important features, such as edges or textures, and gradually builds up a more complex understanding of the image, allowing it to accurately classify or identify objects within it.

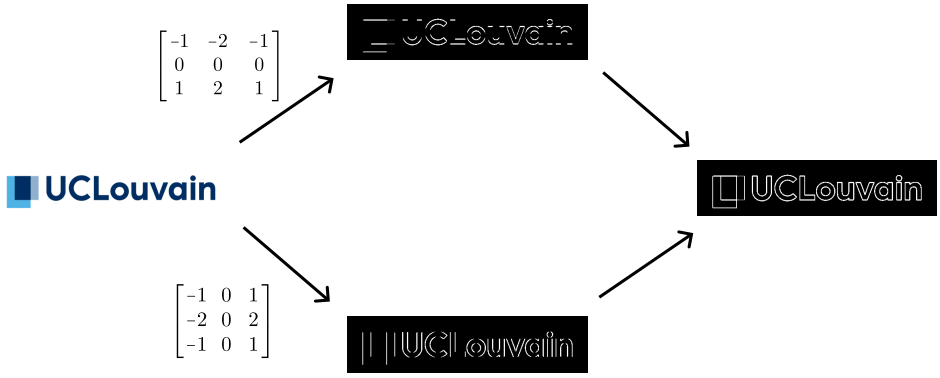


Fig. 2.8 Illustration of a convolution with a Sobel filter.

Convolutional Neural Networks (CNNs) are a class of deep neural networks particularly well-suited for processing grid-like data, such as images [82]. They are designed to automatically and adaptively learn spatial hierarchies of features from input images through the use of convolutional layers, pooling layers, and fully connected layers.

Convolutional Layers

The convolutional layer is the core building block of a CNN. It consists of a set of learnable filters (or kernels) that are applied across the width and height of the input image to produce feature maps. Each filter can be thought of as a sliding window that captures specific local patterns in the input image.

To illustrate this, consider the Sobel filter, a well-known edge detection filter used in image processing [150]. The Sobel filter applies convolution to compute the gradient of the image intensity, helping to identify edges. The following matrices show the Sobel filters for detecting horizontal and vertical edges:

$$\text{Horizontal Sobel Filter} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\text{Vertical Sobel Filter} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

When these filters are convolved with an input image, they highlight the edges by emphasizing regions of high spatial gradient. A convolutional layer inside of a CNN works similarly, except that the matrices are learnable parameters.

Pooling Layers

Pooling layers are used to reduce the spatial dimensions (width and height) of the feature maps, thereby decreasing the number of parameters and computation in the network. Common types of pooling include max pooling and average pooling.

- **Max Pooling:** Selects the maximum value within a specified window (e.g., 2x2) and discards the rest.

- **Average Pooling:** Computes the average value within a specified window.

Fully Connected Layers

Fully connected layers are typically used towards the end of the network to perform high-level reasoning. These layers connect every neuron in one layer to every neuron in the next layer, enabling the network to combine features learned at different levels. A fully connected layer is only a synonym for a single layer perceptron.

Residual Networks (ResNets)

Residual Networks, or ResNets, introduced by He et al. [51], address the problem of vanishing gradients in deep networks by introducing skip connections (or shortcuts). These connections allow the gradient to bypass one or more layers, making it easier to train very deep networks.

The core idea of ResNets is to learn residual functions with reference to the layer inputs. Instead of learning the mapping $H(x)$, the network learns the residual mapping $F(x) = H(x) - x$. Thus, the original mapping becomes $H(x) = F(x) + x$.

A block in a ResNet looks like this:

$$\text{Output} = F(x) + x$$

Where $F(x)$ represents the residual mapping learned by the convolutional layers. Skip connections help mitigate the vanishing gradient problem and enable the training of much deeper networks.

2.5.2 Transformers

In Simple Terms

A Transformer is a deep learning model that excels at understanding and processing sequences of data, like sentences. Its key feature is attention, which allows it to focus on the most important parts of the input, even if they're far apart, helping the model capture context and meaning more accurately. This makes Transformers particularly powerful for tasks like language translation and text generation. Transformers have also been adapted for image processing, where they use attention mechanisms to analyze different parts of an image and understand complex visual patterns.

Transformers, introduced by Vaswani et al. [161], have revolutionized natural language processing (NLP) and have been increasingly applied to other domains such as computer vision. Transformers rely on the mechanism of self-attention to model relationships between elements in a sequence, regardless of their distance from each other.

Self-Attention Mechanism

Consider an orthopedic report describing a patient's condition, such as "The patient has a fracture of the distal radius". To accurately interpret the condition, a model should understand how terms such as "fracture" and "distal radius" relate to each other. The self-attention mechanism allows each word or token in an input sequence to focus on other relevant words, capturing dependencies across the sequence.

The self-attention mechanism allows each element of an input sequence to focus on other elements to compute a representation. The attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors.

The attention score for a query q and a key k is computed using the dot product, followed by a softmax function:

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

Where Q, K, and V are the query, key, and value matrices, respectively, and d_k is the dimensionality of the keys.

Transformer Architecture

Transformers as originally described consist of an encoder-decoder structure, where both the encoder and decoder are composed of multiple layers of self-attention and feedforward neural networks. The encoder processes the input sequence,

while the decoder generates the output sequence, one token at a time. In this context, a token is a basic unit of text, such as a word, character, or subword. Although initially designed for text, transformers have been generalized to other modalities, where tokens represent other input units, like image patches.

Key components of the transformer architecture include:

- **Multi-Head Attention:** Allows the model to jointly attend to information from different representation subspaces.
- **Feedforward Neural Networks:** Applied to each position separately and identically. This is a two layer MLP.
- **Positional Encoding:** Adds information about the position of the tokens in the sequence, since the model does not inherently capture this information.

Encoder and Decoder

The encoder and decoder in a Transformer have distinct roles and mechanisms. One key difference is in the attention mechanisms they employ.

Encoder: Bidirectional Attention The encoder utilizes bidirectional self-attention, meaning each token in the input sequence can attend to all other tokens, both before and after it. This allows the model to understand the context of each token in relation to the entire sequence.

Decoder: Unidirectional Attention The decoder, on the other hand, uses unidirectional (or causal) self-attention, where each token can only attend to previous tokens and not future ones. This is essential for autoregressive generation tasks, where predicting the next token should not be influenced by future tokens.

Additionally, the decoder has an extra cross-attention mechanism that allows it to attend to the encoder's output. This way, the decoder can generate the output sequence conditioned on the entire input sequence, enhancing the quality of the generated output.

Encoder-Only vs. Decoder-Only Architectures

In practice, many models use only the encoder or the decoder for specific tasks.

Encoder-Only Models (e.g., BERT) Bidirectional Encoder Representations from Transformers (BERT), introduced by Devlin et al. [35], is an example of an encoder-only architecture. BERT uses bidirectional self-attention to understand the context of each word in a sequence, making it particularly effective for tasks that require a deep understanding of text, such as question answering and language inference.

Decoder-Only Models (e.g., GPT) Generative Pre-trained Transformer (GPT), introduced by Radford et al. [125], is an example of a decoder-only architecture. GPT uses unidirectional self-attention, making it suitable for tasks that involve generating text, such as language modeling and text completion. The model generates text one token at a time, with each token attending only to the previous tokens.

2.6 Summary

In this chapter, we have provided an introduction to deep learning and its key components, including neural networks, activation functions, backpropagation, loss functions, optimization algorithms, and specialized architectures such as CNNs and transformers. These foundational concepts are crucial for understanding the methodologies and techniques employed in the remainder of this thesis. We skipped many aspects that have practical importance, but may not be relevant for the understanding of this work such as normalization or regularization.

PART I
Explainable AI (XAI): Dealing with the black box

3

Fundamentals of XAI

This Section will introduce concepts about explainable AI (XAI) required for a good understanding of the following chapters.

3.1 Introduction

Recent advances in deep learning create an increasing need for explanation techniques to evaluate the prediction quality of neural networks. This is especially required in areas where a black box model is not desired for ethical or security reasons such as deciding the treatment for patient or for granting a loan. In contrast to techniques based on handcrafted features, deep neural networks (DNN) often lack transparency and explainability [3].

The need to assess a posteriori the behavior of a model has led to the development of explainable artificial intelligence (XAI) methods, ranging from the development of more transparent model architectures [167] to post-hoc methods (explanation, by example of black-box methods).

Saliency maps visualization has been adopted as a convenient approach to identify the image parts justifying the network prediction. Those saliency maps are helpful to check that the predictions of a model are grounded on relevant information. It is indeed known that training convergence alone does not exclude undesired DNN predictions [79], typically because the model has learned inputs / outputs correlations that do not correspond to the desired meaningful causal relationship.

Alternatively, when sufficiently accurate, the localization of salient features could convince a user that a model works properly, i.e. uses relevant cues, or

could even help in identifying the parts of a signal that are relevant to solve a problem, e.g. help a medical doctor in identifying the X-ray visual cues that permit to anticipate the evolution of a treatment.

3.2 Related work

This section provides an overview of some common techniques used to interpret the decisions made by complex deep learning models, focusing specifically on post-hoc explanation methods applied to image data.

3.2.1 Explainability in Computer Vision

Gradient-based Methods

Among the first applicable methods to explain the results of deep learning models are the gradient-based methods. They explain the prediction of a model by performing a backpropagation from an output neuron (e.g., a probability obtained for a class) to the input features [145]. This produces a so-called saliency map (or heatmap), providing a visualization of the most important areas for the decision of black-box models. Smilkov et al. introduced **SmoothGrad** [149] which augments the input samples by adding Gaussian noise and calculates the average of the results obtained for each backpropagation. **Integrated Gradient** [153] also computes a backpropagation average, but the result is obtained based on an interpolation between the input image and a baseline image (e.g., black, white image).

Perturbation-based Methods

Next to the gradient-based methods, there are also methods that perturb the input image and analyze how the model response is impacted by those changes to produce an explanation (e.g., **Occlusion** [179] using square patches). Those methods are known as perturbation-based methods. **RISE** [120] is a popular state-of-the-art method that produces small random binary masks, then scaled to the size of the image. The saliency map is computed as a linear combination of the perturbation masks and their relevance, measured based on their impact on the prediction.

CAM-based Methods

Class Activation Maps-based methods (CAM) use the activations of the convolutional layers of CNNs to obtain saliency maps. The most popular method is **Grad-CAM** [139], which weights the activation maps by the gradients obtained by a backpropagation from the output neuron of a class to the last convolutional layer. Variants aggregate the results for the input image at different scales (**CAMERAS** [64]), combine the activations from different layers (**LayerCAM** [69]), or predict the relevance of masks created from the activations (**Score-**

CAM [166]). Since Vision Transformers employ the CLS token for downstream tasks, this limits the application of CAM methods that require the use of the embeddings before a last pooling layer.

3.2.2 Explainability of Vision Transformers

The key difference between a CNN and a transformer lies in the calculation of attention scores for the latter. These attention scores help in representing the relationships that can appear between each of the input features. Consequently, the first attempts to explain the results of visual attention were based on saliency maps created through an upsampling of these attention scores [174]. However, the use of attention scores as explainability scores has limitations [2, 123, 141] (e.g., attention takes into account the query and key, but not the value of the self-attention) that have led to specific explanation methods designed for transformers.

Attention-based Methods

The first one came from Abnar [2] who presented the **Attention Rollout** method. This approach computes the saliency map based on a combination (e.g., average; minimum; maximum) of the attention heads with the addition of an identity matrix representative of the residual connections, arguing that the latter is crucial to compute the propagation of information through the layers. However, this approach does not take into account the fact that some attention heads may be more relevant than others.

Gradient-based Methods

Partial LRP [162] solved this issue by calculating the importance of each attention head using the Layer-wise Relevance Propagation (LRP) [12] method. **Chefer 1** [23] argued that the use of LRP by [162] provided only partial information on the attention head relevance as the LRP rule was not utilized back to the input features. The Chefer method computes class-specific explanations by incorporating relevance (LRP) and gradient information with specific rules designed to handle the skip connections. **Chefer 2** [22] provided a generic solution that can be applied to any transformer-based architecture and to more than two modalities. The latter takes into account the residual connections through an identity matrix to compute attention scores (as proposed by [2]) and utilizes the gradients to obtain the relevance of each head related with respect to a desired class output. The **Transition Attention Maps (TAM)** [178] method takes inspiration from the Markov process. At each block, the representations of the output tokens are considered as states of the Markov chain, with the state transition matrix being constructed based on the attention weights. A class discriminative explanation is achieved by combining the states with the Integrated Gradients obtained with

respect to the last attention module. **Bidirectional Transformers (BT)** [24][1] compute an element-wise product between two terms to obtain a saliency map. The first is Reasoning Feedback. It represents how the classification token (CLS) is used for a class prediction and is calculated with the Integrated Gradients of a chosen class back to the last attention map using a black baseline. The second is Attention Perception. It represents the learning process of the input tokens through the attention blocks. It approximates the relationship between the input and output of the attention blocks and derives two attention maps from it: BT-T (T for token) and BT-H (H for head).

Perturbation-based Methods

ViT-CX [172] adopts a different approach compared to the previous transformer explainability methods. It no longer relies directly on attention weights and gradients but on masks created from patch embeddings (such as Score-CAM [166] using feature maps as masks for CNNs) and the relevance of each mask, computed by evaluating the model with a masked image to obtain a saliency map. This method is similar to perturbation-related methods such as RISE [120] but provides a smaller number of more focused masks because first they are not randomly generated but use transformer embeddings, and second ViT-CX adds a clustering of the embeddings to further reduce the number of masks.

3.3 Evaluation Metrics for XAI

To evaluate and compare various XAI methods, we can employ a range of metrics. These metrics are categorized into four distinct families based on the aspect of explainability they assess, as outlined in the Quantus Framework [52]. Additionally, we include the insertion and deletion metrics from the RISE paper [120], which will be utilized later in this thesis. The following list provides an overview of the types of metrics used, though it is not exhaustive.

Faithfulness Metrics

Faithfulness metrics assess how well the explainability method mirrors the model's predictive behavior. We consider seven such metrics:

- *Faithfulness Correlation* [15]: This metric partitions the input image into feature subsets, replaces them iteratively with a baseline value, and computes the Pearson correlation between the drop in classification probability for a target class and the sum of the relevance attributions for each subset.
- *Faithfulness Estimation* [10]: Similar to Faithfulness Correlation, it calculates the Pearson correlation between the drop in classification probability and

[1] The method is not named in the paper but is referred to as "Bidirectional Transformers" in InterpretDL (https://github.com/PaddlePaddle/InterpretDL).

feature relevance.

- *Monotonicity Metric Arya* [11]: Measures the increase in model performance (classification probability) when features of increasing importance are added.
- *Monotonicity Metric Nguyen* [116]: Also measures the increase in model performance, but through probability estimation uncertainty.
- *Pixel Flipping* [12]: Involves flipping pixels with high relevance scores from the relevance heatmap and observing the evolution of the probability score for a target class.
- *Region Perturbation* [137] and *Selectivity* [108]: Extend the methodology of Pixel Flipping to areas of an image.
- *Insertion* [120]: This metric evaluates the effect of incrementally adding the most relevant features (according to the explainability method) back into the input and observing the change in model output. The idea is to check how quickly the model's confidence recovers as important features are reintroduced.
- *Deletion* [120]: Conversely, this metric measures the impact of progressively removing the most relevant features from the input and monitoring the decrease in the model's output. This helps to assess the importance of the features by noting how quickly the model's confidence drops when key features are removed.

Robustness Metrics

Robustness metrics evaluate the stability of explanations under small input perturbations. We consider three such metrics:

- *Local Lipschitz Estimate* [9]: Measures the consistency of explanations for adjacent samples.
- *Max-Sensitivity* and *Avg-Sensitivity* [177]: Quantify the maximum and average change in explanations when inputs are infinitesimally perturbed, using a Monte Carlo sampling-based approximation.

Complexity Metrics

Complexity metrics assess the conciseness of the explainability method. We consider three such metrics:

- *Sparseness* [20]: Uses the Gini index to determine if only highly attributed features are predictive of the model output.
- *Complexity* [15]: Measures the entropy of the fractional contributions to the total attribution.

- *Effective Complexity* [116]: Evaluates how many attributions exceed a certain threshold.

Randomization Metrics

Randomization metrics measure the model's deterioration due to parameter randomization. We consider two such metrics:

- *Model Parameter Randomization* [4]: Quantifies the similarity between original explanations and those from sequential randomization of model layers.
- *Random Logit Test* [148]: Computes the distance between the original explanation and a random class explanation.

Limitation of metrics

As part of the TRAIL project (TRusted AI Labs), I contributed to a study that investigated the various existing metrics for evaluating XAI methods. The research revealed that the choice of hyperparameters associated with these methods has a substantial impact on their ranking according to these metrics [152]. The resulting paper was published in the proceedings of ECML PKDD 2023 (European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases). The study provided quantitative insights into the correlations and redundancies among 14 commonly used XAI evaluation metrics on 9 XAI methods and 3 dummy methods (a randomly generated map, a Sobel filter, and a map produced from a two-dimensional centered Gaussian) used as sanity checks.

The study highlighted that the faithfulness metrics, the most widely used in the field, exhibited notable variability in their correlations and rankings depending on the choice of baseline hyperparameters. For instance, Pixel Flipping and Monotonicity Arya correlations varied significantly between black, white, random, and uniform baselines, underscoring the need for multiple baselines in reliable evaluations. Furthermore, the reliability analysis showed inconsistencies in the rankings assigned by the different faithfulness metrics, as none consistently assigned the lowest rankings to non-explainable dummy methods across datasets and models, highlighting the challenges in achieving robust faithfulness assessments.

Ultimately, these findings underscore a key limitation in XAI evaluation: the lack of a clear, standardized benchmark to determine a method's quality. Given the diversity of metrics and their unique focus on different properties (e.g., robustness vs. complexity), practitioners are left to select metrics based on the aspect they prioritize. These findings suggest that future efforts in XAI evaluation could benefit from exploring more consistent benchmarks, particularly for faithfulness, and considering novel metrics or hyperparameter settings to improve reliability.

4

PolyCAM for CNNs

4.1 Overview of PolyCAM

During preliminary explorations of explainable methods, the lack of precision of the existing methods was a problem to apply to medical images where a more precise feedback was required.

To propose a solution to this limitation, part of the thesis work was devoted to the development of a method called Poly-CAM. The goal of this method is to generate high-resolution class activation maps (CAMs) without relying on gradient backpropagation, thereby reducing the noisiness of the maps. This method achieves high-resolution saliency maps by multiplexing the activation maps from the early layers of a convolutional neural network (CNN) with oversampled class-specific activation maps computed in the later layers.

By leveraging perturbation-based techniques, Poly-CAM can generate high-resolution saliency maps without the erratic behavior typically associated with gradient backpropagation. An illustration of the results with a CAM in comparison to our proposed method on bone x-rays is presented in Figure 4.1.

Poly-CAM was initially introduced in a conference paper presented at ICPR and was further refined and detailed in a subsequent journal paper published in Machine Vision and Applications.

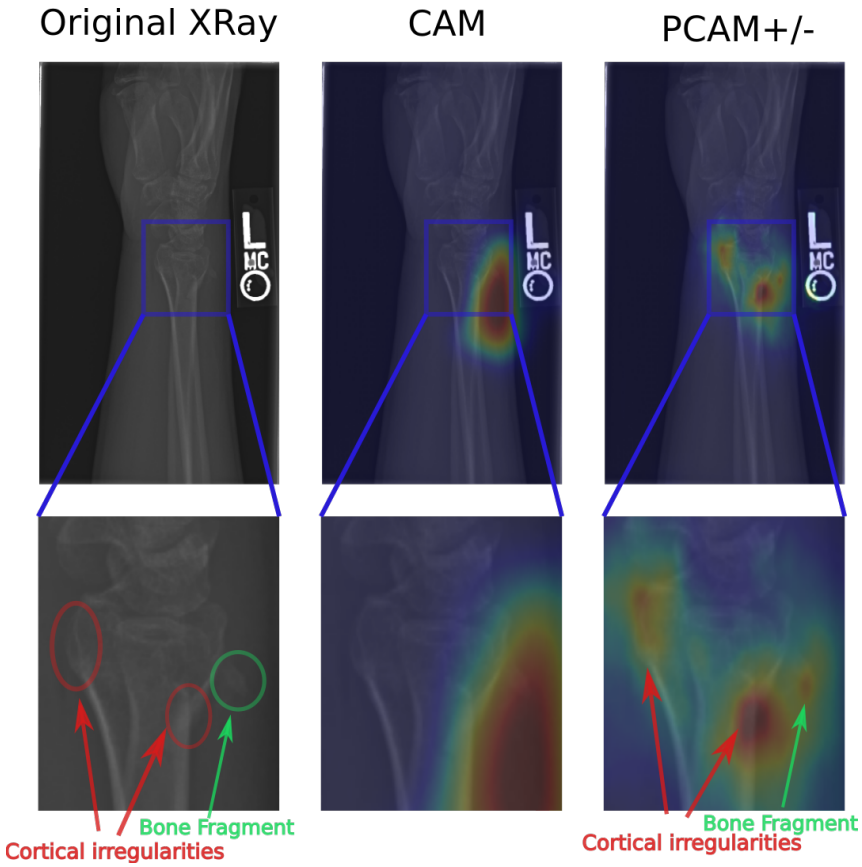


Fig. 4.1 Visual comparison of Class Activation Map [182] and PCAM$^{\pm}$ on a XRay of a bone fracture from MURA dataset [127], for the pathological class label. The bottom row is a zoom on the fracture area. Manual annotations for cortical irregularities and bone fragments (the main signs of the presence of a fracture on this XRay) are shown in red and green ovals. The Class Activation Map is not precise, it seems to include the bone fragment and the right cortical irregularity but due to the low resolution, the highlighted area is very large and go far from the fracture. In comparison, PCAM$^{\pm}$ highlight smaller structures and seems to identify correctly the cortical irregularities and the bone fragment on this image, being probably a greater help for a physician. The model is a ResNet50 [51] initialized on ImageNet, trained on the MURA dataset [127] for 50 epochs with Adam optimizer, an initial learning rate of 6e-5 with a cosine Annealing scheduler without restart, weight decay at 1e-5. Images are resized to 320x320 with random rotation up to 15° during training.

4.2 Methodology and Implementation

This section presents the core contribution of our work. Section 4.2.1 introduces the notations and variables required in the rest of the chapter, while Section 4.2.2 reviews the formal definition of the conventional Class Activation Map method, which serves as a baseline to our work. Section 4.2.3 then introduces our Poly-CAM approach, which proposes to generate a high resolution class activation map by recursively multiplexing the high-resolution activation maps available in the early layers of the network with upsampled versions of the class-specific activation maps computed in the last layers of the network. Eventually, Section 4.2.4 introduces three different methods to associate a weight to each layer activation channel by masking/unveiling the input based on the channel activation. Section 4.2.5 considers a channel switching strategy, meaning that the perturbation is not performed at the input level but directly inside of the network by zeroing specific channels.

4.2.1 Notations

Let $f_\Theta(X)$ denotes the prediction of a CNN with parameters Θ when the image X is provided as input. In the following, for conciseness and because we are interested in analyzing a trained network (parameters Θ are fixed), we omit Θ, and just use $f(X)$ to refer to the CNN prediction associated to X. $f(X)$ is a vector, defined by the output of a softmax. $f_c(X)$ denotes the component of $f(X)$ corresponding to the class c.
A_l denotes the activation tensor of the l^{th} convolutional layer, $1 \leq l \leq L$, while A_l^k refers to the activations of the k-th channel of layer l.
s_l denotes the subsampling factor of layer l compared to the input. It corresponds to the product of stride and pooling factors encountered between the input and layer l.
$\uparrow_{bi} (M, s)$ defines a bilinear upsampling of a matrix $M \in \mathbb{R}^{m \times n}$ by a factor $s \in \mathbb{N}$.
$\downarrow_{av} (M, s)$ denotes a 2D average pooling on any matrix $M \in \mathbb{R}^{m \times n}$ with a stride $s \in \mathbb{N}$.
$u(M)$ linearly maps the value range of the elements in matrix M to the unit interval.
$\oslash$ denotes the element-wise division operator, while $\odot$ denotes the element-wise product operator.

$LNorm(M, s)$ is a local normalization operator. It partitions the matrix $M \in \mathbb{R}^{m \times n}$ in a set of non overlapping blocks of size $s \times s$, with $s \in \mathbb{N}$, and divides each matrix element by the mean value of its corresponding block. Formally, using the above notations,

$$LNorm(M, s) = M \oslash (\uparrow_{bi} (\downarrow_{av} (M, s), s)) . \tag{4.1}$$

ReLU denotes the rectified linear unit operator [31].

4.2.2 Activation Maps in previous work

In [182] CAM_l^c, the Class Activation Map associated to a target class c and a layer l is defined as

$$CAM_l^c = ReLU(\sum_k w_{l,k}(c) A_l^k), \tag{4.2}$$

with A_l^k denoting the k^{th} activation map of the l^{th} convolutional layer, $l \in 1, ..., L$, and $w_{l,k}(c)$ being a scalar weighting factor.

Most of the CAM-based methods [139, 21, 149, 166, 165, 112, 128, 86], adopt this formula. They differ in the way they define the weighting factors, and generally only consider it for the last convolutional layer ($l = L$). Alternatively, Zoom-CAM and Layer-CAM have proposed to combine activation maps from multiple layers, using gradients as dense weighting factors. Our work also combines multiple activation maps, but does it without back-propagated gradients, thereby managing to produce high-resolution saliency maps without inheriting the noise from the gradient. As demonstrated by our results in Section 4.3.2 and Figure 4.5, this has a huge impact on the visual quality of the saliency maps.

4.2.3 Our proposed Poly-CAM

Our method leverages information from multiple those multiple scale layers to produce a high resolution Class Activation Map. Similar to other CAM-based techniques, it builds on the linear combination of activation maps, but combines them through a backward recursive procedure, as depicted in Figure 4.2.
Letting P_l^c denote the class-specific saliency map associated to class c in the l^{th} layer, the recursive process works as follows. In the initial step, the saliency map P_L^c is defined to be equal to the conventional CAM_L^c saliency map, as derived from equation (4.2). Then, at each recursive step, an upsampled version of P_{l+1}^c is tuned (or modulated) by a locally normalized version of the activation map in the l^{th} layer. Mathematically, we have:

$$P_l^c = \begin{cases} CAM_l^c & \text{for } l = L \\ LNorm\left(CAM_l^c, \frac{s_{l+1}}{s_l}\right) \odot \uparrow_{bi} \left(P_{l+1}^c, \frac{s_{l+1}}{s_l}\right) & \text{for } 1 \leq l \leq L-1 \end{cases}, \tag{4.3}$$

with CAM_l^c defined in Equation (4.2), and s_l defining the subsampling factor of layer l compared to the input. The class-specific weights $w_{l,k}(c)$ involved in Equation 4.2 to define CAM_l^c are defined in Section 4.2.4 and 4.2.5, based on perturbations related to the content of the (l, k) channel.

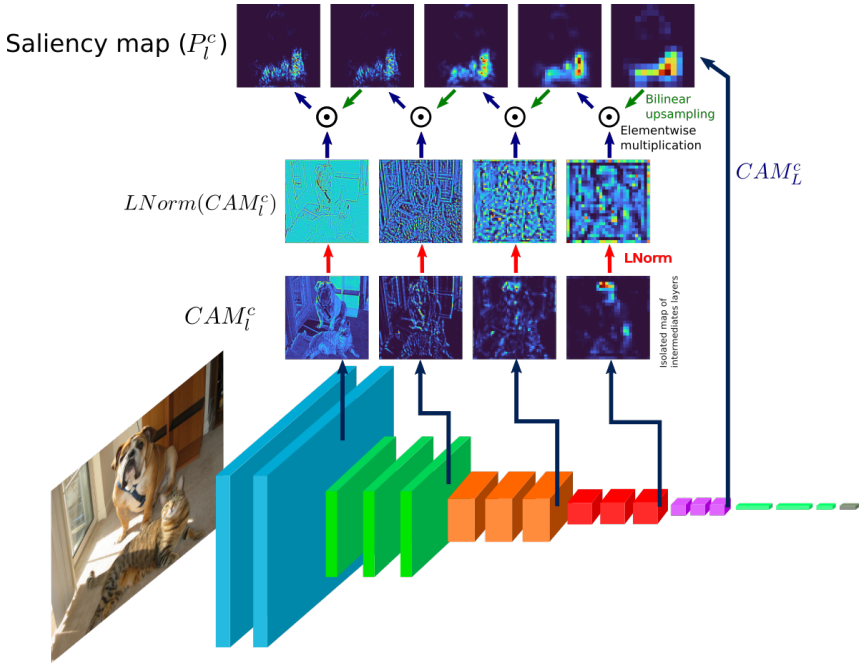


Fig. 4.2 Our Poly-CAM process: the upsampled version of the saliency map in layer l is tuned based on the class activation map of layer $l-1$. Image samples correspond to the 'cat' class, and are computed from VGG16 [146].

Intuitively, Equation (4.3) can be understood based on the following two observations. First, the element-wise multiplication, between the upsampled (and thus smooth) saliency map of layer $l+1$ and the activation map in layer l, aims at restricting the large saliency values in layer $l+1$ to the locations that are activated in layer l. Second, the local normalization ($LNorm$) of the activation map aims at preserving the spatial distribution of saliency across the layers. It ensures that an image block with large saliency in layer l+1 has also a large saliency in layer l, even if the level of activation in this block is small compared to the rest of the image. This is meaningful since the backpropagated saliency should be predominant to assign a saliency level to a spatial region in layer l, while the activation in layer l should simply control the increase in resolution, by tuning the smooth saliency signal inherited from coarser layers based on the local variations of the activation map. Our ablation study in Section 4.3.5 confirms the critical role played by the LNorm operator. This paper also extends the preliminary version of the method by a new algorithm that uses a perturbation at the level of the channels in combination to the aforementioned combination of layers activation

4.2.4 Input perturbation for weight definition

This section presents different alternatives to define the weights $w_{l,k}(c)$, used in Equation (4.2), by using a masking of the input image.

Channel-wise Increase of Confidence Score-CAM [166], SS-CAM [165] and IS-CAM [112] define $w_{l,k}(c)$ based on the so-called Channel-wise Increase of Confidence (CIC), which estimates how the spatial support of the activation map A_l^k contributes to the softmax output f_c. Formally, the channel-wise increase is denoted $\boldsymbol{w}_{l,k}^+(c)$, and is defined as:

$$\boldsymbol{w}_{l,k}^+(c) = f_c\left(X \odot u\left(\uparrow_{bi}\left(A_l^k, s_l\right)\right)\right) - f_c\left(X_b\right), \tag{4.4}$$

with $\odot$ denoting the pixel-wise product, and X_b referring to a baseline image. Previous works have considered baselines that are uniform black, uniform gray, or a blur version of X. In the following, $f_c(X_b)$ is set to zero in all experiments.

Our work proposes two extensions of (4.4), respectively to measure how the softmax output decreases when masking a fraction of the input, and to sum-up the increase and the decrease associated to the unveiling and the masking of the input. Those new weights are defined as follows.

Channel-wise Decrease of Confidence The Channel-wise Decrease of Confidence (CDC) is a dual notion compared to CIC. Instead of measuring the increase of softmax output when the part of the input corresponding to non-zero A_l^k is unveiled and the remaining is masked, CDC measures the decrease of softmax output when the part of the input corresponding to A_k^l is masked. The intuition is that an important part of the input for any class c not only increase the output when shown, but should also decrease it when hidden. Formally,

$$\boldsymbol{w}_{l,k}^-(c) = ReLU\Big(f_c(X) - f_c\left(X \odot \left(1 - u\left(\uparrow_{bi}\left(A_l^k, s_l\right)\right)\right)\right)\Big) \tag{4.5}$$

$ReLU$ is applied to only keep the activation maps that decreases the output when removed.

Channel-wise Variation of Confidence By combining the CIC with the CDC, the Channel-wise Variation of Confidence (CVC) is defined. As,

$$\begin{aligned}\boldsymbol{w}_{l,k}^{\pm}(c) = ReLU\Big(f_c\left(X\right) + f_c\left(X \odot u\left(\uparrow_{bi}\left(A_l^k, s_l\right)\right)\right) \\ - f_c\left(X \odot \left(1 - u\left(\uparrow_{bi}\left(A_l^k, s_l\right)\right)\right)\right)\Big).\end{aligned} \tag{4.6}$$

The Channel-wise Variation of Confidence is influenced by the ability of the activation map to increase the softmax output when inserted, but also by its ability to decrease it when removed.

4.2.5 Channel perturbation for weight definition

Similarly to FD-CAM [86], we propose to define the weight $w_{l,lk}(c)$ by perturbing the l^{th} layer rather than the input. Therefore, channels of a convolutional layer are grouped together based on their similarities. We computed the cos similarity between an activation map A_l^k and every activation maps $A_l^{k'}$ from the same layer l, with $k \neq k'$.

$$cos(A_l^k, A_l^{k'}) = \frac{v_l^k . v_l^{k'}}{\|v_l^k\| . \|v_l^{k'}\|}, \tag{4.7}$$

with v_l^k and $v_l^{k'}$ as the vectors obtained after flattening A_l^k and $A_l^{k'}$. For each channel in a layer, we identify a the predefined percentage κ of channels in the same layer with highest cosine similarities. The subset of channels that are similar to channel k in layer l is denoted $S_{k,l}$ (for the sake of simplicity, we omit the dependency on the input image X in the notation). We let $f_{l,S_{k,l}}(X)$ denote the class-specific output of the model f when all the activation maps in layer l are dropped, i.e. all values are set to zero, except for the maps in $S_{k,l}$, which are kept untouched. Similarly, $f_{c,l,\overline{S}_{k,l}}(X)$ denotes the model f_c when the activations maps of the layer l remain untouched, except for ones in $S_{k,l}$ that are set to zero.

We use an approach similar to the one adopted in Equation 4.6 to define the $w_{l,k}(c)$. This weight is denoted $\boldsymbol{w}_{l,k}^{\emptyset\pm}(c)$, where $\emptyset$ refers to the fact that it is obtained by setting some channels to zero. Formally,

$$\boldsymbol{w}_{l,k}^{\emptyset\pm}(c) = f_c(X) - f_{c,l,\overline{S}_{k,l}}(X) + f_{l,S_{k,l}}(X). \tag{4.8}$$

4.3 Evaluation and Results

Four variants of the Poly-CAM method introduced in Section 4.2.3 and Section 4.2.5 are considered, depending on whether $\mathbf{w}_{l,k}$ is defined to be equal to $\boldsymbol{w}_{l,k}^{+}$ (PCAM$^+$), $\boldsymbol{w}_{l,k}^{-}$ (PCAM$^-$), $\boldsymbol{w}_{l,k}^{\pm}$ (PCAM$^\pm$) or $\boldsymbol{w}_{l,k}^{\emptyset\pm}$ ($\emptyset$PCAM).

We follow the assessment method described in [21] and [120] to evaluate our proposal. This assessment consists of evaluating a defined number of images using insertion and deletion metrics, as described in Section 4.3.1 and Section 4.3.3. Datasets, networks, and baseline methods are presented in Section 4.3.1. Qualitative and visual assessment is presented in Section 4.3.2, while quantitative assessment of the saliency maps is considered in Section 4.3.3. Section 4.3.4 presents the results of the Sanity check and robustness metrics. An LNorm ablation study

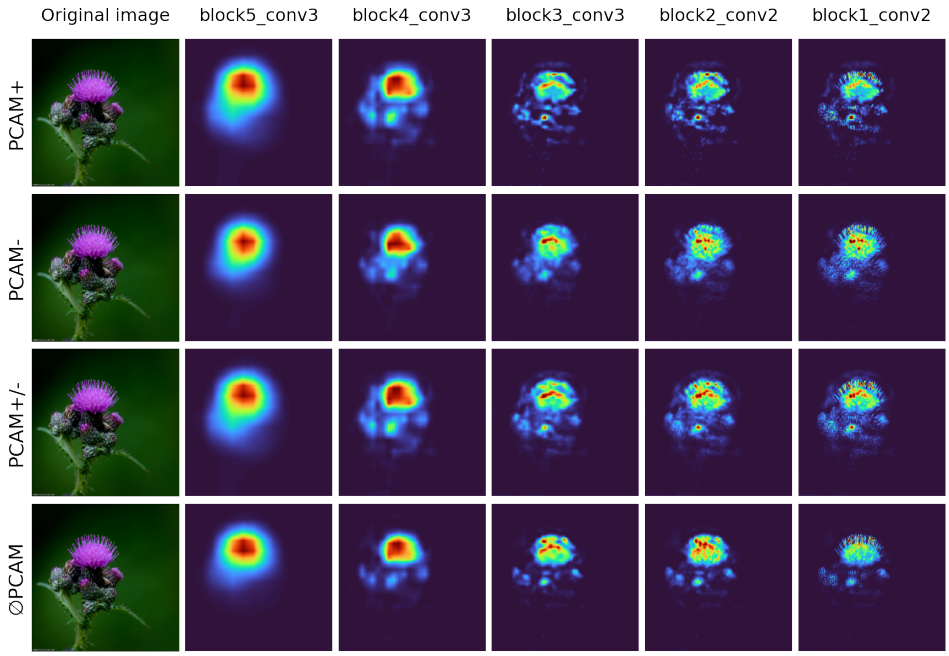


Fig. 4.3 Layer refinement of PolyCAM method for the four proposed variants. The high frequency details appear progressively during the iterative process.

is carried out in the Section 4.3.5. We assessed the speed of execution in Section 4.3.6. Section 4.3.2 explores the usage of the proposed method to an industrial use case.

4.3.1 Experimental set-up and saliency map baselines

For these evaluations, 2000 images were randomly selected from the 2012 ILSVRC validation set [136]. The images are scaled to 224x224x3 pixels and normalized to the same mean and standard deviation as the ImageNet [136] training set (mean vector: [0.485, 0.456, 0.406], standard deviation vector [0.229, 0.224, 0.225]). The models used for faithfulness evaluation are VGG16 [146] and ResNet50 [51], both pretrained from the PyTorch model zoo. The analysis considers, as reference baselines, Grad-CAM [139], Grad-CAM++ [21], Smooth Grad-CAM++ [117], Score-CAM [166], SS-CAM [165], IS-CAM [112], Zoom-CAM [143], Layer-CAM [69], Occlusion [179], Input X Gradient [144], FD-CAM [86], Integrated Gradient [153], SmoothGrad [149] and RISE [120]. The implementations for these methods are the ones from Captum [75] for Integrated Gradient, SmoothGrad and Occlusion, from `https://github.com/eclique/RISE` for RISE, from `https://github.com/X-Shi/Zoom-CAM` for Zoom-CAM, from `https://github.com/crishhh1998/FD-CAM` for FD-CAM and from torchcam [42] for all the other CAM-based methods.

For SS-CAM, IS-CAM, FD-CAM, LayerCAM and ZoomCAM, SmoothGrad and IntegratedGradient, the parameters recommended by the authors or set as default in the reference implementation have been used when available. [1] For FD-CAM, the source code was not compatible with architectures other than VGG at the time of writing without modifications. The experiments were thus limited to VGG16 for this method. For Occlusion, the size of occlusion patch was set to (64, 64) with a stride of (8, 8) as used by [120]. For RISE, 6000 masks were used.

For the Poly-CAM methods (PCAM^{+}, PCAM^{-}, $\text{PCAM}^{\pm}$, $\emptyset$PCAM), the layers corresponding to a change in resolution were considered to recursively compute the saliency map as depicted in Figure 4.2. It corresponds to [block1_conv2, block2_conv2, block3_conv3, block4_conv3, block5_conv3] for VGG16, and [conv1_1, conv2_3, conv3_4, conv4_6, conv5_3] for ResNet50. For $\emptyset$PCAM, κ was set to 0.05, following previous works [86].

4.3.2 Visual qualitative assessment

This section assesses our method visually. Saliency maps were generated for all the baseline methods (see Section 4.3.1) on the 2000 selected images using VGG16 model. For the Poly-CAM methods, saliency maps were also generated for each

[1] It means 35 input perturbations (Gaussian noise with a $\sigma = 2$) for SS-CAM, 50 input perturbations (with a $\sigma = 1$ Gaussian noise) for SmoothGrad, 10 interpolation steps for IS-CAM, threshold at 0.95 for FD-CAM and 50 for IntegratedGradient. For Layer-CAM, the layers corresponding to a change in resolution were used, and recommended scaling has been applied to the first two layers. For Zoom-CAM, all the layers/blocks were fused for VGG16 and ResNet50.

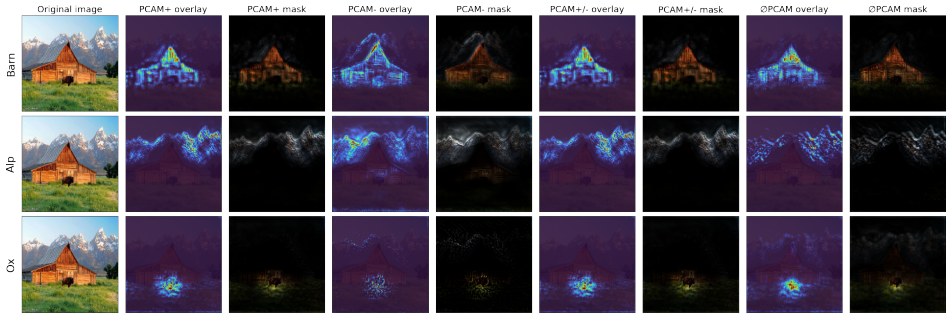


Fig. 4.4 Class specificity for Poly-CAM. The different classes as correctly determined by the four variants. We can note that PCAM$^-$ is less specific than the other methods (part of the mountain is attributed to the barn), while PCAM$^+$ is the most specific. In general, the different PolyCAM methods perform similarly.

target layer. An interactive interface is provided with the source code as a jupyter notebook, in addition to the source code, to allow an easy visualization of any of the saliency maps generated in our experiments. Section 4.3.2 compares the three Poly-CAM variants, as a function of the layer index and targeted class. A comparison with previous works is shown in Section 4.3.2.

PCAM variants

PCAM produces saliency maps at various resolutions. Figure 4.2 and Figure 4.3 show how Poly-CAM progressively refines the last layer saliency map through a backward recursive strategy. We observe that the structures are coarse at block5_conv3, to gain in accuracy when accounting for earlier network layers, doubling the resolution at each step. The elements highlighted by the three variants are similar on the majority of images. However, variations appear on some images, PCAM$^-$ highlighting more frequently contextual elements compared to PCAM$^+$ (and PCAM$^\pm$ sitting between the two). baseline image (without the core object being classified) does not aid in accurate classification, while removing these features from the original image impedes classification. ∅PCAM produces similar results but with sharper and more refined highlights. For example, imagine an image of a cow with grass in the background. Replacing the grass could lower the class probability of "cow" and increase the likelihood of other classifications such as "Dalmatian" and "carpet". Conversely, adding only the grass background while masking the cow would be too generic to significantly increase the probability of the "cow" class.

All Poly-CAM variants are class specific as displayed in Figure 4.4, where the saliency maps associated to the the Barn, Alps and the Ox classes are clearly distinct, with a level of accuracy close to segmentation. It is worth noting that PCAM$^+$ is more specific in highlighting the part of the image related to the class

of interest. This is in line with the above observation that PCAM$^-$, and a bit less PCAM$^\pm$, are stronger in highlighting contextual information.

Comparison with previous works

Saliency maps have been produced for all baseline methods listed in Section 4.3.1. A sample of this comparison is presented in Figure 4.5. We can see that Poly-CAM methods accurately identify quite relevant elements in the image like a spider net or the pipes of an organ. CAM and perturbation methods cannot achieve this level of precision, gradient base methods highlight elements of the image that are not related to the class, while Zoom-CAM and Layer-CAM increase the resolution but are more noisy, halfway between gradient and more classical CAM-based methods. The spotted salamander is highlighted by all methods but the Poly-CAM methods are the only ones to identify the spots. The oranges are identified by Poly-CAM methods while the smile sketched on them is correctly excluded. This is in contrast with other methods that are either too low resolution, or do not exclude the smile, while SmoothGrad seems to give more importance to the smile than to the texture of the orange.

Industrial defect localization

To illustrate the importance of fine details identification in the industry, we also performed an experiment involving defect detection on stone tiles. The used dataset is the Stone Tiles dataset[2], which consists of 605 labeled images of stone tiles. The labels are either "Good", "Broken", "Damaged" or "Glued". We split the dataset randomly into a validation set (10% of the images) and a training set (90% of the images). We fine tuned a VGG16 model on this dataset, pretrained on ImageNet from the PyTorch model zoo. The model was trained for 50 epochs with a fixed learning rate of 0.001, without early stopping, achieving an accuracy of 89.7%. Score-CAM and $PCAM^\pm$ were performed on the validation set images for the classes predicted by the models. We compared the saliency maps produced by ScoreCAM and the proposed method for images that are classified as other than "Good", either correctly or wrongly, to assess if the explanations given by the two methods allow to better grasp the reasons of the classification.

Figure 4.6 show explanation for correct classifications of stone tiles. Both ScoreCAM and PolyCAM allow to understand the pixels related to a 'Broken' stone tile in the image but PolyCAM is more precise. For 'Damaged' and 'Glued' labels, the area highlighted by Score-CAM are generally very wide since the defects can be distributed on the whole image. ScoreCAM maps thus bring little information about what the model learned to detect theses labels. On the other hand, Poly-CAM show more fine details for those classes and allow to identify that the model use the scratches on the surface of the 'Damaged' tile and the small traces of glue on the 'Glued' tile to make a classification. Figure 4.7 shows the two

[2]Downloaded from the Euresys website: `https://www.euresys.com`

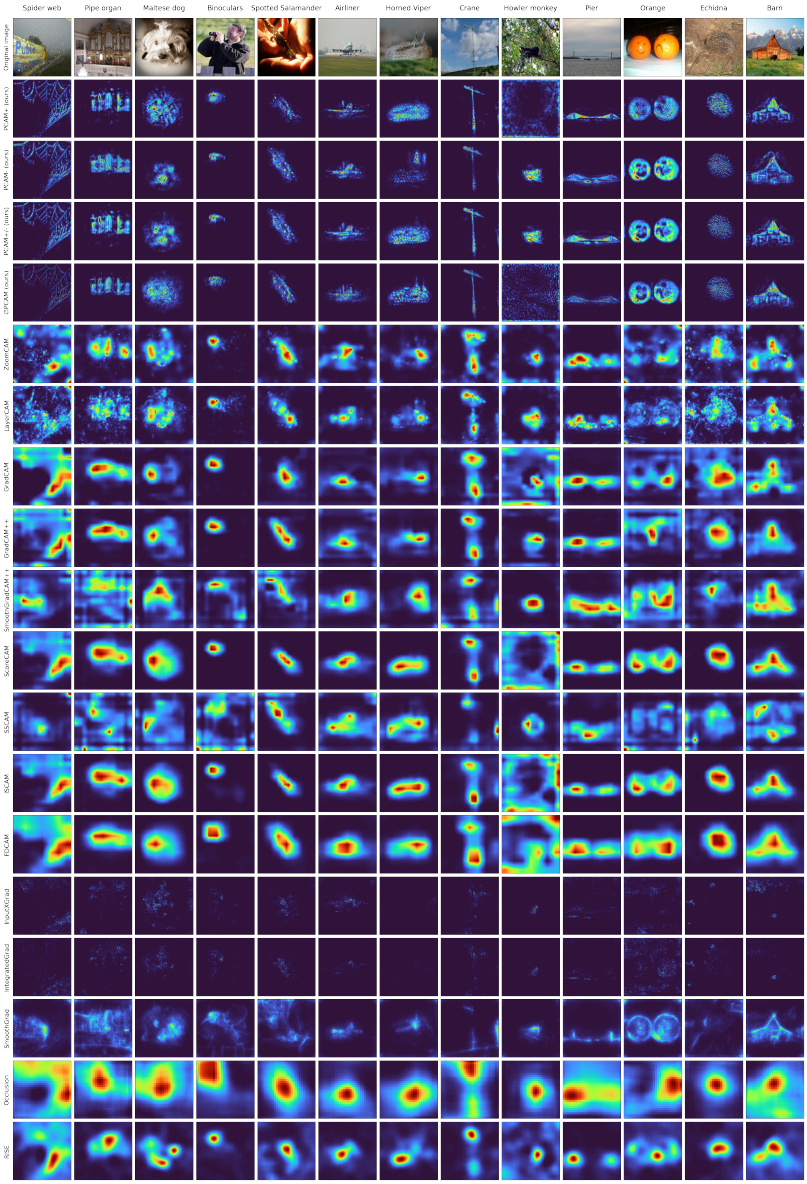


Fig. 4.5 Visual comparison of methods. The compared methods are the PolyCAM variants proposed in this paper, Zoom-CAM [143], Layer-CAM [69], Grad-CAM [139], Grad-CAM++ [21], Smooth Grad-CAM++ [117], Score-CAM [166], SS-CAM [165], IS-CAM [112], Input X Gradient [144], IntegratedGradient [153], SmoothGrad [149], Occlusion [179], RISE [120].

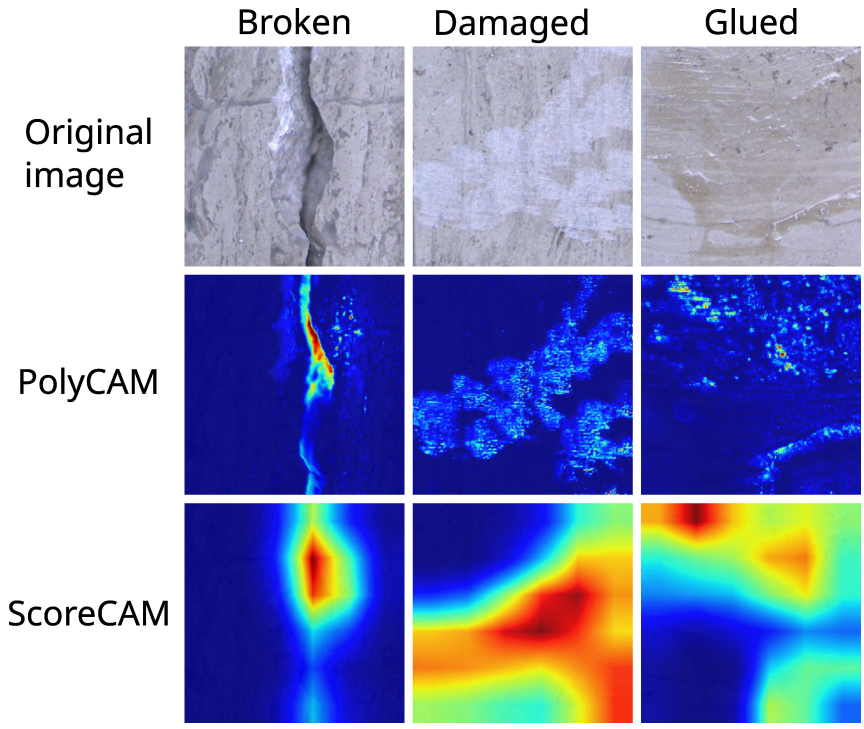


Fig. 4.6 Correct stone tile classifications explanations. The figure show representative Poly-CAM and Score-CAM explanations of correctly classified images of stone tiles using VGG16. Poly-CAM show a more precise identification of the causal elements of the classification for the three classes, in particular for Damaged and Glued tiles where the structures of the scratches and the glue are more clearly identified.

samples that are erroneously classified as defectives. For both images, Poly-CAM help to identify more precisely than Score-CAM elements in the images that can be interpreted by the model as 'Broken' or 'Damaged'

4.3.3 Faithfulness Quantitative Assessment

In evaluating saliency maps as introduced in Chapter 3, there remains no consensus on optimal metrics for assessing their relevance [122]. Saliency maps, which highlight critical regions tied to model predictions, are often evaluated by their capacity to localize relevant semantic objects [139]. However, as discussed in [120], segmentation masks alone may not fully capture the discriminative features supporting a class label. To address this, we utilize metrics that examine changes in model predictions as pixels are added or removed based on their saliency scores.

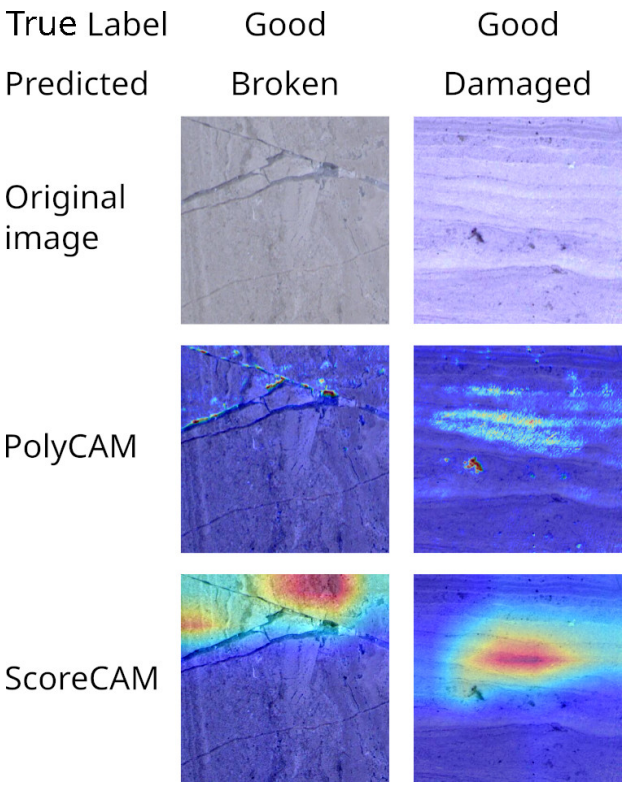


Fig. 4.7 False positive images of stone tiles. This figure show explanations of images labeled as good but falsely labeled as broken or damaged. Poly-CAM supports, i.e. explains, the wrong decision. We can identify using Poly-CAM that the dark veins are interpreted as fracture lines for the left image, while the lighter areas are shown as scratches for the right images. In comparison, Score-CAM also highlight roughly the veins on the left image, but the explanation on the right image is much less clear.

Specifically, we evaluate how the model's softmax output varies when salient pixels are either inserted or deleted from a baseline image. The insertion metric assesses the increase in the softmax score as salient pixels are progressively added to a blurred baseline image, indicating how quickly the model's confidence grows. In contrast, the deletion metric measures the decrease in the softmax score as these pixels are sequentially removed, providing insight into how critical the identified features are for maintaining the model's prediction.

A combined score derived from both metrics, calculated by subtracting the deletion score from the insertion score, offers a balanced measure of the saliency map's effectiveness.

While these metrics provide valuable quantifiable insights, they have notable

limitations. Specifically, modifying pixel composition can yield out-of-distribution images, potentially misleading the model. To mitigate this risk, we use a blurred baseline (Gaussian kernel, 11x11, sigma=5), which aligns with common practices in recent studies. Furthermore, although high scores in these metrics suggest a strong influence of salient pixels on predictions, they do not guarantee that all critical features have been identified. Therefore, we complement these metrics with qualitative visual assessments, which remain the gold standard for evaluating saliency map fidelity.

Experiments were conducted with 224 steps, adjusting 224 pixels per step, to provide fine-grained fidelity measurements for each assessment.

Quantitative assessment

Table 4.1 compares the faithfulness metrics for all methods. Among the Poly-CAM variants, PCAM$^{\pm}$ gives the best results compared to PCAM^{+}, PCAM^{-} and $\varnothing$PCAM for all metrics on VGG16. On ResNet50, The results are very close for all the variants, PCAM$^{\pm}$ gives a insertion metric similar to PCAM^{+} and slightly superior to PCAM^{-} and $\varnothing$PCAM, while PCAM^{-} and $\varnothing$PCAM give a better deletion metric compared to PCAM^{+} and PCAM$^{\pm}$. For insertion-deletion on ResNet50, PCAM$^{\pm}$, $\varnothing$PCAM and PCAM^{-} are on par.
Interestingly, the insertion metrics of PCAM$^{\pm}$ is systematically better than all other CAM-based approaches. Compared to the non-CAM methods, the PCAM$^{\pm}$ method gives similar or better insertion results than perturbation or gradient methods, respectively.
In terms of deletion, PCAM$^{\pm}$ tends to perform better than most other CAM-based methods, but appears to be weaker than gradient-based methods. InputXGrad and IntegratedGradient achieve at the same time very poor results on the insertion metric and thus have a poor insertion-deletion. This is not surprising since gradient-based methods give lots of importance to the parts of the input that largely impact the loss and thus the output. As a consequence, the deletion metric is (trivially) good for those methods since this metric measures the decrease of output when important parts are removed from the input. The poor insertion metric however reveal that the parts that are considered as being important by gradient methods are not sufficient to explain the network prediction. This observation reveals the limits of the metrics when applied to dissimilar kinds of techniques.

4.3.4 Sanity check and robustness

We followed the method in [5] to implement a sanity check of our method. It consists in progressive, iterative, layer-wise randomization of the network parameters while regenerating an explanation after the randomization of each additional layer to evaluate the influence of the model's parameters on the explanation. So, the PCAM saliency maps have been visualized at each step of a cascading ran-

domization of a VGG16 network, from last to first layer. The purpose of this sanity check is to verify that the PCAM methods do not work as edge detectors, and effectively derives class-specific saliency maps. All PCAM variants successfully passed the test, as shown in Figure 4.8.

To evaluate the robustness of our explanation method, a sensitivity analysis has also been run, following the methodology introduced in [47] and [177]. The approach consists in assessing the extent of change in the saliency map produced when small perturbations are introduced into the input image. Essentially, we assess the maximum difference in the explanation that occurs due to these tiny input changes. A method with a higher sensitivity is more prone to adversarial attacks. Results are presented in Table 4.2. They reveal that PCAM has a small explanation sensitivity , similar to the ones obtained by other CAM-based methods, and one or two orders of magnitude below the sensitivities obtained by gradient-based and perturbation methods.

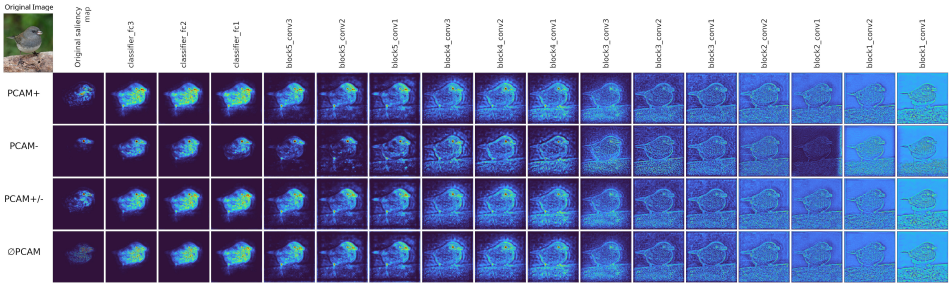


Fig. 4.8 Cascading randomization of VGG16. Sanity check on Poly-CAM methods [5]. Progression from left to right corresponds to a progressive, layer-wise randomization of the network parameters. It show how the saliency map changes with increasing up to complete randomization of the VGG16 model, starting by the last layer up to the first layer. The methods are sensible to model randomization, which mean they pass this sanity check. It is interesting to note that the class specificity is lost rapidly after randomising the first classifier layer, then more and more features are lost while randomization progress up to the first layer of the network.

4.3.5 Ablation study on LNorm

The importance of including the LNorm operator in Equation 4.3 is challenged in this section. We produced saliency maps using both the complete method and a variant where LNorm has been ablated. Formally, the saliency map of the LNorm ablated method becomes

$$\overline{P}_l^c = \begin{cases} CAM_l^c & \text{for } l = L \\ \left(CAM_l^c, \frac{s_{l+1}}{s_l}\right) \odot \uparrow_{bi} \left(\overline{P}_{l+1}^c, \frac{s_{l+1}}{s_l}\right) & \text{for } 1 \leq l \leq L-1 \end{cases}, \tag{4.9}$$

Representative examples are presented in Figure 4.9 to compare the conventional and ablated PCAM. We clearly observe that the ablated method tends to ignore some of the class-relevant features, and to focus on a limited set of highly contrasted features (such as eye, mouth, or beak).

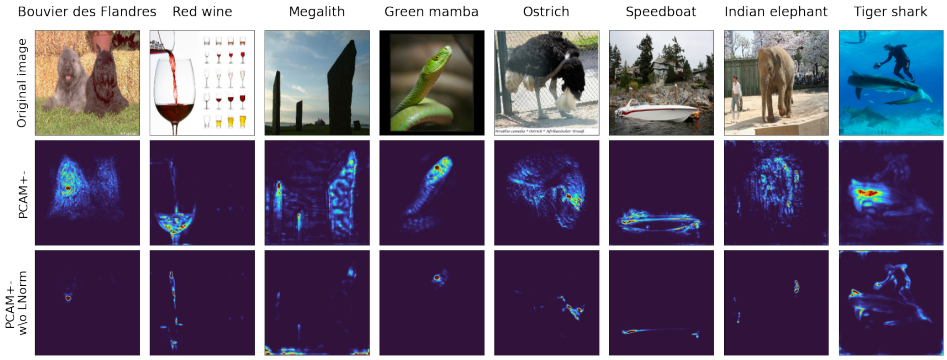


Fig. 4.9 Visual comparison of PCAM$^{\pm}$ with and without LNorm. Without LNorm the visualization tends to concentrate on very focal elements of the images like eyes or mouth. Sometime some elements of the image that are not in object of the target class become also highlighted, like an object behind the elephant, or the diver next to the shark.

4.3.6 Speed of execution

Using the same experimental setup described in the previous sections, we measured the average average execution time per image over the first 100 images of the ILSVRC2012 validation set for both the perturbation-based methods and the subset of CAM-based methods relying on perturbations. Measurements were repeated for batch sizes of 32, 64, and 128 images on an 8Gb NVIDIA RTX3070M and reported in Table 4.3.

The results show that the different PolyCAM variants fall between RISE and ScoreCAM in terms of computation time. FDCAM is the fastest method using perturbations, this is understandable since the perturbations only need to be performed for the head of the network and don't require to recompute the whole backbone, but is also limited to the resolution of the last convolutional layer. It is interesting to note that the ∅PCAM method, that compute scores similarly to FDCAM, is the fastest of or four variant, with a similar execution speed to ScoreCAM. ∅PCAM seems to be the most appropriate compromise in terms of high resolution saliency maps quality and execution time.

4.4 Discussion

Our Poly-CAM method produces high resolution saliency maps without relying on gradient backpropagation. Two variants of our Poly-CAM framework are investigated. In the first variant, the values weighting the activation maps are obtained by masking or unveiling image pixels, or both. In the second variant, an approach using perturbations inside of the network by switching off and on convolution channels is implemented. Our experiments reveal that a strategy combining masking and unveiling, either in the pixel space or at the level of channels, provides the more versatile solution. It achieves state of the art performances in term of faithfulness insertion-deletion metrics and outperforms current available methods in term of precision of visualization. As a main original contribution, our method allows for the high-resolution visualization of image regions that contribute to the network prediction. The channel switching strategy having the advantage of being quicker to compute. Despite our work being a valuable step towards a more explainable AI, there is still plenty of room for improvement in this domain. One of the questions raised by this work is related to the way the importance of a pixel should be quantified. Indeed, the importance of a group of pixels appears to be different when this group is removed or when it is inserted (for example the importance of contextual information is more important when removing it than when inserting it), which can not be properly reflected by a single saliency map.

A practical usage of PolyCAM on bone radiographs is presented in Chapter 6 to explore spurious correlations and bias in the data.

Table 4.1 Faithfulness metrics for all methods: CAM-based, gradient and perturbation methods.

Methods	VGG16			ResNet50		
	Ins	Del	Ins-Del	Ins	Del	Ins-Del
PCAM^{+} (ours)	0.58	0.17	0.41	**0.67**	0.29	0.38
PCAM^{-} (ours)	0.60	0.16	0.45	0.66	0.27	**0.39**
PCAM$^{\pm}$ (ours)	**0.61**	0.15	**0.46**	**0.67**	0.28	**0.39**
∅PCAM (ours)	0.60	0.17	0.43	0.66	0.27	**0.39**
GradCAM	0.58	0.18	0.40	0.65	0.31	0.35
GradCAM++	0.57	0.19	0.38	0.65	0.31	0.34
SmoothGradCAM++	0.54	0.21	0.33	0.63	0.32	0.30
ScoreCAM	0.59	0.19	0.40	0.65	0.31	0.34
SSCAM	0.50	0.23	0.27	0.59	0.36	0.24
ISCAM	0.59	0.19	0.40	0.65	0.32	0.33
FDCAM	0.60	0.20	0.40	-	-	-
ZoomCAM	0.60	0.14	**0.46**	0.66	0.29	0.37
LayerCAM	0.58	0.14	0.44	0.65	0.30	0.35
IntegratedGradient	0.41	**0.10**	0.31	0.52	**0.16**	0.36
InputXGrad	0.37	0.12	0.26	0.47	0.18	0.28
SmoothGrad	0.54	0.20	0.34	0.62	0.29	0.33
RISE	**0.62**	0.18	0.44	**0.67**	0.28	**0.39**
Occlusion	**0.62**	0.23	0.39	0.66	0.33	0.33

Insertion (higher is better), deletion (lower is better) and Insertion-Deletion (higher is better) with VGG16 and ResNet50 on the 2012 ILSVRC validation set. **Boldface** and underline indicate the best result and the best result amongst CAM-based methods respectively. Comparison of our Poly-CAM methods with gradient methods: Input X Gradient [144], Integrated Gradient [153], SmoothGrad [149], perturbation methods: Occlusion [179] and RISE [120], and CAM methods: Grad-CAM [139], Grad-CAM++ [21], Score-CAM [166], SS-CAM [165], IS-CAM [112], Smooth Grad-CAM++ [117], FD-CAM [86], Zoom-CAM [143] and Layer-CAM [69]. FD-CAM source code was only compatible with VGG16 at the time of writing.

Table 4.2 Sensitivity max table.

Method	Sensitivity max	
	VGG16	ResNet50
IntegratedGradient	0.3576	0.5299
InputXGrad	0.6132	0.7225
SmoothGrad	5.6824	7.7777
RISE	0.7864	0.7841
Occlusion	2.5176	3.4378
GradCAM	0.0625	0.0212
GradCAM++	0.0525	0.0199
SmoothGradCAM++	0.5451	0.1594
ScoreCAM	0.0466	0.0193
ISCAM	0.0433	0.0334
FDCAM	0.0433	-
ZoomCAM	0.0987	0.0485
LayerCAM	0.0937	0.0590
PCAM^{+} (Ours)	0.0650	0.0262
PCAM^{-} (Ours)	0.0837	0.0578
$\text{PCAM}^{\pm}$ (Ours)	0.0659	0.0659
$\varnothing$PCAM (ours)	0.0783	0.0419

Sensitivity max metric measures maximum sensitivity of an explanation using Monte Carlo sampling-base approximation [177]. A method with a higher sensitivity is more prone to adversarial attacks. Captum implementation was used with 10 perturbations per input and a epsilon radius of a L-Infinity ball set to 0.02 for sampling (defaults parameters from the implementation) [75]. The compared methods are the three Poly-CAM variants proposed in this paper (PCAM^{+}, PCAM^{-}, $\text{PCAM}^{\pm}$), Zoom-CAM [143], Layer-CAM [69], Grad-CAM [139], Grad-CAM++ [21], Smooth Grad-CAM++ [117], Score-CAM [166], SS-CAM [165], IS-CAM [112], FD-CAM [86], Input X Gradient [144], IntegratedGradient [153], SmoothGrad [149], Occlusion [179], RISE [120].

Table 4.3 Execution time per image.

Batch size	128	64	32
PCAM^{+} (ours)	4.32	4.38	4.67
PCAM^{-} (ours)	4.38	4.68	4.71
PCAM$^{\pm}$ (ours)	8.48	8.92	9.60
$\emptyset$PCAM (ours)	1.51	1.61	2.01
ScoreCAM	1.48	1.52	1.64
ISCAM	12.85	13.29	14.18
SSCAM	47.13	48.01	52.12
FDCAM	0.17	0.17	0.17
RISE	OOM*	OOM*	18.50
Occlusion	2.62	2.62	2.62

Execution speed to compute the saliency maps for a VGG16, expressed in second per image. Mean over 100 computations.
*OOM: Out of Memory

5

Transformer Input Sampling (TIS)

5.1 Overview of TIS

Lately, the rise of the transformer architecture [161] in multiple modalities provides a new challenge in terms of explainability. Especially in the field of computer vision, where the convolutional neural network (CNN) has been the dominant architecture type since AlexNet in 2012 [76], with many explainability methods targeting these CNN architectures [139, 166, 40]. The switch from CNN to transformer in the other area of this thesis leads to the necessity to explore XAI techniques adapted to transformer architecture.

The proposed Transformer input sampling (TIS) method is a perturbation-based methods, with a main contribution of defining perturbations as a sampling of the tokens before the first transformer layer, but after the linear projection and position encoding of the patches inside a Vision Transformer (ViT) [36]. This definition avoids the generation of outlier inputs, thereby limiting the risk of misleading the interpretation of the transformer predictions. Another advantage is that the reduction of the number of tokens at the transformer input also increases the inference speed for each perturbation, enabling more samples to be evaluated with the same computing power. This also renders the method more versatile in comparison to perturbations at the pixel level as proposed by ViT-CX [172]. Since the method sample tokens that are the building blocks of transformers in any modality, the method can potentially be extended to other modalities and to multimodal transformers.

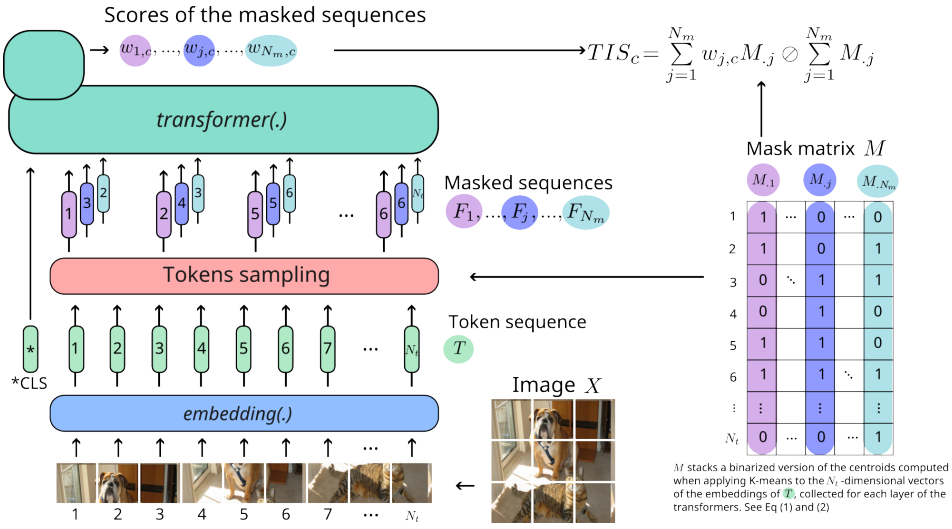


Fig. 5.1 Illustration of the Transformer Input Sampling (TIS) process. The columns $M.j$ of the matrix M are the masks used to produce each sampled sequences F_j. The scores $w_{j,c}$ are the scores for each sequences F_j for a target class c.

The method was published in a paper in the proceedings of the IEEE/CVF International Conference on Computer Vision Workshop.

5.2 Methodology and Implementation

Section 5.2.1 introduces useful notations. Section 5.2.2 gives a general overview of our proposed method. Section 5.2.3 details the generation of masks, and its corresponding token sampling process. Section 5.2.4 explains the mask scoring process, leveraging the variable input length property of transformers, and the saliency map computation as a score-based weighted sum of masks.

5.2.1 Notations

Let $f(X)$ denote a vision transformer model [36, 157] applied to an image X. This model is composed of an embedding computation module (patch and positional embedding) denoted embedding(X), whose result is a matrix $T \in \mathbb{R}^{N_t \times D}$ composed of N_t tokens of dimension D, and a transformer encoder [161] with a task-focused head denoted transformer(X), such as $f(X) = \text{transformer}(\text{embedding}(X))$. In the following, the result of $f(X)$ is a vector of dimension C, defined as the output of a softmax function, and $f_c(X)$ corresponds to the score given by the model to a particular class c for the image X. Let $A_{i.}$ be the i-th row of a given matrix A, and $A_{.j}$ the j-th column of a given matrix A. Consider $\oslash$ as the element-wise

division operator, and $\odot$ as the element-wise product operator. Let $\text{topk}(A, n)$ be the set of n largest elements in a given set A.

5.2.2 General Overview

The proposed method computes class-specific saliency maps. It relies on the output score associated with the class of interest when inputting different subsets of the input tokens in the transformer part of the model. A schematic illustration of the process is depicted in Figure 5.1. The tokens are sampled before the transformer encoder. This is similar in principle to the Masked Autoencoders [50], with masks being generated based on the activations of the transformer model. Previous works have shown that, even if the multi-head attention modules of a vision transformer are position invariant, the tokens keep the localization information from the beginning up to the end of the model thanks to the multiple residual connections [126]. This location-preserving property in the embedding space enables the use of the embedding to guide the masking process, similarly to what is done by Score-CAM for a convolutional neural network [166]. It is worth noting that unlike perturbations methods in the input space that modify the pixels values such as RISE [120], Score-CAM [166] or ViT-CX [172], our method leverages the ability of the transformer to accept a sequence of tokens with variable length to completely remove a portion of the tokens (i.e., the patches) in a way that the model can only perform computations on the remaining tokens. Since this is done just after the positional embedding and before any self-attention, the non-sampled tokens do not have any influence on the output. This is in contrast with the generation of outlier images that can be produced when corrupting the input.

5.2.3 Mask Generation and Token Sampling

The first step when generating a mask to control the sampling of a token sequence $T \in \mathbb{R}^{N_t \times D}$, composed of N_t tokens (excluding the CLS classification token) with dimension D, is to concatenate the activation/embeddings from every layer in the transformer into a matrix $A \in \mathbb{R}^{N_t \times L.D}$ with L being the number of layers of encoders in the transformer. Since the computational requirements increase with the forward passes computed for each mask and many maps are redundant, we use a clustering process to reduce the number of masks, similarly to ViT-CX [172]. A K-Means clustering is used on the columns of A to produce a smaller matrix $K \in \mathbb{R}^{N_t \times N_m}$ with N_m being the number of masks. The number of centroids of K-Means N_m is a parameter of our method. The choice of N_m is evaluated in the Supplementary material and set to 1024 in the remaining of the paper.

$$K = KMeans(A, N_m) \tag{5.1}$$

Unlike previous works based on masks generated from the activation maps with continuous values [172, 166, 40], we propose to binarize the masks so that each value in the matrix means whether we will keep the corresponding token or not when computing the class score. We thus produce a binary matrix $M \in \{0,1\}^{N_t \times N_m}$.

Formally,

$$M_{ij} = \begin{cases} 1 & \text{if } K_{ij} \in \text{topk}(K_{.j}, N_k) \\ 0 & \text{otherwise} \end{cases}, \tag{5.2}$$

with N_k being the number of tokens to sample.

We obtain N_m sequences of sampled tokens. The j^{th} sequence $F_j \in \mathbb{R}^{N_k \times D}$, is associated to the mask $M_{.j}$ in M, and is defined as follows,

$$F_j = \{T_i | M_{ij} = 1\} \tag{5.3}$$

5.2.4 Mask Scoring and Saliency Map

The class-specific relevance score $w_{j,c}$ of each mask M.j is obtained by passing its corresponding set of tokens F_j in the transformer and retrieving the model output for the target class c. Formally,

$$w_{j,c} = transformer_c(F_j), \quad \text{for } 1 \leq j \leq N_m \tag{5.4}$$

Since each token is related to a patch in the input image, the more a particular token is relevant for a given model output, the more the corresponding patch is also relevant. Therefore, it becomes relevant to compute a saliency map as the sum of the masks weighted by the score obtained by the corresponding sampled tokens. This sum can be improved by dividing by the sum of the masks to account for possible token frequency bias, similar to the pixel coverage bias addressed in ViT-CX [172]. Hence,

$$TIS_c = \sum_{j=1}^{N_m} w_{j,c} M_{.j} \oslash \sum_{j=1}^{N_m} M_{.j} \tag{5.5}$$

In the following, the resulting saliency maps are bilinearly upsampled to the resolution of the input image.

5.3 Experimental Setup

This section describes the experimental setup used to benchmark the proposed method in comparison to previous works. For our proposed TIS method, we em-

ploy a token masking ratio of 0.5, translating to 98 tokens over 196 (formally, $n_k = 98$ in Equation 5.2.3) and 1024 masks ($N_m = 1024$ in Equation 5.2.3). This set of parameters is discussed in supplementary material. Good results are obtained with values ranging from 128 to 1024 masks, with little gain beyond 1024. The methods used for comparison are ViT-CX [172], Transition Attention Maps (TAM) [178], the two methods from Chefer [22, 23], Attention Rollout [2], the token (BT-T) and head (BT-H) methods from Bidirectional Transformers [24], RISE [120], Integrated Gradient [153] and SmoothGrad [149]. The parameters used are 20 steps for TAM, 4000 masks for RISE, 50 interpolations for Integrated Gradient, and 50 perturbations for SmoothGrad. We used the released codes from the authors for ViT-CX[1], RISE[2], Chefer[3], TAM[4] and Bidirectional Transformers[5]. We applied Captum [75] implementations for SmoothGrad and Integrated Gradient.

5.3.1 Transformer Models

The two models used in the experiments are ViT and DeiT, typically used to solve computer vision tasks such as image classification. The Vision Transformer (ViT) [36] is an encoder-only transformer architecture. In particular, each image is divided into N non-overlapping patches which are then projected into the embedding space as a sequence of tokens that serve as input to the transformer backbone. In addition, a learned classification token (CLS token) is prepended to this sequence. After the final encoder layer, the representation of the CLS token depicts a global embedding of the image and is classically used as input to a head trained for downstream tasks such as classification. DeiT [157] derives from ViT, but in addition to the CLS token it also has a distillation token that is combined with a second classification head dedicated to learning by distillation from the predictions of a teacher network. In the following experiments, the ViT model denotes the ViT-Base variant [36], and the DeiT denotes the DeiT-Base variant [157]. We utilized the implementations from the timm library [170] using ImageNet 21k pretraining with ImageNet 1k finetuning weights for both models.

5.3.2 Metrics

In the domain of explainable AI (XAI), explainability metrics provide a way to evaluate the performance of explanation methods, minimizing the subjectivity of human judgment as discussed in Chapter 3.

Given the absence of ground-truth explanations, these metrics evaluate XAI methods across diverse properties, allowing for a more objective comparison. To cover the range of evaluated properties in state-of-the-art explainability metrics,

[1] https://github.com/vaynexie/CausalX-ViT
[2] https://github.com/eclique/RISE
[3] https://github.com/hila-chefer/Transformer-Explainability
[4] https://github.com/XianrenYty/Transition_Attention_Maps
[5] https://github.com/jiaminchen-1031/transformerinterp

we report results based on four selected metrics: Insertion and Deletion[120] (a faithfulness metric), Pointing Game[180] (a localization metric), Max-Sensitivity[177] (a robustness metric), and Sparseness [20] (a complexity metric). These metrics were chosen based on two criteria: their frequent use in evaluating state-of-the-art XAI methods (e.g., Insertion, Deletion, Pointing Game), and the range of properties they represent (e.g., Sparseness for conciseness, Max-Sensitivity for robustness).

For the Insertion and Deletion metrics, 224 steps were used in the iterative computation and each metric was computed using four baselines (blur, random, black, and mean). Regarding the Pointing Game metric, we excluded images where the bounding box covered more than 50% of the image, thereby following the recommendations in [166, 172]. This results in 2892 images excluded and 2108 images included for this metric. For Max sensitivity, we used Captum's implementation with a number of perturbed samples set to 10 and a perturbation radius set to 0.02. For Sparseness, in the case of negative values, we shifted the minimum value to zero before applying the metric[6]. Since this metric serves as an additional indicator (concise explanations) rather than a ranking, the corresponding results are presented in the Supplementary material.

5.3.3 Assessment Protocol

Given the evaluation metrics, the assessment adopts the protocol used in previous works [120, 22, 23, 172, 2] on explainable AI applied to convolutional neural networks and vision transformers. It consists in evaluating the saliency maps generated with the different methods on a random subset of the ImageNet validation set [136]. We set the size of this subset to 5000 images [172, 24].

5.4 Experimental Results

This section analyzes the results obtained by our method and compares them to previous works from a qualitative and quantitative point of view.

5.4.1 Qualitative Assessment

General Comparison

In the field of explainable AI, metrics primarily represent approximations of isolated properties, unable to fully quantify the relevance and quality of saliency maps. Consequently, visualizing the generated maps is also crucial. In Figure 5.2 we observe that maps generated by our TIS method are generally more expressive, often highlighting the whole object with a variable range of intensity, for example with the Maltese dog where the head of the dog is the most highlighted, followed by the dog's body with intermediate intensity, and then the background

[6] https://github.com/oliviaguest/gini

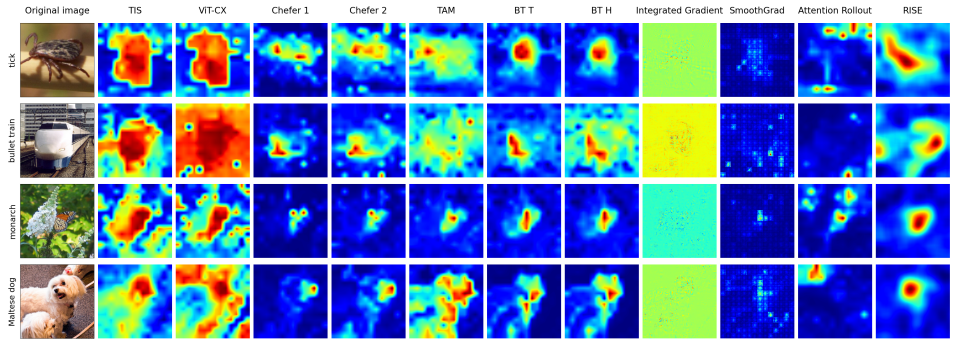


Fig. 5.2 Comparison of the explainability methods for the ViT-Base model [36] on four random images from the ImageNet Validation set [136].

with low intensity. In general, other methods tend to be more categorical with a generally very localized high signal and most of the remaining of the map being low signal. ViT-CX and TAM are the only methods that seem to also display this behavior, while ViT-CX often highlight more background information and TAM is sparser. In contrast, the Integrated Gradient and the SmoothGrad methods produce maps with a lot of isolated peaky points, related to the importance of the gradient at the input. They are not always class specific and tend to be noisy and hard to interpret.

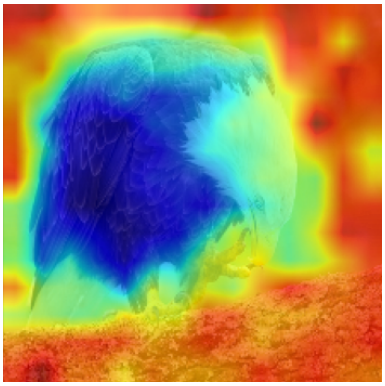
Class Disagreement

When generating the saliency maps for both the target class from the ImageNet Dataset and the model predicted class, we noticed that major disagreements between the ground truth and the model can lead to bad saliency maps for the target class, and good saliency maps for the model predicted class. An example is provided in Figure 5.3 where a bird with a target class of "Kite" is present, the model top prediction is "Bald Eagle" with a confidence level of 0.998, while the confidence of the target class is 0.0004. The saliency map for "Bald Eagle", the predicted class, clearly highlights the bird, while the saliency map for "Kite", the target class, highlights the background. We observed this behavior for multiple images, the stronger the disagreement between the model and the target, the stronger this phenomenon. Through our experiments, we discovered that highlighting the target class can be forced by removing the softmax at the end of the model. However, this comes at the price of class specificity. This behavior is thus strongly related to the class specificity of the method, leading us to interpret it as proof of our method's strong class specificity.

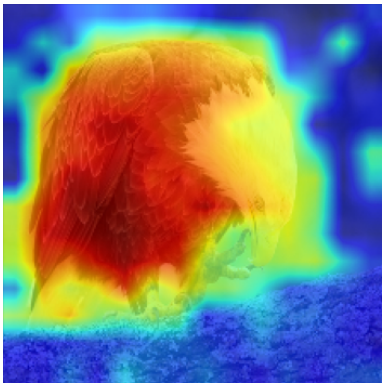
(a) Original image

(b) Kite

(c) Bald Eagle



(d) TIS for "Kite"



(e) TIS for "Bald Eagle"

Fig. 5.3 Class mismatch between the target and predicted class. 5.3a is the original image. The dataset target class is "Kite" while the model predicts "Bald Eagle". For illustration purposes, 5.3b and 5.3c display other images of a Kite and a Bald Eagle, respectively. 5.3d is the saliency map produced by TIS for class "Kite" (dataset target) and 5.3e is the TIS saliency map for the model predicted class "Bald Eagle".

Table 5.1 Results of the Insertion and Deletion metrics and their difference (Insertion - Deletion) for ViT-Base [36]. [7]

	Insertion ↑				Deletion ↓				Insertion - Deletion ↑			
Method	Mean	Blur	Black	Rand	Mean	Blur	Black	Rand	Mean	Blur	Black	Rand
TIS	**0.52**	0.66	**0.50**	**0.47**	0.10	0.39	0.10	0.09	**0.42**	0.28	**0.40**	**0.38**
ViT-CX	0.51	0.61	0.41	0.39	0.20	0.42	0.14	0.18	0.28	0.20	0.31	0.35
TAM	0.43	0.61	0.41	0.39	0.14	0.43	0.14	0.13	0.28	0.18	0.27	0.26
Chefer1	0.42	0.61	0.41	0.39	0.15	0.44	0.14	0.13	0.28	0.17	0.27	0.26
Chefer2	0.43	0.61	0.41	0.39	0.15	0.44	0.14	0.13	0.28	0.17	0.27	0.26
Att. Rollout	0.31	0.55	0.30	0.29	0.29	0.52	0.28	0.27	0.02	0.03	0.02	0.02
BT H	0.45	0.63	0.43	0.41	0.12	0.41	0.12	0.11	0.33	0.21	0.32	0.30
BT T	0.46	0.62	0.44	0.42	0.13	0.42	0.12	0.11	0.33	0.21	0.32	0.30
RISE	0.46	0.62	0.45	0.42	0.16	0.45	0.16	0.15	0.30	0.17	0.29	0.27
IntegratedGrad	0.19	**0.69**	0.16	0.15	**0.08**	**0.31**	**0.06**	**0.06**	0.11	**0.38**	0.10	0.08
SmoothGrad	0.37	0.59	0.36	0.35	0.10	0.45	0.10	0.09	0.27	0.14	0.26	0.26

Table 5.2 Results of the Insertion and Deletion metrics and their difference (Insertion - Deletion) for DeiT-Base [157]. [7]

	Insertion ↑				Deletion ↓				Insertion - Deletion ↑			
Method	Mean	Blur	Black	Rand	Mean	Blur	Black	Rand	Mean	Blur	Black	Rand
TIS	**0.57**	0.65	**0.57**	**0.54**	0.15	0.40	0.15	0.14	**0.42**	0.25	**0.42**	**0.41**
ViT-CX	0.51	0.61	0.51	0.48	0.20	0.42	0.20	0.18	0.31	0.19	0.31	0.30
TAM	0.50	0.59	0.50	0.46	0.23	0.45	0.23	0.19	0.27	0.14	0.26	0.26
Chefer1	0.51	0.60	0.51	0.48	0.22	0.45	0.22	0.18	0.29	0.15	0.29	0.29
Chefer2	0.50	0.60	0.50	0.47	0.23	0.45	0.23	0.19	0.28	0.14	0.27	0.28
Att. Rollout	0.37	0.54	0.37	0.34	0.41	0.53	0.41	0.37	-0.04	0.01	-0.05	-0.03
BT H	0.52	0.60	0.52	0.49	0.19	0.43	0.19	0.16	0.33	0.18	0.33	0.33
BT T	0.52	0.60	0.51	0.48	0.19	0.43	0.19	0.16	0.33	0.17	0.32	0.32
RISE	0.55	0.61	0.55	0.52	0.25	0.46	0.25	0.21	0.30	0.15	0.30	0.31
IntegratedGrad	0.32	**0.68**	0.30	0.28	**0.14**	**0.38**	**0.12**	**0.13**	0.18	**0.30**	0.18	0.15
SmoothGrad	0.45	0.62	0.43	0.43	**0.14**	0.44	0.14	**0.13**	0.31	0.18	0.30	0.31

5.4.2 Quantitative Assessment

Faithfulness Results for Insertion and Deletion metrics are provided in Table 5.1 and Table 5.2 for ViT and DeiT, respectively. Our proposed method performed best on the Insertion for all baselines, except the blur baseline where it finished second behind Integrated Gradient by a thin margin. Interestingly, it's worth noting that Integrated Gradient had the worst performance among all methods for the other Insertion baselines. Concerning the Deletion metric, our method performed second, just behind Integrated Gradient. This is not surprising since the gradient on which Integrated Gradient is based corresponds to the pixels with the highest influences on the output. When balancing the two metrics by the subtraction of the Deletion metric from the Insertion metric, our method appears to surpass other methods by a wide margin for all baselines, except for the blur baseline where it finishes second.

Localization The results for the Pointing Game metric can be found in Table 5.3. Our method performed best in comparison to the other methods for DeiT on this metric and fell just behind the BT methods for ViT. Furthermore, TIS is the

only method that achieves a score over 0.8 on both models (0.825 and 0.823). Our proposed method is thus competitive in terms of the localization property.

Robustness In Table 5.4, we show the results related to the Max Sensitivity metric. Two groups emerge from these results. The first group contains RISE, TAM, BT-H, Chefer2 Rollout, TIS and ViT-CX (ranked by lowest sensitivity score respectively) and has good robustness when small perturbations are inputted to an image (Max Sensitivity ≤ 0.2). On the contrary, Integrated Gradients and Chefer1 in the second group are at the other end of the range (Max Sensitivity ≥ 0.8), being very sensitive to perturbations. TIS has appropriate scores with respect to the metric (not being too sensitive) but is not the best method in terms of robustness.

Deletion for TiS and Integrated Gradients

Based on the results indicating that Integrated Gradients may outperform TIS in terms of the Deletion metric, we explored the results obtained by both methods when applying the deletion metric to an image (Figure 5.4). Integrated Gradients exhibit a faster drop in the metric and achieve a better overall result. However, upon examining the perturbed image at intermediate steps, it became apparent that Integrated Gradient significantly affects the model by removing target pixels everywhere in the image, while the overall shape of the bird remains distinguishable to a human observer. In contrast, TIS effectively masks the object.

Table 5.3 Results of the Pointing Game metric [180] for the ViT [36] and DeiT model [157]. [7]

Method	DeiT	ViT
TIS	**0.825**	0.823
ViT-CX	0.700	0.700
TAM	0.635	0.737
Chefer1	0.748	0.768
Chefer2	0.654	0.727
Attention Rollout	0.118	0.127
BT H	<u>0.775</u>	**0.855**
BT T	0.755	<u>0.846</u>
RISE	0.766	0.753
Integrated Gradient	0.297	0.633
SmoothGrad	0.742	0.499

5.5 Discussion

In this chapter, we introduced a method to explain vision transformers using token sampling guided by the model embeddings. This is an alternative to methods

[7] The best result is in bold, and the second best result is underlined

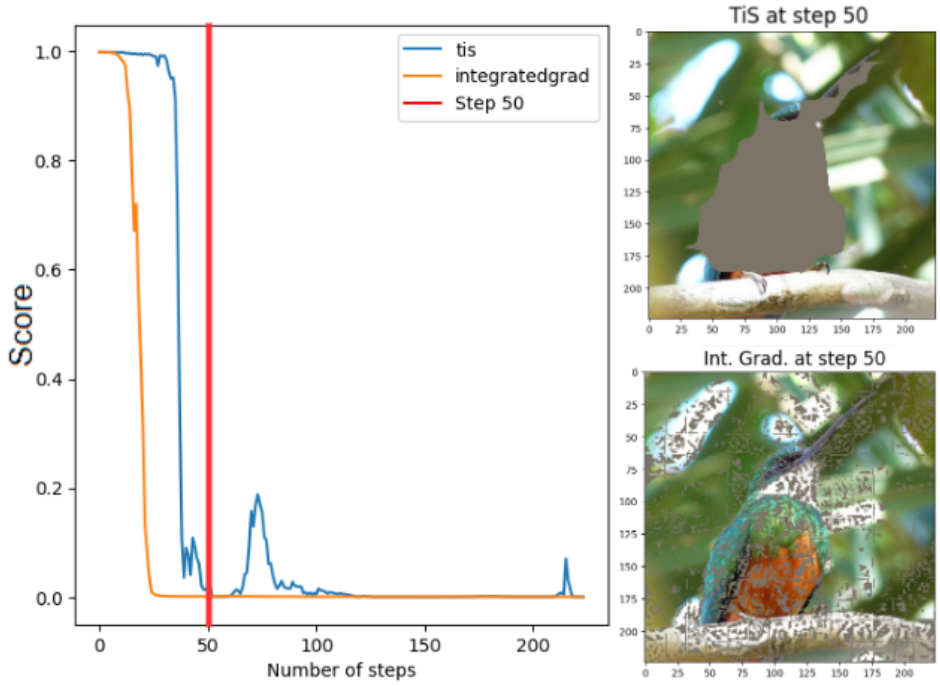


Fig. 5.4 Comparison of TIS and Integrated Gradients results after 50 steps of the deletion metric. The target class is a jacamar (bird). Integrated Gradient perturbs the image diffusely, resulting in a better metric while still keeping the bird visible. On the other hand, TIS masks the bird itself, even though it may take more steps to reduce the target score.

Table 5.4 Results of the Max Sensitivity metric [177] for the ViT [36] and DeiT model [157]. [7]

Method	DeiT	ViT
TIS	0.162	0.156
ViT-CX	0.173	0.172
TAM	<u>0.085</u>	<u>0.060</u>
Chefer1	1.017	0.752
Chefer2	0.087	0.082
Attention Rollout	0.143	0.144
BT H	0.088	0.620
BT T	0.086	0.062
RISE	**0.011**	**0.009**
Integrated Gradient	0.827	0.891
SmoothGrad	0.218	0.412

based on attention and gradients to explain transformers. The main contribution of our method in comparison to other perturbation methods, such as RISE or VIT-CX, is to provide a more versatile and complete ablation of masked input information instead of masking in input space. Even if the absence of a real ground truth metric in the explainability field makes the evaluation difficult, we showed the competitiveness of our method amongst all metrics with current explainability method. A common downside of perturbation-based methods is the requirement for more computing power, as multiple forward passes must be performed.This limits the application in use cases such as low-power or embedded devices. TIS shows good performances with as few as 128 samples and half of the tokens, significantly reducing the inference time. Although this work has only explored vision transformers, our method also has the advantage of being potentially applicable to any type of transformer using conventional encoding and/or decoding layers. Although, on the other hand, it is not directly applicable to modified transformers with hierarchical mechanisms such as a Swim transformer [100]. Since TIS is not limited by design to vision transformers, future works should explore the adaptation of the token sampling to transformers working with other modalities and/or multi-modal transformers.

In practice, in the context of this thesis, the work was limited to the development of the vision method and did not continue beyond that point, in order to focus on the other thesis objectives concerning multimodal self-supervised training. The UMons team, with whom I collaborate, has since taken the lead in continuing the development of a multimodal adaptation.

6

Practical use case of PolyCAM: Bone radiography analysis and Deep learning biases

6.1 Introduction

The main idea of using explainability methods in this thesis is to get an insight into elements used by the model to produce a results, for example a diagnosis.

The objective of this chapter is to explore the application of the PolyCAM method introduced in previous chapter to identify if radiographic features used by the algorithm align with the actual pathology or if the model relies on irrelevant correlations, indicating potential biases. The hypothesis is that the deep learning model may rely on spurious features, rather than true pathological indicators.

6.2 Materials and Methods

6.2.1 Dataset and model training

The MURA (Musculoskeletal Radiographs) dataset, which is a retrospective collection publicly available through Stanford University, was used [127]. It comprises approximately 40,005 bone radiographs of the upper extremity, organized

into 14,656 studies with normal or abnormal classifications. For this study, we employed the original validation set containing 1,199 studies and 3,197 images as a test set and randomly divided the remaining images at the study level into five folds (subsets of the data used for cross-validation). The employed model was a ResNet50 convolutional neural network, which was initialized using the pre-trained weights from IMAGENET1K_V1 in torchvision 0.15 [51]. The training process utilized PyTorch 2.0 [118] on an Nvidia RTX 3070 with a batch size of 16 for 20 epochs, following a 5-fold cross-validation approach (where each fold served as the validation set and the remaining four folds were used for training). The Adam optimizer was employed with an initial learning rate of 6e-5, cosine annealing scheduling without restart, and a weight decay of 1e-5. A weighted binary cross-entropy loss function was utilized in conjunction with random rotation between -15° and +15° and resizing to 320x320 pixels as data augmentation techniques. The selection of the learning rate and weight decay was guided by the use of Optuna framework [6] incorporating a multivariate Tree-structured Parzen Estimator [41].

6.2.2 Explanation

To visualize the relative importance of pixels in radiograph images for deep learning model predictions, saliency maps were generated using the PolyCAM algorithm [40] for each image from the first 200 wrist studies in the test set. This algorithm combines CAM-like properties with pixel perturbations to produce high-resolution saliency maps.

These heatmaps were visually examined to identify the specific image elements used by the neural network for prediction and diagnosis, as well as whether they corresponded to actual pathological areas or were influenced by other image elements.

6.2.3 Bias correction

Based on the findings presented in the Results Section (i.e. that the presence of casts considerably interfered with the model's ability to correctly identify bone pathology), we implemented an additional debiasing step. The primary objective of this step was to address the issue where casts were being utilized as decisive factors by the model, frequently exceeding the importance of the actual pathology itself.

It can be theorized that presenting additional radiograph images featuring casts that are not considered pathological could alter the model's erroneous behavior towards casts by disrupting the correlation between the presence of a cast and actual pathology. To verify this hypothesis, we reviewed 8,760 radiographs from the training set to identify those containing casts. These images were then cropped using GIMP (GNU Image Manipulation Program) v2.10.34 to create new

images where only the cast is visible, without any accompanying pathology. A visual example is illustrated at Figure 6.1. These cropped images were subsequently added back into their original folds (to prevent potential contamination between folds and thus between the training set and validation set), with a label updated to "not pathological". Each cropped cast image was duplicated 20 times to amplify the debiasing effect.

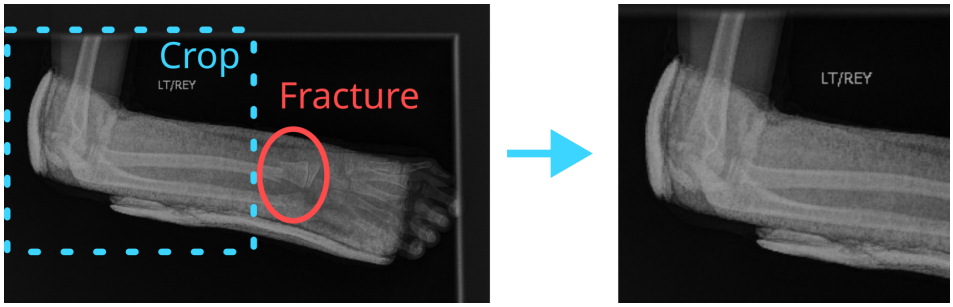


Fig. 6.1 Illustration of the cropping operation. A new image is produced by keeping an area of the radiograph without the bone pathology and with a cast.

The training procedure previously described was re-executed using the revised dataset and new saliency maps were generated for the debiased models.

An examination of the cast's utilization as primary pathological element was conducted for both the original and modified models. This entailed counting the number of saliency maps where most high-salience values are situated on the cast among images classified as pathological by each model.

6.2.4 Performances Evaluation

To evaluate the performance of the models, we employed bootstrapping with 1,000 resamples with replacement from the test set. We computed the Area Under the Receiver Operating Characteristic (ROC) curve, the Area Under the Precision-Recall curve, accuracy, and F1 score on the ensemble of models across the five folds using average output aggregation for each resample, thereby generating a confidence interval for each metric. This analysis was performed utilizing scikit-learn 1.3.0.

6.3 Results

The results are divided in two parts, the first part presents the results using the MURA dataset as provided by Stanford without any modification, while the second part presents the results for the modified dataset to reduce the bias related to the presence of a cast on the input radiograph.

Metric	Original	Modified
AUROC	0.896 (0.876, 0.914)	0.895 (0.876, 0.913)
AUPRC	0.886 (0.859, 0.912)	0.886 (0.857, 0.908)
Accuracy	0.840 (0.816, 0.861)	0.839 (0.820, 0.861)
F1 score	0.805 (0.777, 0.833)	0.806 (0.780, 0.833)

Table 6.1 Performances of the ensemble of models form the 5-fold on the test-set, with a training on the original and the modified datasets. Results are shown as the median of a 1000 resample with replacement of the test-set with the 95% confidence interval between parenthesis. AUROC = Area Under the Receiver Operating Characteristic Curve, AUPRC = Area Under the Precision-Recall Curve.

6.3.1 Classical dataset

Pathology detection performances of the base models

The Area under the ROC curve on the test set for the 5-folds ensemble was 0.896 (IC95: 0.876-0.914), the AUPRC was 0.886 (IC95: 0.859-0.912), the accuracy was 0.840 (IC95: 0.816-0.861) and the F1-score was 0.805 (IC95: 0.777-0.833). Table 6.1 present those results and a comparison with results for the modified dataset.

Saliency maps of base models

The 200 studies analyzed with PolyCAM corresponded to 549 radiographs and an equal number of saliency maps. The elements most often highlighted as pathological were fractures, articular joint disorders, osteosynthesis material, but also the presence of a cast and sometime the annotations (e.g. side indicated with a letter L or R), see Figure 6.2 for an overview.

Concerning the casts, 46 dense/plaster-like casts and 13 lighter/synthetic-like casts were manually identified on the images, a total of 59 images containing casts. In the 200 images, the cast was the most prominent element in 31 images (30 plaster and 1 synthetic cast).

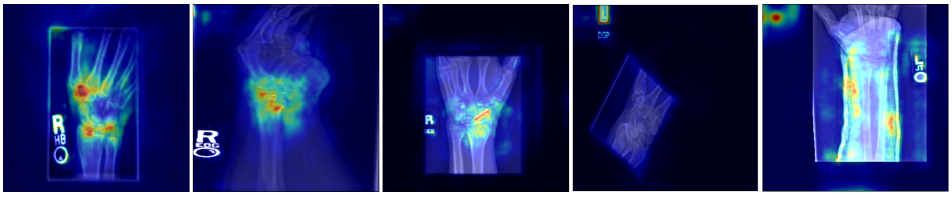


Fig. 6.2 Illustration of the most frequently identified elements in saliency maps, from left to right: Articular joints/osteoarthritis, fractures, osteosynthesis material, annotations (e.g., side indicators), and casts.

6.3.2 Modified dataset

From the review of 8,760 radiograph images, 1,021 instances featuring casts were identified (247 elbow, 27 finger, 186 forearm, 96 hand, 77 humerus, and 388 wrist). This led to the generation of 587 cropped images (168 elbows, 20 fingers, 169 forearms, 89 hands, 58 humeri, and 83 wrists) from which casts were isolated. Cropped images were only obtained from suitable base images—those where it was possible to isolate a large enough area with the cast but without pathology to produce a suitable cropped image. Each cropped image was duplicated 20 times within its original fold as part of the modified dataset.

Pathology detection performances of the modified models

The results are shown in Table 6.1. The metrics do not differ statistically from the results of the models trained on the untouched dataset.

Saliency maps of the modified models

The number of casts triggering a pathological label prediction on the 200 visualized images decreased from 31 with models trained on the original dataset to 12 with models trained on the modified dataset. Compared to the saliency maps produced by the base model, the modified models exhibited reduced correlation between cast presence and pathology detection, either due to increased emphasis on other image elements (e.g., fractures) as depicted in Figure 6.3 or because predictions were no longer systematically pathological when alternative features were not identified. While this process significantly mitigated artifacts, some casts still remained highlighted after debiasing, as illustrated in Figure 6.4.

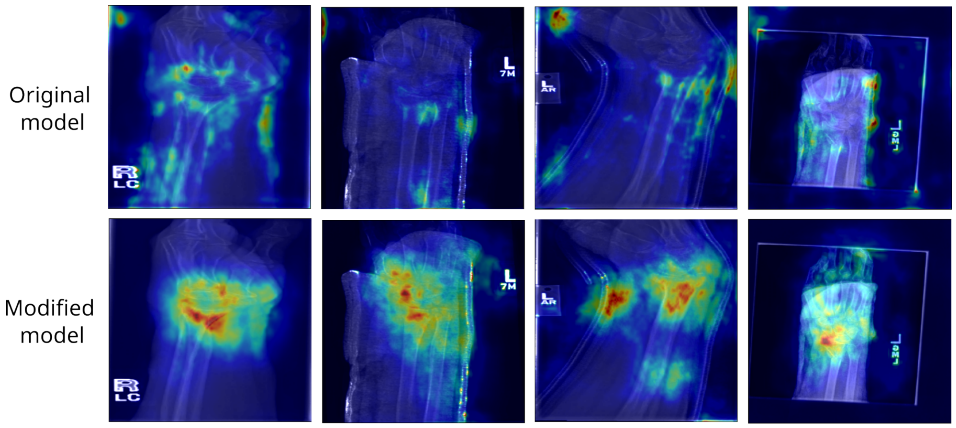


Fig. 6.3 Debiased Model Attention and Reduced Bias. Comparison of saliency maps from ResNet50 trained with and without additional cast images. The model's attention is redirected towards the bone, rather than the cast, after addition of cast images into the dataset.

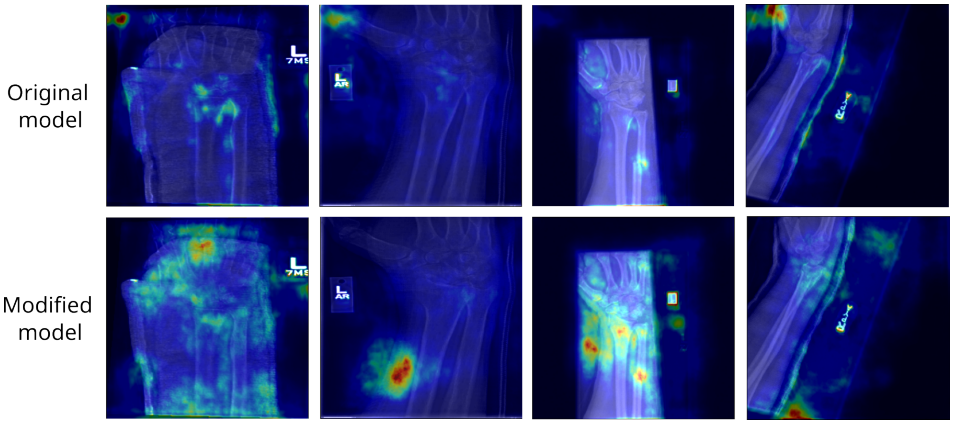


Fig. 6.4 Failure cases and incomplete debiasing. Saliency map of ResNet50 trained with and without additional cast images. Even if the cast bias is reduced by the process, some level of bias persist in certain radiographs.

6.4 Discussion

In summary, addressing sources of bias in deep learning models, especially in medical imaging, is essential for ensuring accurate and fair diagnoses. Our PolyCAM method demonstrates that it's possible to identify such biases, particularly those arising from irrelevant features like casts. By incorporating additional non-pathological images containing casts, we can reduce the model's reliance on these elements without compromising its ability to detect pathologies.

Moving forward, it is important to acknowledge the limitations of the MURA dataset used, which only provides binary labels (normal/abnormal). While this dataset is the largest publicly available for bone radiographs, the lack of more detailed labels restricts the depth of analysis and model training.

To overcome these limitations, the next part of the thesis will explore the use of self-supervision techniques coupled with the creation of a larger dataset from our hospital, allowing us to leverage the data's scale and complexity without requiring manual annotation.

PART II
Vision Language Self-Supervion: Using existing reports as supervision

7

Fundamentals of Self-Supervised Learning and Vision-language

7.1 Introduction

As the field of artificial intelligence continues to evolve, self-supervised learning (SSL) has emerged as a transformative approach, particularly in environments where annotated data is limited or costly to acquire. By exploiting the inherent patterns in data, SSL enables the development of robust representations without the extensive need for labeled datasets, making it particularly promising for applications in medical imaging and natural language processing.

The role of self-supervision is to learn effective representations by capturing inherent patterns in data rather than relying on human-based annotations. Various techniques are available and tailored to specific modalities. In the context of this thesis, the pertinent modalities are imagery and text, which will be the primary focus.

In medical imaging, annotated datasets are sparse, expensive, and often protected by confidentiality agreements. Traditional supervised learning approaches struggle under these conditions, as demonstrated in the previous chapter using the MURA dataset, creating a significant barrier to the advancement of AI in healthcare. SSL and Vision-Language Pretraining (VLP) offer a compelling alternative by leveraging unannotated data, opening new avenues for innovation in medical AI.

This chapter aims to provide a comprehensive background that sets the stage for the methods explored in this thesis to utilize raw data from the Cliniques Universitaires Saint Luc without extensive manual annotation.

7.2 Unimodal self-supervision

Unimodal self-supervised learning focuses on developing models using single types of data, such as text or images. These methods are crucial as they form the foundation upon which multimodal models are built.

Text Self-Supervision In the field of natural language processing, two self-supervised methodologies stand out: Masked language modeling and causal language modeling.

Masked text modeling involves masking a certain percentage of words in a sentence and training a model to predict these masked words based on the context provided by the surrounding words. For example, in the sentence "The patient underwent a total ____ arthroplasty surgery", the model should predict that the masked word is likely to be a type of joint, such as "knee" or "hip". BERT (Bidirectional Encoder Representations from Transformers) is a well-known model that uses this technique [35]. Some newer models such as RoBERTa (Robustly optimized BERT approach) are similar to BERT but include optimizations such as more extensive training data and larger batch sizes [99].

Causal language modeling (also called generative pretraining), on the other hand, involves training a model to predict the next word in a sentence, thereby building a coherent sequence of text. Given a sentence fragment such as "The surgeon inserted a nail in the ____", the model is trained to predict what comes next, such as "femur". GPT (Generative Pre-trained Transformer) is central to this approach [125, 17]. Successive versions, such as GPT-2, GPT-3, or GPT-4, include larger datasets and more complex architectures, which improve their language generation capabilities. Several open-source basic models exist as alternatives to closed models, such as the LLaMA models [158] or Mistral/Mixtral [67, 68], which offer similar capabilities with different architectures and training datasets.

Image Self-Supervision In computer vision, self-supervised learning techniques are designed to capture the visual structure inherent in images. Contrastive learning involves training the model to distinguish between different representations of the same image and those of different images. By applying enhancements such as cropping, color jittering, and rotation to an image, multiple versions or "views" of that image are created. The model then learns to bring the representations of these views closer together while pushing the representations of different images apart, as shown in Figure 7.1. SimCLR (Simple framework for Contrastive Learning of visual Representations) uses this method [25].

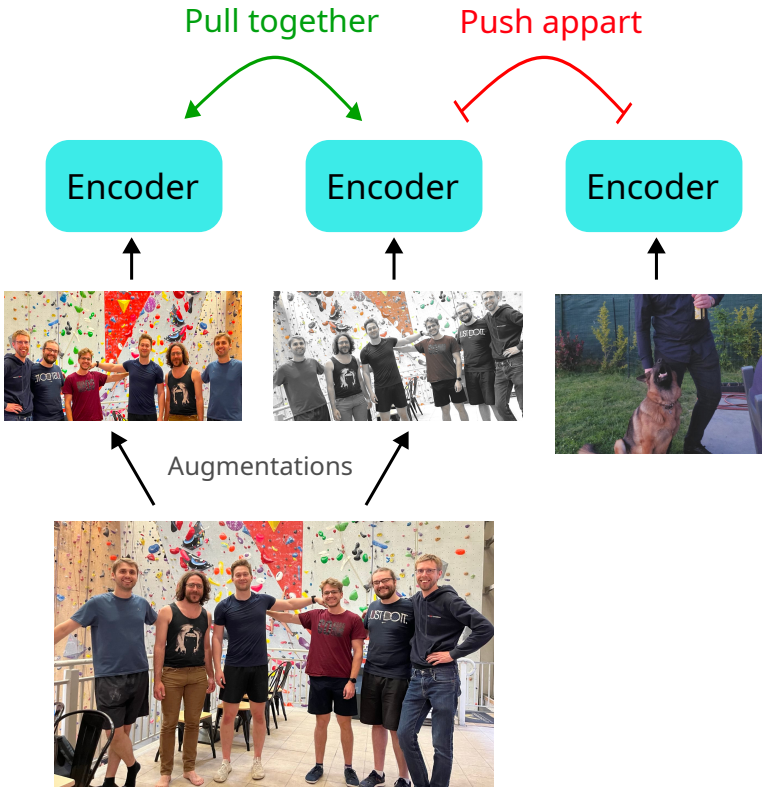


Fig. 7.1 Contrastive loss for image self-supervision. An image is transformed through various augmentations, producing different views. The model aims to "pull together" (encode similarly) augmented versions of the same image while "pushing apart" (encode dissimilarly) augmented versions of different images. This method helps the model recognize the underlying features of images for better self-supervised learning.

Self-distillation Siamese networks, such as DINO (Self-DISTILlation with NO labels), involve two neural networks where one serves as a "teacher" and the other as a "student." The teacher network, often a moving average of the student, provides pseudo-labels to guide the student network [18]. The student learns to match the teacher's output for different augmented views of the same image.

Masked image modeling techniques, such as Masked Autoencoder (MAE) and SimMIM, involve predicting the missing parts of an image from the visible portions, inspired by the principles behind masked text modeling [50, 173]. For instance, given an occluded radiographic image, the model learns to reconstruct the masked or missing parts accurately.

These unimodal methods not only advance individual fields of natural language processing and computer vision but also form the basis for more complex,

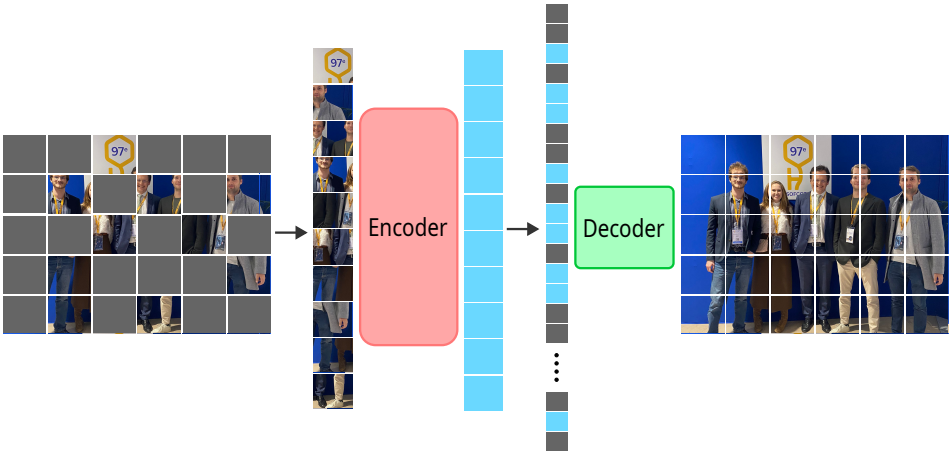


Fig. 7.2 Masked Image Modeling Process. The model predicts the masked parts of an image based on the visible context.

integrated vision-language models.

7.3 Multimodal self-supervision

Vision-Language Pretraining (VLP) is a specialized form of self-supervised learning that simultaneously trains image and text representations by leveraging the relationship between a given image and its related text. This synergy is essential for tasks such as medical report generation, where understanding both visual and textual data is crucial.

One common approach to VLP is contrastive learning, where models like CLIP [124], ALIGN [65], DeCLIP [92], and GLIP [91] are trained to align images with their corresponding descriptions. This involves creating pairs of images and texts that should be pulled closer in the embedding space while pushing non-matching pairs apart. The contrastive learning framework thus learns meaningful cross-modal representations by contrasting positive (matching) and negative (non-matching) pairs, as illustrated in Figure 7.3.

In addition to contrastive learning, other methods employ pseudo-tasks such as alignment prediction and masked multi-modal modeling. Models like ViLBERT [102] and VisualBERT [90] use these techniques to predict whether a given image and text pair are aligned or to fill in missing elements in either modality.

Some advanced models combine multiple unimodal and multimodal self-supervision techniques to enhance their performance. For instance, Flava [147] incorporates various self-supervised tasks to create robust joint representations of images and texts. The fusion between modalities can be performed at different stages of the model architecture. For example, early fusion models like

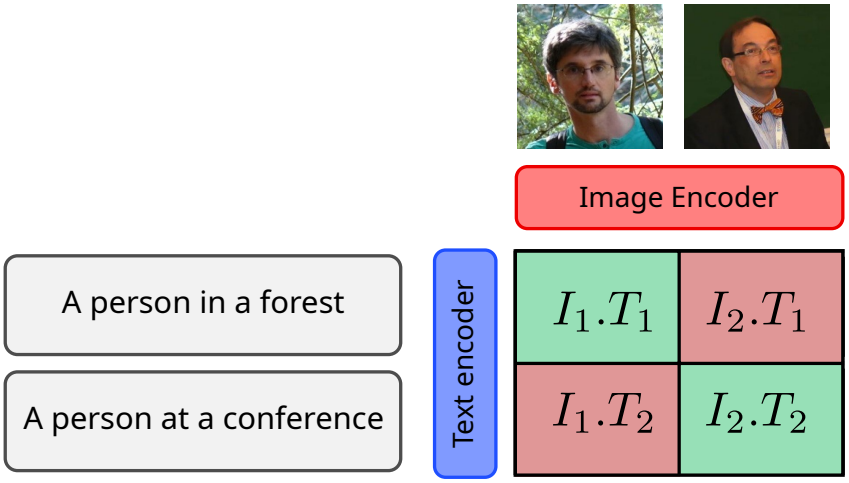


Fig. 7.3 Process of Vision-Language Pretraining with contrastive learning. Two images and their corresponding textual descriptions are encoded through respective image and text encoders. The model then aims to align the matching image-text pairs (I1, T1 and I2, T2) while distinguishing non-matching pairs (I1, T2 and I2, T1) by bringing matching pairs closer and pushing non-matching pairs apart in the embedding space.

VilBERT [102] and VisualBERT [90] use a common encoder capable of handling both text and images, while late fusion models like CLIP [124] employ separate encoders for each modality, which are then integrated at a later stage. Intermediate fusion strategies also exist, where partial integration occurs at different levels within the model [88, 147].

Training these models usually requires large-scale datasets of images and their corresponding captions or descriptions. ALIGN [65], for instance, is trained on 1.8 billion image-text pairs, showcasing the extensive data requirements for effective VLP.

Image captioning is another tactic for VLP, where models such as SimVLM [168] and Virtex [32] are designed to generate textual descriptions for images. These models learn to capture the semantics of both images and texts by training on large-scale datasets containing image-caption pairs. This is achieved through an image encoder that embeds images as tokens, which can then be processed by a transformer, all trained on a causal language modeling task.

Building on this approach, other models leverage pre-trained large language models, initially trained on text-only causal language models, and learn to integrate images that align with the language model's existing representation. This enables the exploitation of vast text datasets and the high performance of pre-trained large language models, as well as the specificity of smaller text-image datasets. A schematic representation of a classical vision-language model pipeline

is presented in Figure 7.4.

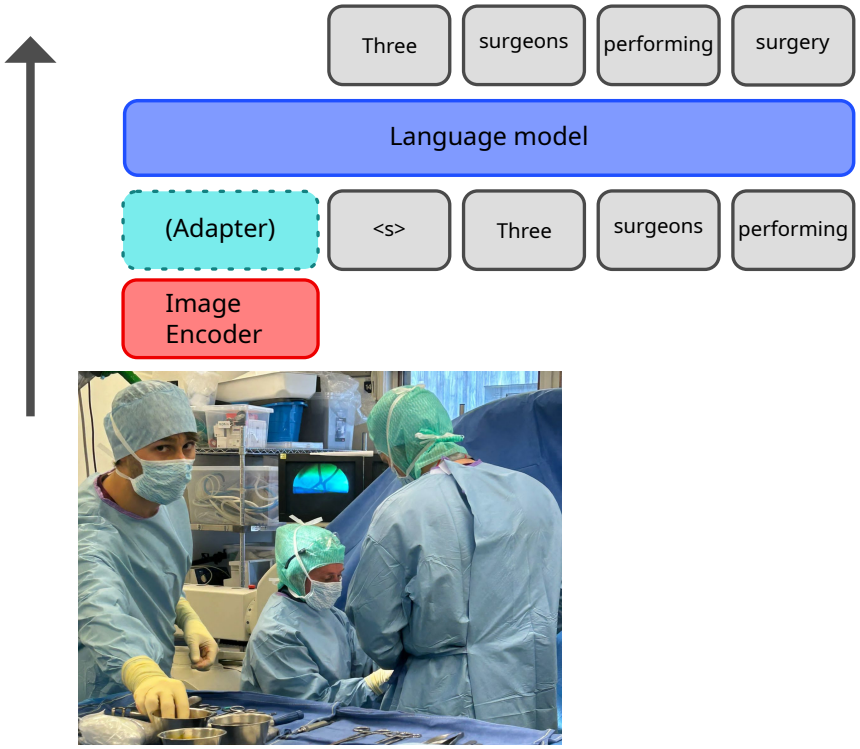


Fig. 7.4 A schematic representation of a vision-language model architecture for caption generation. The image showcases the integration of an image encoder and a language model using an intermediate adaptive layer. Tokens generated from the image using the image encoder are passed through the adapter before being fed into the language model in addition to previously generated text tokens, if any, or a special beginning of sentence token. The language model then iteratively predicts the next token of the caption.

This can be achieved while keeping the language model frozen using methods such as Frozen [159], or by keeping both the image encoder and text decoder frozen and training an intermediate adaptive layer to transfer information from the image to the text model using methods like BLIP2 [87] or Flamingo [7].

Some models, such as Llava [97, 96], take a more flexible approach by unfreezing the language model and proposing a method to preprocess image-caption pairs to produce a synthetic visual question answering dataset using a pre-trained large language model. Other models, like idefics2 [81], focus on improving image captioning and visual question answering performance by incorporating additional training objectives and techniques, this particular model is trained on the Obelics dataset [80], a large-scale dataset of object-centric image captions.

More recent models, such as Florence 2 [171], have explored the use of VLP for a broader range of tasks, including image-text retrieval, visual question answering, object detection, segmentation, dense captioning, phrase grounding, and referring expression comprehension.

7.4 Medical applications of self-supervised Vision-Language Pretraining

Multiple adaptations of VLP methods to clinical datasets have been envisioned. ConVIRT [181] is a precursor in the application of contrastive VLP. Recent advancements have extended beyond global contrastive alignment between image and text by the incorporation of local alignment as exemplified by GLORIA [60], LoVT [110], MGCA [163] or PRIOR [27].

Most previous works on medical VLP have been validated on chest radiographs, using (Bio)ClinicalBERT [8] as a text encoder. (Bio)ClinicalBERT has been trained on medical reports from the MIMIC III dataset [71] and shows superior performance in comparison to a biomedical model, trained on biomedical domain corpora such as PubMed abstract and PMC full-text articles, like BioBERT [83]. Both ClinicalBERT and BioBERT are trained on English texts.

In contrast, our work considers French documents and bone radiography. This poses multiple challenges since, at the time of writing, many useful tools are English-only (e.g. CheXpert labeler [62], RadGraph [63], negBIO [119]).

To exploit French medical documents in the frame of a vision-language pretraining, we considers two alternatives in the next chapter, corresponding to French-only models and multi-lingual models.

On the one hand, French-only models, such as CamemBERT [105], have been adapted to the biomedical domain with models like Dr BERT [77] or CamemBERT-BIO [156]. However, the amount of data used for training these models is smaller to their English counterparts (e.g., 4.5B + 13.5B words for BioBERT, 3.1B words for PubMedBERT [48], versus 1B for NACHOS used by DrBERT, and 413M for CamemBERT-bio).

On the other hand, multimodal languages such as mBERT [35], XLM-Roberta (XLMR) [29], or MLUKE [131] benefit from having more data for pretraining than French-only models. Moreover, they enable cross-lingual transfer of knowledge. Methods like Self-alignment pretraining (Sap) [95] have been applied to multilingual general models such as XLMR [29], showing promising performance on tasks such as Biomedical Entity Linking. This pretraining involves aligning the embeddings of synonyms of concepts from the Unified Medical Language System (UMLS), a compendium that integrates and harmonizes various medical terminologies and classifications. Models of this kind become particularly valuable when biomedical resources for a specific language are scarce.

8

8.1 Introduction

In the medical domain, particularly in radiography, large-scale datasets are generally limited to English reports and to specific body areas. To the best of our knowledge, the only large publicly available radiography-report dataset is MIMIC-CXR[70], containing 377,110 Chest Xray images and their corresponding free-text reports in English. This raises a significant challenge when applying the models derived from those data to images other than Chest Xrays.

Moreover, privacy regulations such as the General Data Protection Regulation (GDPR)[130] impose strict limitations on the distribution and sharing of medical databases containing sensitive patient information. To address this limitation, one viable approach would be to utilize local data available within a given hospital or healthcare institution. Hospitals typically maintain their own databases of medical images and associated reports, which are collected as part of routine clinical practice. While these local datasets may not be as extensive as publicly available datasets, they still contain valuable information that can be leveraged for training and evaluating machine learning models.

Therefore, in this chapter, we propose to exploit bone X-Rays paired with reports sourced from the Saint Luc University Hospital. This is achieved in two steps. First, the latent spaces associated with deep vision and language encoders are aligned using 219,675 paired studies from our hospital (corresponding to 789,397 individual X-ray images), resulting in a pretrained vision and language model that is shown to outperform alternative baselines when fine-tuned on a downstream task benchmark. Second, pseudo-labels are extracted from the textual reports, using a generative large language model, to learn how to solve a

specific visual task based on the features computed by the pretrained vision encoder, without any manual annotation. Additionally, our work demonstrate the feasibility of the approach with another language than English. Therefore, it develops a novel open source method for pseudonymizing French medical reports.

The main contributions in this chapter are:

- We provide a comprehensive guide on leveraging medical data from a single hospital, outlining our step-by-step approach from raw data to a trained model, with a particular focus on the anonymization process, and demonstrating how self-supervised learning can be effectively applied to this data. In particular, a French adaptation of the DEDUCE [107] method was developed and is made available[1] to facilitate the pseudonymization of medical reports in French.

- We leveraged bone radiographs and their associated French reports to pretrain a versatile vision-language model, to be used as a backbone for a variety of tasks trained with limited supervision. The obtained representation, when fine-tuned to address a downstream task, is shown to result in performance that are competitive with models trained with a significantly larger amount of human supervision.

- To bring the vision-language self-supervision beyond pretraining, we trained a bone fracture detection without any manual annotation using pseudo-labels extracted from the local radiology reports by a large language model.

This work is submitted for review as a journal paper.

8.2 Methodology

This section first presents data preprocessing in Section 8.2.1, followed by Vision-Language pretraining in Section 8.2.2, and pseudo-label training in Section 8.2.4.

A general graphical overview is presented in Figure 8.1.

8.2.1 Data preparation

This section describes the steps envisioned to create datasets that are relevant for vision language pretraining, and bone fracture pseudo-label generation, respectively.

In order to protect the privacy of patients, following the GDPR[130], anonymization techniques are employed when possible, and we resort to pseudonymization when complete anonymization is not feasible.

[1] `https://github.com/aenglebert/deduced`

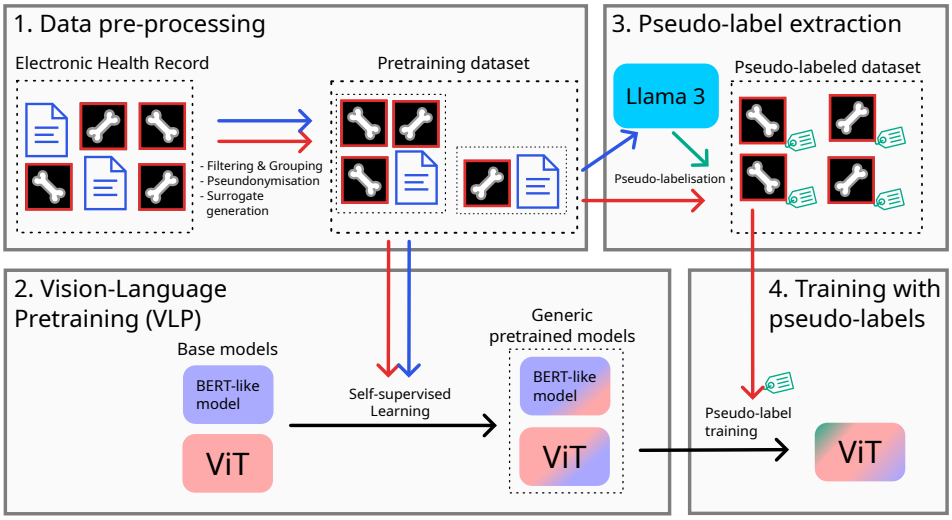


Fig. 8.1 General overview: The Electronic health records (EHR) data is leveraged to generate a pretraining dataset (Block 1), enabling self-supervised vision-language pretraining of generic backbone models (Block 2). Subsequently, these backbones can be fine-tuned using external datasets (not depicted in the figure), as demonstrated in Section 8.3.3. To further develop the self-supervision capabilities, a large language model is used to derive task-specific pseudo-labels (Block 3) that are used to train the head of the pretrained vision model backbone to solve the task of interest (Block 4).

Images preprocessing

Images from a hospital are typically stored in a PACS (Picture Archiving and Communication System). To address situations where imaging devices embed text containing sensitive information within images, such as patient names in dose reports, the EasyOCR[2] framework is utilized to detect and extract text from images, with the goal of identifying potentially problematic images.

Subsequent manual inspection of the extracted texts revealed that images containing private patient information exhibited significantly more text than conventional X-ray images, which typically include simple indications such as laterality or patient position. As a consequence, applying a simple threshold to the amount of text found in the image appeared to be sufficient to filter out images raising privacy issues.

Reports preprocessing

The radiology reports were filtered to only include those that describe the specific images we have previously obtained.

[2] https://github.com/jaidedai/easyocr

Given that the reports are stored in PDF format, the Pdfminer Python module was employed to extract text while simultaneously filtering out headers and footers containing administrative information based on the hospital's specific templates.

Despite these precautions, protected health information (PHI) can still be contained in the text, such as the patient's name and date of birth. Manual elimination of this information from the large volume of documents would be impractical. Consequently, the decision was made to create surrogates documents [19] that keep the useful information from the originals but with fictitious PHIs. DEDUCE [107], a rule-based tool designed for identifying PHIs in Dutch medical texts, was adapted to work with French[3].

To ensure authenticity in the surrogate data, last names and first names were sourced from the Belgian *Direction générale Statistique* (StatBel)[4] and the French *Institut national de la statistique et des études économiques* (INSEE). For health institution names, lists of nursing homes and hospitals from the Belgian *Institut national d'assurance maladie invalidité* (INAMI) were used. A list of all cities in Belgium provided addresses. To further protect privacy, a random shift (between -1000 and +1000 days) was applied to dates, while phone numbers, URLs, and email addresses were simply removed.

Pseudo-labels creation

In order to train a model on a specific task, a labeled dataset is needed. To generate such dataset without manual annotations, we have considered using a large language model (Lama 3 70B [158]) to create pseudo-labels from the textual reports associated with images in the train set. Since Llama 3 has been released as open source, data can be processed locally without sending GDPR protected data to third party. The Llama 3 model was prompted to identify the presence of bone fracture for each radiology report and output the result as 0 or 1, respectively for the absence or presence of bone fracture. To filter out ambiguous results, a simple sampling-and-voting method [89] was used by repeating the process three times with a stochastic nucleus sampling [57]. Only unambiguous reports (i.e. reports with identical labels for the three extractions) were kept. The resulting labels can then be assigned to the images described in each report.

8.2.2 Vision-Language Pretraining

This section describes how the representation of medical images can be adapted to fit the representation of (French) clinical reports.

Figure 8.2 illustrates the VLP and downstream tasks evaluations.

[3]available at `https://github.com/aenglebert/deduced`
[4]https://statbel.fgov.be/fr/themes/population/noms-et-prenoms

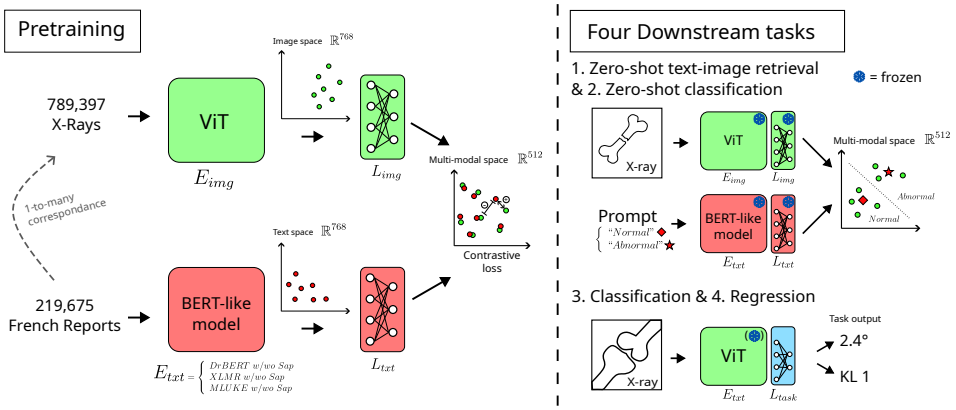


Fig. 8.2 Vision-Language Pretraining (VLP) consists in the alignment of the embeddings for both X-Rays and French Reports. Once pretrained, the encoders can be adapted to different downstream tasks.

Vision-language pretraining (VLP)

In this work, we employed a traditional bi-encoder global contrastive framework, analogous to that proposed by ConVIRT [181]. Previous works such as ConVIRT [181], GLORIA [60], MGCA [163] or PRIOR [27] utilized a (Bio)ClinicalBERT [8] as the text encoder and ResNet50 [51] pretrained on ImageNet [136] as initialization for the image encoder. However, this text encoder is designed for English language and is consequently not ideally suited to the reports of our hospital. Therefore, we explored French and multilingual alternatives to define our text encoder. For the image encoder component, we opted for the more recent ViT [36] model instead of a ResNet. More details about the explored text and image encoders are provided in Section 8.2.2 and Section 8.2.2, respectively. The output CLS (classification token) of the image and text encoders serves as a global representation of the image and text, respectively, and are each linearly projected as a 512-dimensional vector. The objective of the Vision-Language Pretraining is to bring closer the representation of images to the representation of the corresponding report by fine-tuning both image and text encoders. In practice, a CLIP loss, as described by Radford et al [124], is employed to minimize the cosine distance between image and text vectors from the same study, while simultaneously reducing the distance between text and image vectors from different studies. In our work, a study denotes the outcomes of a radiological examination. Hence, it is specific to one patient and to one visit to the hospital, and includes one report and potentially multiple X-ray images.

In preliminary experiments, we evaluated the effectiveness of using one random image from each study compared to utilizing all images from the study and pooling the results (either by averaging the CLS tokens or using attention pool-

ing). It revealed that, the benefits of using multiple images pretraining were not clearly evident, while the complexity of the framework increased. Consequently, we opted to randomly select one image for each study in a batch.

The source code for the vision-language pretraining is available at `https://github.com/aenglebert/multimodal_bone`.

Text encoder and self-alignment pretraining (Sap)

We selected three candidate text encoders:

- XLM-Roberta (XLMR) [29], which is a multi-lingual text encoder based on Roberta [99] and trained using Masked Language Modeling using texts in 100 languages.
- MLUKE [131], a multi-lingual version of LUKE [176] trained with Masked Language Modeling and Masked Entity Prediction on 24 languages.
- Dr BERT [77], a french encoder based on CamemBERT [105] and trained using Masked Language Modeling on a French biomedical corpus.

We also augmented the comparison with the self-alignment pretraining method (Sap) [95]. In this method, a pretraining consists in the alignment of the embeddings of synonyms of concepts from the Unified Medical Language System (UMLS), a compendium that integrates and harmonizes diverse medical terminology and classifications. For XLMR, the original XLMR SapBERT model was used, and we pretrained MLUKE and Dr BERT using the source code of the authors with the parameters described in their paper and UMLS 2020AA, as implemented in the original code.

Image Encoder

The image encoder was initialized from a ViT B16 224x224 pretrained on ImageNet [36]. Multiple resolutions have been explored, 224x224 as the native resolution of the model, and resolutions increased to 336x336 and 448x448. To increase the resolution of the image encoder, the 224x224 image encoder is first pretrained with Vision-Language pretraining as explained in Section 8.2.2 and then modified in two possible ways:

- Interpolation of the position embeddings [36] (named ViT B16 336 and ViT B16 448 in the following). This technique involves the interpolation of the trained position embeddings to enable the input of a greater number of tokens to the vision transformer, thereby accommodating images with higher resolutions, all while preserving their semantic significance.
- Increasing of the patch size using the pseudoinverse resizing methods described in FlexiVit [14]. This approach ensures the retention of the original

number of tokens, while each token covers a larger number of input pixels. Unlike FlexiVit, the resizing is performed once to initialize a ViT B24 336 and a ViT B32 448 models. This method requires less computation in comparison to interpolating the position embeddings.

The vision-language pretraining is then continued with images of increased resolution.

8.2.3 Downstream tasks

This section introduces the different downstream tasks considered to evaluate the performance of the pretraining using external datasets.

The source code for the different downstream tasks is available at `https://github.com/aenglebert/ortho_vlp_eval`.

Trained tasks

For trained task, we adopted an evaluation strategy similar to previous studies [60, 110, 163, 27], with two settings: linear classification on a frozen image encoder, and full fine-tuning. Two tasks are performed depending on the datasets, classification or regression. To assess data efficiency, we compared training using either the entire training set or a smaller part of the training set (from 1 to 10%).

The objective was to evaluate performance in relation to pretraining and not to obtain maximal performance on the downstream tasks per se. For the linear evaluation, a single linear layer was appended to the CLS token of the image encoder to facilitate classification.

Zero-Shot Tasks

For the zero-shot tasks, the vision-language pretrained models are utilized without additional fine-tuning. Two tasks are investigated: zero-shot classification and zero-shot retrieval. In zero-shot classification, a text prompt is classically associated to each class, and images are assigned to the class whose text prompt embedding is the closest (in cosine distance) to the image embedding. In zero-shot retrieval, a fixed number of images with the closest embedding from a class text prompt embedding are retrieved. In practice, this is achieved by leveraging a measurement of distance between the projected CLS (classification token) from both encoders in the multi-modal space, specifically, for the image under consideration and for the reference prompt associated with a given class.

Four prompting strategies were kept for the evaluation:

- **Text binary**: A simple prompt with the name of the target class is used. The negative being a "normal" prompt.
- **Text enumeration**: The class prompt is constructed as a comma separated list of sub-classes of the target class.

- **Latent minimum**: The same sub-classes are encoded as separate prompts by the text encoder. Multiple embedding thus exists that belongs to the same target class.

- **Latent mean**: The same sub-classes are encoded separately by the text encoder. The target class embedding is produced by averaging the sub-classes embedding.

For the classification task, the predicted class is assigned based on the distance in the multi-modal space between the image and a reference prompt (cfr. zero-shot downstream tasks in Figure 8.2).

For the retrieval task, the top k images with the lowest distance from the text query in the multi-modal space are retrieved. Precision is computed across various values of k. This evaluation is closer to zero-shot experiments presented in previous works on the CheXpert 8x200 dataset [181].

8.2.4 Pseudo-label training

To demonstrate that a task-specific model can be trained solely from available clinical data, without resorting to dedicated manual annotation, we have considered the generation of pseudo-labels using a LLM. We focused on the common and clinically significant task of bone fracture detection, which is one of the primary reasons for bone radiography. Images from the same hospital as the one considered during the VLP process (but not used during VLP) have been considered to define a training and test set for this task. The label of the test images have been manually corrected and validated. For the training set. Labels were automatically generated by processing reports using a Llama3 70B model [158] (see Section 8.2.1 for details). Regarding the task-specific model architecture, a single linear classification layer was trained on top of the frozen encoder. The training was conducted using random subsets of the training data of various sizes, repeated 8 times per model for a given number of training images, allowing for the computation of a confidence interval.

8.3 Experimental validation

This section provides an overview of our experimental procedures and results. In Section 8.3.1, we detail the data processing steps, including the creation of the pretraining dataset and pseudo-labeled dataset. In Section 8.3.2, we outline the Vision-Language Pretraining (VLP) process, followed by evaluations on downstream tasks in Section 8.3.3. Section 8.3.4 describes the training on pseudo-labels. Finally, Section 8.3.5 explores the latent space of the models to gain insights into their performance.

Table 8.1 PHI identification metrics

	Count	Precision	Recall	F1-score
Patient names	132.0	0.96	1.00	0.98
Person names	100.0	0.66	0.94	0.78
Locations	52.0	0.98	0.86	0.92
Institutions	23.0	0.76	0.83	0.79
Dates	427.0	0.99	0.98	0.98
Ages	39.0	0.86	0.97	0.91
ID numbers	19.0	0.95	1.00	0.97
Phone numbers	47.0	0.98	0.93	0.96
URL/e-mails	13.0	1.00	1.00	1.00

8.3.1 Data processing

Dataset for vision-language pretraining (VLP)

We obtained approval from the Hospital Ethics Committees (Belgian registration number B403201523492) to conduct this study, which involves the retrospective analysis of data from patients treated in the orthopedics department at Cliniques Universitaires Saint Luc in Brussels.

Our initial step involved identifying relevant patients by filtering the PACS (Picture Archiving and Communication System) to maintain patients who underwent imaging studies prescribed by the Orthopedic surgeons of the hospital, and related to osteoarticular conditions. This process was performed for imaging studies from February 2002 to the 31 of December 2021. To ensure data anonymization, privacy related metadata were systematically removed and a new unique random identifier was assigned to each individual patient and to each study. Following the use of the EasyOCR framework for OCR detection and the removal of images containing more than 35 characters, a manual review of the remaining text extracts did not reveal any Protected Health Information (PHI).

The documents were restricted to radiology reports and aligned with X-ray studies based on their dates (before pseudonymization). In cases where multiple studies and X-ray reports exist for a specific date, we align them in chronological order while disregarding ambiguous instances that necessitate manual examination. After parsing from pdf and pseudonymization using our modified DEDUCE described in Section 8.2.1, 100 reports were randomly selected in the dataset and manually annotated for patient names, person names, locations, institutions, dates, ages, id numbers, phone numbers and url/e-mails. The proposed method was then compared with the annotations, the precision, recall and F1-score were computed for each PHI with results available in Table 8.1.

The process effectively removes sensitive information from pseudonymized documents, with high recall scores for critical data points like patient names (1.0)

and ID numbers (1.0). Recall scores for institutions (0.83) and locations (0.86) are slightly lower due to partial annotation. Precision is generally lower, with person names having the lowest precision (0.66) due to misannotation as locations, and institutions having a precision of 0.76 due to the opposite issue.

Surrogate documents were generated by inserting fictional, yet realistic, protected health information (PHI) into the pseudonymized documents, which were then included in the dataset.

The resulting number of paired studies amounts to 219,675, corresponding to 789,397 individual X-ray images in total.

A set of 4096 studies was excluded from any self-supervised training, and used to test for pseudo-label training in the next section.

Pseudo-labeled dataset for task-specific supervision

This dataset was constructed using images not seen during VLP. For this purpose, 4096 reports with their related X-Rays were left out.

To produce a supervision, the reports were processed using a Llama 3 70B model [158], with GPTQ 4bits quantization [43]. A simple sampling-and-voting method [89] with a self-ensemble of size 3 was used using a nucleus sampling [57] with a top p sampling with probability set to 0.95 and a softmax temperature of 0.8. We then keep only the labeled reports with consistent results for the three runs, resulting in 3802 labels. We then randomly sampled to keep only one labeled report per patient, resulting in 1351 labeled studies.

A test set was reserved, containing 256 labeled studies. The remaining 1095 studies were used for the validation and train set.

8.3.2 Vision-language pretraining on Bone X-Rays and French Reports

The Vision-Language pretraining (VLP) described in Section 8.2.2 was performed on the pretraining dataset described in Section 8.3.1.

A validation set composed of 4096 studies was excluded from the training set and used to adapt the learning rate and stop training on plateau. The training set was then composed of 215,579 studies.

The training was executed on a single NVIDIA A100 80GB GPU using PyTorch 2 with fp16 mixed precision. A batch size of 96 was employed for the initial 224x224 resolution, alongside a LION [26] optimizer quantized in 8bits [33] with a learning rate of $1e^{-5}$, which was reduced by a factor of 2 following a plateau of 3 epochs of validation loss. Additionally, a weight decay of $1e^{-5}$ was applied. Training ceased after 10 epochs without any improvement in validation loss, and the model exhibiting the best validation loss was retained. The training with the 336x336 and 448x448 resolutions was restarted from the training described above with a batch size kept as 96 for the ViT B24 336 and B32 448, but was reduced to 64 for the ViT B16 336 and to 48 for the ViT B16 448 models due to increased memory requirements. A learning rate of $1e^{-6}$ was used for this second training

Table 8.2 Downstream tasks datasets

		Zero-shot		Trained	
		Retrieval	Classif.	Classif.	Regression
Datasets	MURA[127]	✓	✓	✓	-
	FracAtlas[1]	✓	✓	✓	-
	OAI KL[113]	-	-	✓	-
	OAI HKA[113]	-	-	-	✓
	RSNA Bone Age[49]	-	-	-	✓

phase with the same learning rate scheduling and stopping strategy as the first phase.

Data augmentations included a random resized crop to 512x512 followed by a normalization (mean 0.5, std 0.25), an horizontal flip ($p = 0.5$), an affine transformation (random rotation from -20 to $+20°$ and translation from -10 to $+10\%$), brightness and contrast adjustment (random from 0.8 to 1.2 for both), Gaussian blur (random sigma from 0.1 to 3.0), and final resizing to 224x224, 336x336 or 448x448 depending on the image encoder resolution.

Performance evaluations were carried out on the downstream tasks specified in Section 8.3.3.

8.3.3 Evaluation on downstream tasks

It is crucial to note that our objective is not to reach state of the art performance for each dataset but rather to assess the effectiveness of the pretraining process. With this objective in mind, no data augmentation was conducted beyond normalization and resizing to the target resolution. In addition to an evaluation on the validation dataset used in Section 8.3.4, 5 more datasets are used for these evaluations, they are summarized in Table 8.2.

Trained classification

A linear layer has been added to the pre-trained image encoder, to be trained with a binary cross-entropy loss function, weighted by the ratio between positives and negatives in the training set. The initial learning rate was set to 1e-4 and halved after 3 epochs without a decrease in validation loss. Training ceased after 10 epochs without improvement in validation loss, with the best-performing model retained for evaluation on the test set.

Two training scenarios are considered. In the first one, the vision encoder is frozen, and only the linear projection layer is updated during training. In the second one, after having been frozen for 200 steps, the image encoder is unfrozen and fine tuned with a learning rate reduced to 1e-6 to mitigate rapid overfitting of the Vision Transformer (ViT) model.

The results for the trained classification task are shown in Table 8.3 for the lin-

Table 8.3 Classification results obtained with **linear projection of a frozen vision encoder**

	FracAtlas (AUROC)		MURA (AUROC)			OAI KL (AUROC)		Avg Δ
Train set ratio	10%	all	1%	10%	all	10%	all	
A. General initialization methods (ViT B16 224)								
ImageNet Init.	78.6	86.8	70.1	81.1	83.0	58.6	67.8	0
Random Init.	62.6	66.2	56.7	57.4	58.7	51.9	53.8	-15.7
B. English based VLP (ResNet50, results from the paper [181])								
ConVirt	-	-	81.2	85.1	87.6	-	-	-
C. Our French based VLP - Text encoder (+ ViT B16 224)								
Dr BERT	87.3	89.8	81.2	84.4	86.2	68.7	71.7	+6.2
Dr BERT$_{+Sap}$	89.3	90.8	82.2	84.9	86.6	70.5	73.0	+7.3
MLUKE	88.8	91.8	80.3	84.7	86.4	67.1	70.5	+6.2
MLUKE$_{+Sap}$	90.5	92.8	82.2	84.8	86.9	68.7	71.6	+7.4
XLMR	88.7	91.5	80.5	84.3	86.0	68.1	71.5	+6.4
XLMR$_{+Sap}$	88.2	91.0	83.2	85.7	87.0	69.8	72.4	+7.3
D. ViT resolution increases to 336x336 from C.								
B16 XLMR$_{+Sap}$	**92.2**	**94.4**	85.1	**87.1**	88.3	74.3	**76.2**	**+10.2**
B16 mluke	90.2	93.1	84.9	86.4	87.8	71.9	74.5	+9.0
B24 XLMR$_{+Sap}$	89.7	92.3	83.9	86.1	87.5	70.3	72.8	+8.1
B24 mluke	89.4	92.6	83.0	85.5	87.2	69.2	72.1	+7.5
D. ViT resolution increases to 448x448 from C.								
B16 XLMR$_{+Sap}$	91.2	94.2	**85.6**	86.7	**88.4**	**74.5**	**76.2**	+10.1
B16 mluke	89.8	93.0	84.7	86.9	88.3	73.0	75.2	+9.3
B32 XLMR$_{+Sap}$	89.2	92.5	83.6	86.0	87.4	70.1	72.6	+7.9
B32 mluke	89.2	92.2	83.6	85.7	87.3	69.4	72.2	+7.7

Comparison with text encoders presented in Section 8.2.2, with our without text synonym self-alignment (Sap). Different training set ratios are considered to evaluate how the amount of training samples impacts the benefit obtained from VLP pretraining. As MURA is an order of magnitude larger than the other datasets, a ratio of 1% has been applied in addition to the 10% ratio. Avg Δ denotes the average difference to ImageNet model. All vision-language pretrained (VLP) models performs better than ImageNet ViT, with less data. Our models trained with French reports are also on par with ConVirt for MURA dataset at 224x224 resolution. Resolution increase improves the results.

ear projection appended to frozen models and Table 8.4 for fine-tuning case. Different training set ratios are considered to evaluate how the amount of training samples impacts the benefit obtained from VLP pretraining. The results produced

Table 8.4 Classification performance obtained with **fine-tuning of the vision encoder**, followed by a linear projection.

	FracAtlas (AUROC)		MURA (AUROC)			OAI KL (AUROC)		Avg Δ
Train set ratio	10%	all	1%	10%	all	10%	all	
A. General initialization methods (ViT B16 224)								
ImageNet Init.	80.3	88.6	70.3	81.8	87.0	66.1	75.3	0
Random Init.	66.6	69.0	57.5	60.0	64.9	51.8	54.1	-17.9
B. English based VLP (ResNet50, results from the paper [181])								
ConVirt	-	-	81.3	86.5	89.0	-	-	-
C. Our French based VLP - Text encoder (+ ViT B16 224)								
Dr BERT	88.5	91.4	81.2	85.8	89.5	71.6	77.6	+5.2
Dr BERT$_{+Sap}$	89.5	93.2	82.0	85.0	88.9	72.4	78.1	+5.7
MLUKE	88.9	92.4	78.4	84.3	88.8	71.2	78.1	+4.7
MLUKE$_{+Sap}$	89.3	93.2	82.0	86.0	89.6	71.5	77.0	+5.6
XLMR	89.7	93.9	79.9	84.1	89.6	71.9	77.6	+5.3
XLMR$_{+Sap}$	89.9	92.4	82.7	85.9	89.3	71.6	78.1	+5.8
D. ViT resolution increases to 336x336 from C.								
B16 XLMR$_{+Sap}$	**93.3**	**96.1**	84.3	**88.0**	**90.8**	74.7	82.0	**+8.5**
B16 MLUKE	90.5	93.3	82.3	86.7	90.2	74.5	80.2	+6.9
B24 XLMR$_{+Sap}$	90.3	93.3	82.5	86.3	89.5	72.0	78.5	+6.1
B24 MLUKE	89.6	93.5	80.3	85.6	89.5	71.6	77.5	+5.4
D. ViT resolution increases to 448x448 from C.								
B16 XLMR$_{+Sap}$	91.3	95.3	**84.4**	87.6	90.5	**75.3**	**82.3**	+8.2
B16 mluke	90.6	94.1	82.2	86.8	90.5	74.7	80.2	+7.1
B32 XLMR$_{+Sap}$	91.2	93.0	82.7	85.9	89.4	71.8	78.7	+6.2
B32 mluke	89.3	93.3	80.8	85.6	89.3	71.6	77.7	+5.4

Comparison with the text encoders presented in Section 8.2.2, with our without text synonym self-alignment (Sap). Different training set ratios are considered to evaluate how the amount of training samples impacts the benefit obtained from VLP pretraining. As MURA is an order of magnitude larger than the other datasets, a ratio of 1% has been applied in addition to the 10% ratio. Avg Δ denotes the average difference to ImageNet model. All VLP models performs better than ImageNet ViT with less data. Our models trained with French reports are also on part with ConVirt for MURA dataset at 224x224 resolution. Resolution increase improves the results.

by our models always show superior performances in comparison to models initialized from ImageNet or from scratch, both in linear evaluation and fine-tuning. At a resolution of 224x224, our results are comparable or better than results from

the ConVirt paper, where an English-based dataset of bone X-Rays was used. The comparison is limited since, although the settings are similar, they are not identical. Notably, the architecture of the image encoder and the pretraining datasets are different.

The Sap process tends to improve the classification results of the three encoders. Among the different text encoders used as initialization, all are performing similarly well on average when coupled with Sap for the 224x224 resolution comparison, with small variations on the individual datasets.

The increase of resolution during the vision-language pretraining from 224x224 to 336x336 has a positive impact on the performances. Keeping the patch size of 16x16 while increasing the resolution by interpolating the position embedding requires more computations but shows better performances in comparison to increasing the patch size to 24x24, allowing to keep a complexity similar to a ViT B16 on 224x224 images.

Further increasing the resolution to 448x448 does not improve significantly the results, and is often detrimental in comparison to a 336x336 resolution. A possible explanation for this phenomenon could be related to the reduction of the batch size during vision-language pretraining for the ViT B16 448 models.

Regression

In the regression context, a single linear layer is appended to the CLS token of the image encoder, with additional scale and bias parameters initialized using the mean and standard deviation of the training set for each dataset. To train the resulting model, a smooth L1 loss function [61] is used, and the mean absolute deviation (MAD) serves as the test prediction evaluation metric.

In practice, as for the classification case, two training scenarios are considered. The first one keeps the encoder frozen, while the second fine tunes it.

For the RSNA Pediatric Bone Age dataset, the linear layer takes as an additional input the sex of the patient. We compare the effectiveness of training with either the entire training set or 10% of both the RSNA Pediatric Bone Age and OAI HKA Angles datasets. As with the classification task, data augmentation is minimized, involving only resizing to the target resolution and normalization with a mean of 0.5 and a standard deviation of 0.25. For the OAI HKA measurement exclusively, resizing was conducted while preserving the aspect ratio by padding the image. This approach aimed to prevent distortion of the angles within the image.

The results of the regression task can be seen in Table 8.5.

For the RNSA bone age estimation, our pretraining is beneficial in all finetuning cases in comparison to an ImageNet or random initialization, and with all or 10% of the training set. The increase of resolution is also beneficial, but only when we scale the number of token by interpolating the position embeddings and keeping the 16x16 patch size. Increasing the resolution by changing the patch size

Table 8.5 Comparison of VLP models to ImageNet for regression tasks.

dataset	OAI HKA (Mean error in °)		RSNA bone age (Mean error in months)			
eval	lin.	ft	lin	lin	ft	ft
Train ratio	all	all	10%	all	10%	all
A. General initialization methods (ViT B16 224)						
ImageNet Init.	**2.04**	1.68	16.20	14.99	15.56	12.08
Random Init.	2.66	2.61	32.08	31.35	32.37	22.71
C. Our French based VLP - Text encoder (+ ViT B16 224)						
Dr BERT	2.33	1.64	16.43	15.37	14.91	11.45
Dr BERT$_{+Sap}$	2.24	1.63	15.85	14.54	15.16	11.39
MLUKE	2.37	1.58	16.56	15.51	14.99	11.34
MLUKE$_{+Sap}$	2.26	1.69	15.86	14.71	15.00	11.55
XLMR	2.26	1.62	16.34	15.29	15.32	11.59
XLMR$_{+Sap}$	2.40	1.56	15.50	14.46	14.78	11.41
D. ViT resolution increases to 336x336 from XLMR$_{+Sap}$ in C.						
ViT B16	2.27	1.56	14.73	13.77	14.23	10.88
ViT B24	2.31	1.60	15.51	14.65	14.98	11.15
D. ViT resolution increases to 448x448 from XLMR$_{+Sap}$ in C.						
ViT B16	2.31	1.54	**14.43**	**13.41**	**13.51**	**10.15**
ViT B32	2.36	1.58	15.41	14.49	15.04	11.26

Two training scenarios are envisioned: linear layer training with frozen encoder (lin.), and entire network fine-tuning (ft). Our VLP models do not perform better than ImageNet for angle measurement, probably due to scale invariant pretraining. The Bone age estimation resulting from the linear projection is improved by the VLP pretraining when the full model is fine-tuned, but not with fixed encoder. Resolution increases performances for most scenarios.

does not yield better results.

Concerning the OAI HKA measurement, the error rate is higher than the basic ImageNet model in linear evaluation, and only a full fine-tuning allows to produce similar performances on 224x224 images. The increase of resolution has no effect for the linear evaluation, and slightly increases the performances for the fine-tuned models. These results are not surprising given that the aspect ratio of the images is modified during the data augmentation of our vision-language pretraining. This makes our models invariant to this kind of deformations but also impedes the ability to measure angles.

Table 8.6 Zero-Shot classification on MURA with different image and text encoders.

	Text Binary	Text Enumeration	Latent Minimum	Latent Mean
A. VLP pretraining - Text encoder (+ ViT B16 224)				
Dr BERT	67.4	66.8	64.8	65.6
Dr BERT$_{+Sap}$	69.2	76.9	68.8	76.8
MLUKE	73.9	74.7	72.3	74.6
MLUKE$_{+Sap}$	68.0	72.5	67.0	68.9
XLMR	60.3	72.6	69.4	68.1
XLMR$_{+Sap}$	65.4	78.4	73.1	72.9
B. VLP pretraining - 336x336 Image encoder & Text encoder				
B16 & XLMR$_{+Sap}$	64.0	**79.2**	74.4	74.2
B24 & XLMR$_{+Sap}$	64.0	78.1	72.7	73.0
B16 & MLUKE	75.9	75.9	74.7	77.4
B24 & MLUKE	74.4	74.3	73.7	75.0
C. VLP pretraining - 448x448 Image encoder & Text encoder				
B16 & XLMR$_{+Sap}$	62.7	**79.2**	73.4	74.0
B32 & XLMR$_{+Sap}$	65.5	78.5	72.2	72.4
B16 & MLUKE	74.6	74.5	73.5	77.0
B32 & MLUKE	73.9	74.1	73.5	75.0

Sap denotes the use of textual synonyms self-alignment. Four strategies have been considered to localize normal/abnormal classes in the embedding space, from text prompts. 'Text binary' simply uses the name of the classes (normal/abnormal) as text prompts. 'Text enumeration' uses a prompt consisting of a list of pathologies (see text for details) separated by commas for the abnormal class. 'Latent minimum' uses individual embeddings for each sub-classes. Eventually, 'latent mean' averages, in the embedding space, the prompts derived from each pathology associated to the abnormal class. Best overall in **Bold**, best for each strategy in underline. MLUKE performs best with latent mean strategy while XLMR + Sap is best with text enumeration. Performances increase with the 336x336 resolution, the 448x448 resolution does not improve.

Zero-Shot Classification

This task was explored using MURA and FracAtlas datasets. We chose not to pursue exploration on the OAI KL dataset for this task. This is because, unlike distinctive classes such as the presence or absence of bone fractures, the semi-quantitative KL scale poses a bigger challenges in being accurately reflected through text prompts.

Regarding the MURA dataset, the class prediction associated to a study is generated by averaging the results for all images within a given study. This dataset

Table 8.7 Zero-Shot classification on FracAtlas with different image and text encoders.

	Text Binary	Text Enumeration	Latent Minimum	Latent Mean
A. VLP pretraining - Text encoder (+ ViT B16 224)				
Dr BERT	56.3	49.3	51.6	47.2
Dr BERT$_{+Sap}$	72.7	56.1	47.8	56.6
MLUKE	72.8	62.0	67.3	66.9
MLUKE$_{+Sap}$	61.1	55.2	41.3	52.4
XLMR	70.7	62.5	58.3	60.4
XLMR$_{+Sap}$	61.3	57.0	57.7	59.0
B. VLP pretraining - 336x336 Image encoder & Text encoder				
B16 & XLMR$_{+Sap}$	71.0	66.2	65.8	68.6
B24 & XLMR$_{+Sap}$	62.6	59.1	60.8	61.1
B16 & MLUKE	**77.3**	56.6	63.9	70.0
B24 & MLUKE	72.4	58.2	66.3	67.4
C. VLP pretraining - 448x448 Image encoder & Text encoder				
B16 & XLMR$_{+Sap}$	69.2	63.6	64.5	66.5
B32 & XLMR$_{+Sap}$	61.0	57.8	57.5	58.4
B16 & MLUKE	73.5	57.9	66.7	69.9
B32 & MLUKE	71.5	58.1	63.6	64.3

Sap denotes the use of textual synonyms self-alignment. Four strategies have been considered to localize normal/abnormal classes in the embedding space, from text prompts. 'Text binary' simply uses the name of the classes as text prompts. 'Text enumeration' uses a prompt consisting of a list of pathologies separated by commas for the abnormal class. 'Latent minimum' uses individual embeddings for each sub-classes. Eventually, 'latent mean' averages, in the embedding space, the prompts derived from each pathology associated to the abnormal class. Best overall in **Bold**, best for each strategy in underline. Similarly to Table 8.6, the couples MLUKE with binary strategy and XLMR + Sap with text enumeration are the best performers. The 336x336 resolution shows superior performance with no benefits to further increase to 448x448.

poses a challenge for zero-shot binary classification between normal and abnormal due to contextual variability in defining abnormality. For instance, the presence of osteoarthritis in a radiograph taken for an elderly individual following trauma to rule out a bone fracture or dislocation could be noted, yet the overall radiograph might still be treated as normal within the trauma context.

Therefore, we explored several strategies, as described in Section 8.2.3, to localize the normal and abnormal classes in the shared embedding space.

The methodology described in Section 8.2.3 was evaluated and reported on

the test set of both datasets. Evaluation of this task is conducted using the Area under the ROC curve (AUROC).

Results are presented in Table 8.6 and Table 8.7 for MURA and FracAtlas, respectively.

These results exhibit significant variations in performance depending on the prompting strategy and on the chosen models. This is not surprising since we employ a straightforward distance measurement between image embeddings and anchor points that differ substantially from conventional radiology reports.

Depending on the models, a text enumerating various pathologies considered as abnormal or the average of the embeddings of these pathologies performs best on MURA, while a simpler text query performs better for the binary bone fracture classification of FracAtlas. For both datasets, encoders based on MLUKE and XLMR + Sap performs best. The overall effect of the Sap pretraining is not clearly demonstrated as beneficial in this experiment. For MLUKE, this degrades systematically the performances.

Increasing the resolution from 224x224 to 336x336 improves results for both setups explored, while further increase to 448x448 does not improve results.

Our analysis reveals that the models exhibit significant sensitivity to the prompt employed, raising the possibility that alternative prompting strategies may yield improved outcomes for one model or another. Consequently, it remains challenging to definitively determine which encoder is optimal on this task.

Zero-shot Text-Image Retrieval

In this experiment, the MURA and FracAtlas datasets were also used. Instead of evaluating only using the test set, as decided for the classifications tasks (to allow comparison with previous works from ConVirt), we split each dataset in 5 folds and performed the retrieval task on each fold. The retrieval precision was computed on the top k retrieved images with k=10 and 50 and without any training on theses datasets.

Results are represented in Table 8.8 for both MURA and FracAtlas.

For MURA, the differences between models are smaller in comparison to zero-shot classification. The text synonyms self-alignment pretraining (Sap) increases performances of the Dr BERT model on FracActlas, while degrading performances of the other models. On MURA however, the difference between models with or without Sap is smaller and not significantly different.

8.3.4 Pseudo-label training

The pseudo-labeled dataset described in Section 8.3.1 was used in this section. The dataset is composed of 1351 studies, of which 256 are reserved as a test set. The remaining 1095 studies (accounting for 3657 X-Rays) are randomly sampled as a validation set using 10% of the studies, and a train set using a subset with a

Table 8.8 Retrieval Results on FracAtlas and MURA Datasets.

	Dr BERT	+Sap	XLMR	+Sap	MLUKE	+Sap
FracAtlas						
Negation P@10	$64_{\pm 12.9}$	$90_{\pm 6.1}$	$85_{\pm 6.1}$	$68_{\pm 10.4}$	$95_{\pm 3.5}$	$88_{\pm 7.6}$
Negation P@50	$56.0_{\pm 5.4}$	$82.8_{\pm 5.8}$	$77.2_{\pm 3.3}$	$65.2_{\pm 4.6}$	$78.4_{\pm 1.9}$	$71.6_{\pm 2.9}$
Text enum. P@10	$51_{\pm 7.4}$	$75_{\pm 15.0}$	$89_{\pm 12.4}$	$60_{\pm 6.1}$	$77_{\pm 6.7}$	$75_{\pm 8.7}$
Text enum. P@50	$48.6_{\pm 4.4}$	$64.0_{\pm 4.8}$	$68.4_{\pm 6.6}$	$60.2_{\pm 2.4}$	$64.8_{\pm 2.9}$	$63.4_{\pm 4.9}$
Lat. mean P@10	$52_{\pm 4.5}$	$63_{\pm 9.7}$	$76_{\pm 8.2}$	$69_{\pm 8.9}$	$80_{\pm 6.1}$	$74_{\pm 6.5}$
Lat. mean P@50	$51.2_{\pm 4.3}$	$62.6_{\pm 5.9}$	$64.2_{\pm 4.9}$	$66.8_{\pm 5.4}$	$72.4_{\pm 2.1}$	$59.2_{\pm 4.9}$
MURA						
Negation P@10	$86_{\pm 8.2}$	$86_{\pm 4.2}$	$85_{\pm 3.5}$	$84_{\pm 6.5}$	$89_{\pm 4.2}$	$91_{\pm 5.5}$
Negation P@50	$86.4_{\pm 4.0}$	$86.2_{\pm 2.3}$	$87.0_{\pm 3.2}$	$84.6_{\pm 3.0}$	$87.8_{\pm 2.3}$	$85.8_{\pm 4.0}$
Text enum. P@10	$93_{\pm 5.7}$	$89_{\pm 4.2}$	$80_{\pm 6.1}$	$84_{\pm 2.2}$	$89_{\pm 5.5}$	$84_{\pm 9.6}$
Text enum. P@50	$86.6_{\pm 2.9}$	$89.8_{\pm 4.9}$	$83.2_{\pm 1.9}$	$87.2_{\pm 1.1}$	$87.2_{\pm 2.6}$	$85.8_{\pm 1.8}$
Lat. mean P@10	$85_{\pm 5.0}$	$87_{\pm 8.4}$	$79_{\pm 4.2}$	$81_{\pm 11.4}$	$90_{\pm 3.5}$	$78_{\pm 4.5}$
Lat. mean P@50	$84.4_{\pm 2.9}$	$90.4_{\pm 3.4}$	$81.2_{\pm 2.7}$	$84.8_{\pm 2.6}$	$91.2_{\pm 1.3}$	$80.8_{\pm 3.5}$

P@X denotes the retrieval precision among the top X samples. (± Standard deviation, computed using a 5-fold). For the FracAtlas dataset, more variability is seen, with performances diverging significantly among models. XLMR exhibits superior performance compared to XLMR+Sap, while Dr BERT+Sap demonstrates good performance. Consistently with previous experiments, MLUKE remains among the top performers, while Dr BERT alone consistently ranks at the bottom. For the MURA dataset, all models perform well on the retrieval task, showing no significant differences in performance.

ratio of 1.0, 0.5, 0.25, 0.125, 0.0625 and 0.03 of the remaining data. The effective train set size is thus comprised between 30 and 986 studies ($\approx$ 99 to 3291 X-Rays).

A linear layer has been added to the frozen pre-trained image encoder, to be trained with a binary cross-entropy loss function, weighted by the ratio between positives and negatives in the training set. The initial learning rate was set to 1e-4 and halved after 3 epochs without a decrease in validation loss. Training ceased after 10 epochs without improvement in validation loss, with the best-performing model retained for evaluation on the test set.

The results are represented in Figure 8.3.

The VLP models achieved significantly better performance than a model trained on ImageNet, even when using one order of magnitude fewer images during training.

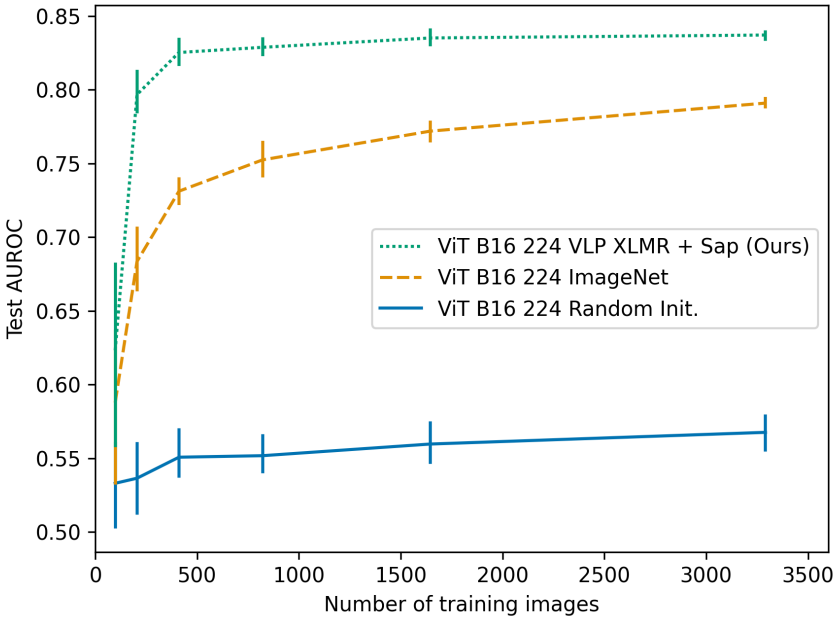


Fig. 8.3 Classification AUROC achieved when training a **linear projection of a frozen vision encoder** on varying numbers of images obtained from the same hospital as the dataset used for vision language pretraining (VLP). The vertical bars represent the 95% confidence intervals, calculated from 8 training sessions, with different seeds used for sampling these images. The VLP models achieved better performance than a model trained on ImageNet, even when using an order of magnitude fewer images during training. To enhance clarity, only two of our VLP models are displayed in the plot.

8.3.5 Latent space exploration

In this section, we will explore in more details the native latent space organization of our models. The goal is to enhance our understanding of the results obtained in Section 8.3.3. Particularly to get an insight for the zero-shot results variability.

Using 200 images of each anatomical region of the MURA dataset and their associated embeddings produced by the self-supervised models, we employed a t-SNE [104] algorithm to explore visually intrinsic data distribution in the 512-dimensional space. The result can be seen in Figure 8.4 for ImageNet and VLP pretrained with XLMR + Sap. Notably, while a ViT B16 model trained on ImageNet has already begun to incompletely cluster the anatomical locations, a ViT models pretrained from Section 8.2.2 on bone X-Rays and French reports exhibits an improved ability to differentiate the different anatomical locations, with finger

and hand being unsurprisingly the two locations with the higher overlap.

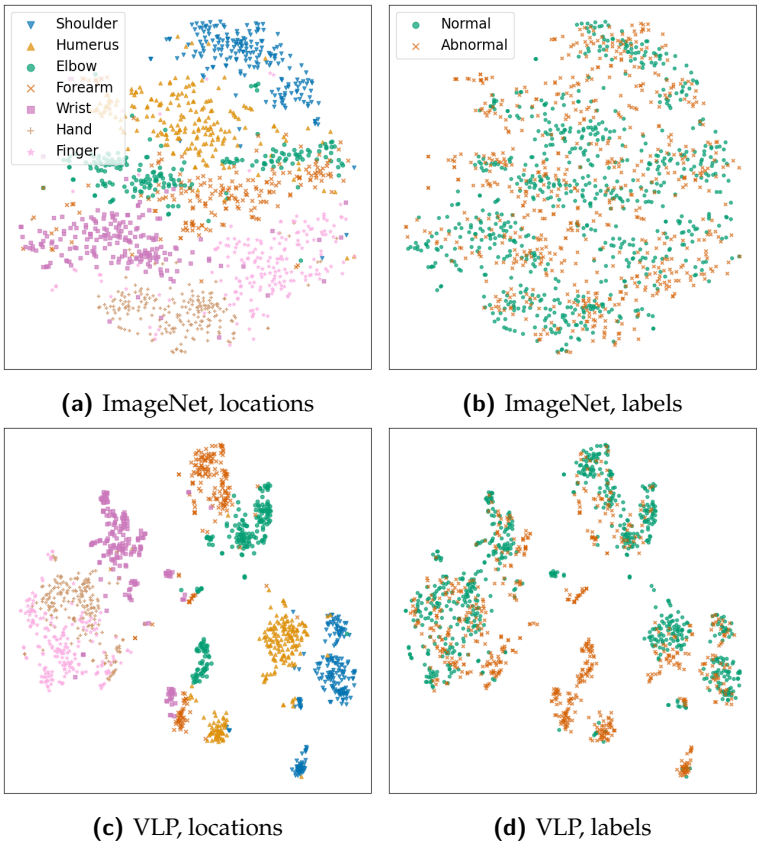
Two observations can be drawn from this analysis. Firstly, VLP models naturally form dense and well-separated clusters by anatomical region, unlike ImageNet. Secondly, for the VLP models, each anatomical region appears to be made up of large well-distinct sub-clusters and other smaller sub-clusters grouped together (as observed in the first row of the figure). These grouped smaller subclusters, corresponding to several anatomical regions, are drawn from pathological samples (as observed in the second row of the figure). By manually exploring the images belonging to different subclusters, we observe that osteosynthesis radiographs containing metal plates are more often grouped together and aggregated on separated groups on the t-SNE, in comparison to smaller or not displaced bone fractures that tend to remain closer to normal images of the same anatomical location in the t-SNE plot.

8.4 Discussion

In this chapter, we demonstrated the possibility to leverage raw radiographic images and associated french reports from a single hospital to train deep learning backbone without manual annotation. The whole pipeline to prepare the text and image data is made available with a special emphasis on the anonymization process adapted to French language. We examined various text encoders initializations and found that a multilingual text encoder outperforms those limited to biomedical French-only texts. Pretraining the text encoder through self-alignment using UMLS ontology has also improved performance on supervised downstream tasks. We generated pseudo-labels for bone fracture detection without relying on externally annotated datasets, allowing training of a task-specific model without the need for manual annotation. While the ultimate goal would not be to deploy the model without any manual data verification, pushing the boundaries of what can be achieved without manual annotation significantly reduces the overall need for manual annotation, thereby streamlining the training process and accelerating clinical implementation. In comparison to ImageNet, we observed a notable performance enhancement across different classification tasks, both when only training the final linear layer or when fine-tuning the whole network on external datasets. There was also an improvement in regression tasks with fine-tuning of the model. While results in zero-shot settings are promising, they exhibit more variability, particularly in zero-shot classification, revealing the need for a minimal amount of annotations to solve tasks properly.

Increasing the resolution from 224x224 to 336x336 yielded better results, particularly when position embedding interpolation is used. However, further resolution increase to 448x448 showed limited or no additional gain, considering the increased computational complexity.

Alternatively, a new direction for research could be to redefine the task as a



(a) ImageNet, locations **(b)** ImageNet, labels

(c) VLP, locations **(d)** VLP, labels

Fig. 8.4 t-SNE visualizations of the embeddings of MURA images with ImageNet ViT (no VLP) and VLP ViT with XLMR + Sap. Best viewed in color. The VLP models show a better clustering in comparison to the ImageNet model for both anatomical locations and labels. No training on MURA was conducted for any of the models. Clusters tends to form predominantly based on the anatomical location. However, within a specific anatomical site, various clusters frequently emerge, notably clusters with osteosynthesis material (visualized in (d) as clusters composed of only abnormal images).

report generation and visual question answering problem, shifting away from the current approach of using contrastive loss in the embedding space. The next chapter provide an initial exploration of this new direction.

9

Vision-language model

9.1 Introduction

In the previous chapter, we investigated the use of self-supervised vision-language pretraining (VLP) to improve the analysis of bone X-rays using French reports. Our results showed that models pretrained on paired image-text data from the Cliniques Universitaires Saint Luc outperformed those initialized on ImageNet in various downstream tasks. However, these models required fine-tuning for each specific task to achieve optimal results.

Building on this foundation, we now explore the application of recent advancements in vision-language models to automate the generation of standardized medical reports and enable visual question answering (VQA). Our goal is to prepare the ground for the development a vision-language model capable of interpreting bone radiographs and answering questions about specific image elements. Unlike previous chapters, the research presented here is still in its early stages, and the goal of this chapter is mainly to present our preliminary findings and initial explorations that may set the stage for future work.

9.2 Methodology

This section outlines the steps taken to leverage vision-language models for generating standardized medical reports and performing visual question answering. We first describe the dataset preparation and then detail the model fine-tuning process.

9.2.1 Dataset Preparation

We utilized the same dataset described in the previous chapter, which originally comprised 219,675 studies from Cliniques Universitaires Saint Luc in Brussels. Each study includes one or multiple X-ray images paired with a corresponding French report. However, for this chapter, we applied additional filtering to remove a few reports with insufficient text, resulting in a final dataset of 219,650 studies. We focus on two main tasks using this dataset: generating standardized reports and visual question answering. The whole process is summarised in Figure 9.1.

Generating Standardized Reports

We utilized the Llama 3 70B model [158] (4-bit AWQ quantized [93]) to standardize the French reports. The original pseudonymized reports exhibited varying styles, ranging from verbose sentences to concise keyword-based entries, and often included contextual information such as comparisons to previous reports or clinical context that cannot be inferred from the images. It has been shown that removing references to prior reports in chest radiology reports improves performance and reduces hallucinations of models trained on these filtered reports [129]. To convert these reports into a structured format, we prompted the frozen language model to act as a radiologist who only has access to the images. Because the model cannot actually process images directly, we provided the original report as input, framing it as a report from a colleague that should not be referenced as a source of information. This approach allowed the model to generate standardized reports while omitting irrelevant information.

Visual Question Answering (VQA)

Building on the LLaVA framework [97, 96], we leveraged Llama 3 to generate a set of questions and answers based on the reports from the previous step. Although this process has been adapted for the medical field in other contexts, such as general images from biomedical papers [85] or chest x-rays [84], it had not been applied to bone x-rays or the French language until now. We instructed the Llama 3 model to simulate a scenario where an emergency doctor, without access to the report, asks questions to an experienced radiologist who responds based on the images. 5 sample dialogs have been given to the model in the system prompt to guide generation towards the desired style. The generated Q&A set was then further processed using Llama 3 to be formatted into a standardized JSON file.

9.2.2 Model Training and Fine-Tuning

We employed the Idefics 2 model [81] as the backbone for our vision-language tasks. The model was fine-tuned using QLoRA [34] on the standardized report and VQA datasets.

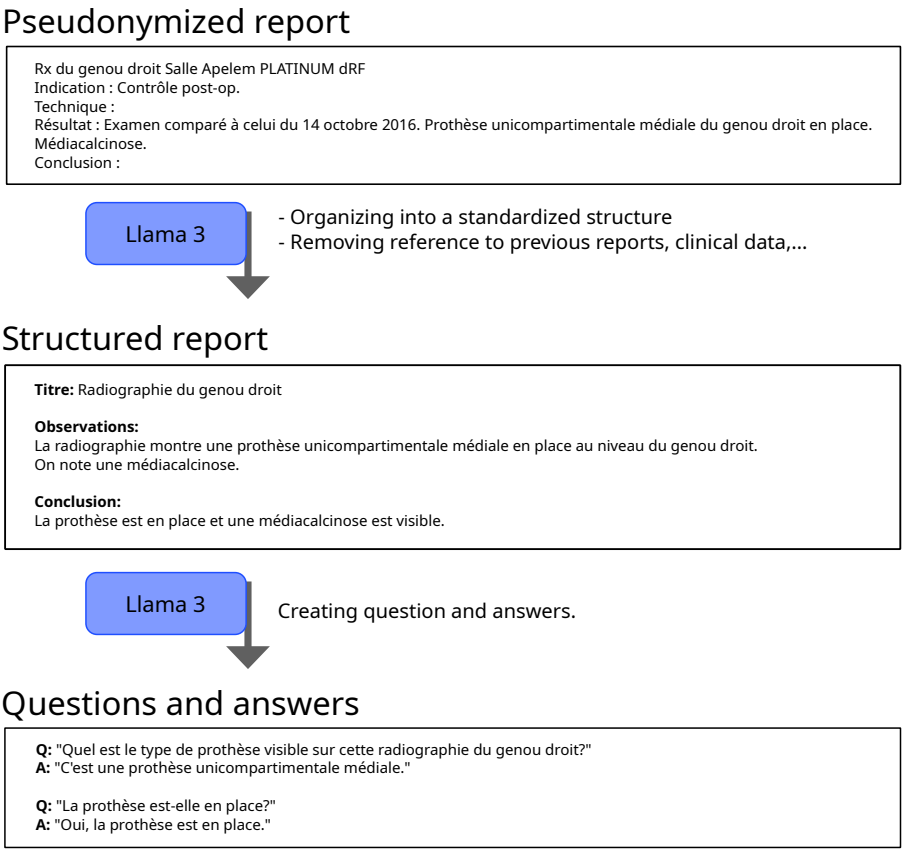


Fig. 9.1 Process pipeline involving to convert pseudonymized original reports into structured standardized reports and to generate a visual question answering (VQA) set. The model is first used to create standardized reports, which are then utilized to formulate Q&A pairs, further processed into a standardized JSON format (not shown in the figure).

Quantized Low-Rank Adaptation (QLoRA)

QLoRA is a technique that adapts large language models to specific tasks by applying quantization to the frozen model weight, while using low-rank adaptations (LoRA) to injects trainable rank decomposition matrices into the model [59, 34]. This approach significantly reduces the computational requirements, allowing the Idefics 2 model to be fine-tuned on our dataset using a single Nvidia A100.

Training Procedure

A single training was performed on both the reports and VQA. At each training step, we randomly selected either report generation or VQA for each study. To initiate report generation, we provided the model with an instruction to create a

report, whereas for VQA, we simply input a question to the model.

For QLoRA, a rank of 128 and an alpha of 128 were used, with a dropout of 0.1 while the model weight were quantized in 4bits normalized float [34]. The fine-tuning was performed using an AdamW optimizer [101] quantized in 8bits [33] and a learning rate of $1e^{-6}$, with a batch size of 32 studies (each study composed of up to 6 radiographs with a max resolution of 512x512) during 1 epoch.

9.3 Preliminary assessments

Our initial assessments of the model's performance are qualitative in nature. We generated a dozen reports on radiographs that were not seen during training. The model appears to understand the two tasks of report generation and visual question answering (VQA), and it correctly generates reports with the appropriate structure. Notably, the model is able to identify the title of the study, which is typically the anatomical location, whereas the baseline model (Idefics 2) can only identify the images as radiographs but fail to identify the content. However, our model tends to produce hallucinations.

The model's performance in generating reports is stronger for more common diagnoses, such as wrist fractures, as shown in Figure 9.2 where it accurately identifies the fracture and displacement. However, it tends to misattribute common characteristics to less common conditions, such as mistaking a unicompartmental knee arthroplasty for a total arthroplasty, as seen in Figure 9.3. In some cases, the model's localization of fractures is also incorrect, as in Figure 9.4, where it incorrectly identifies the 5th metacarpal as the site of the fracture instead of the proximal phalanx of the index finger.

In visual question answering, the model exhibits a notable bias towards the wording of the question. Specifically, when asked if a fracture is present, the model tends to affirm the presence of a fracture, even if none exists. Conversely, when asked about the presence of a bone lesion, the model often responds that there is no bone lesion, even if a fracture is actually present. This suggests that the model's answers are influenced by the specific phrasing of the question, rather than solely by the visual evidence. This is shown in Figure 9.2, Figure 9.3, and Figure 9.4.

9.4 Discussion

As highlighted previously, this chapter presents preliminary explorations and needs further refinements before producing usable vision-language models. A possible approach to consider could be to manually review the produced datasets to identify and address potential issue introduced in the reports during preprocessing which could screw the model training. This will ensure the robustness and reliability of our dataset. For instance, when discussing the absence of a

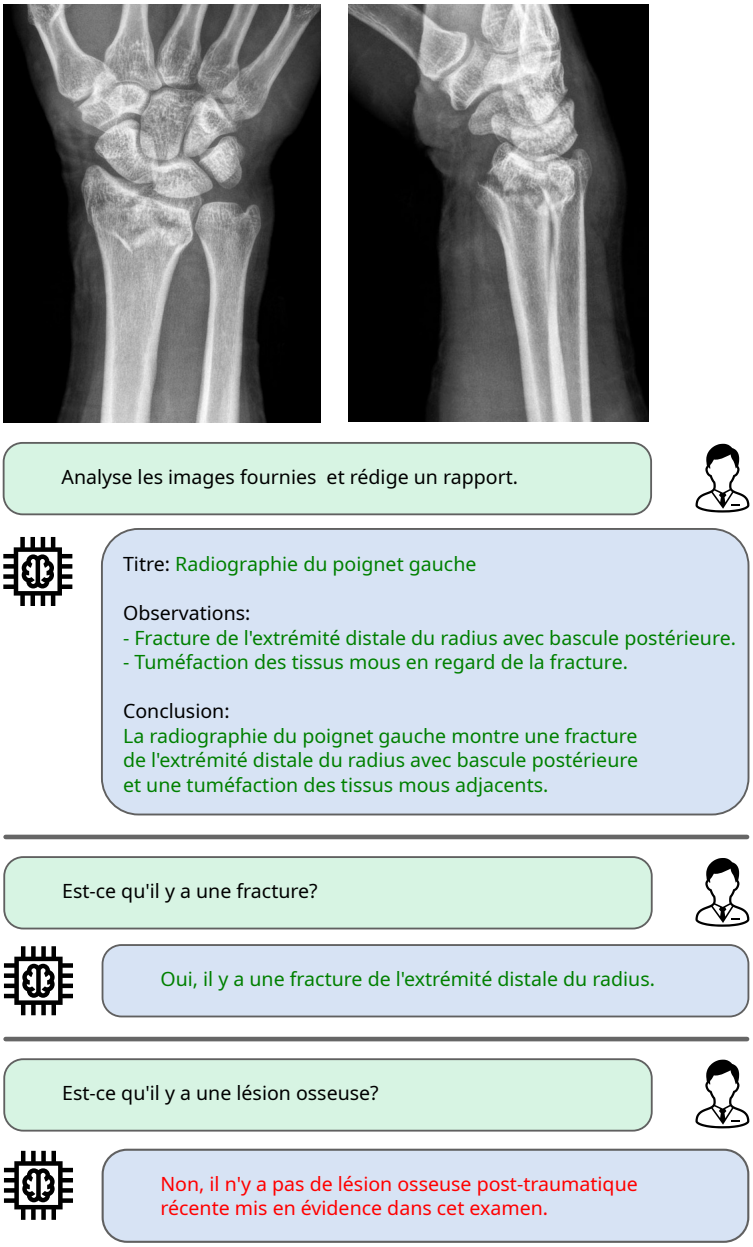


Fig. 9.2 This figure demonstrates preliminary results from our vision-language model. The model processes a left wrist radiograph. The primary outcomes reveal that the model accurately identifies a fracture of the distal radius and associated tissue swelling. However, the model also produces a notable hallucination, incorrectly stating the absence of any bone lesion post-trauma when asked. Note: Images sourced from `https://radiopaedia.org/` for illustrative purposes, to maintain patient confidentiality and GDPR compliance on our data.

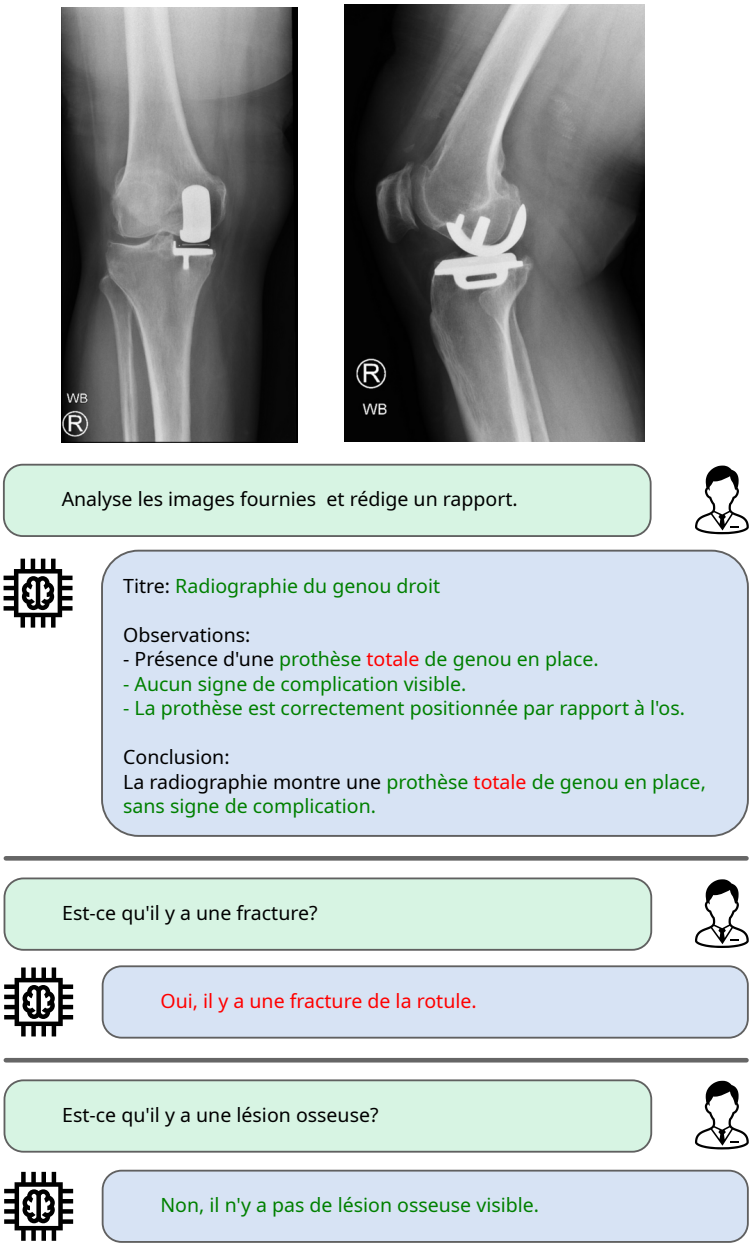


Fig. 9.3 This figure demonstrates preliminary results from our vision-language model. The model processes a knee arthroplasty radiograph. The primary outcomes show that the model incorrectly identifies a total knee arthroplasty instead of a unicompartmental knee arthroplasty. Additionally, the model incorrectly detects a fracture when directly asked. Note: Images sourced from `https://radiopaedia.org/` for illustrative purposes, to maintain patient confidentiality and GDPR compliance on our data.

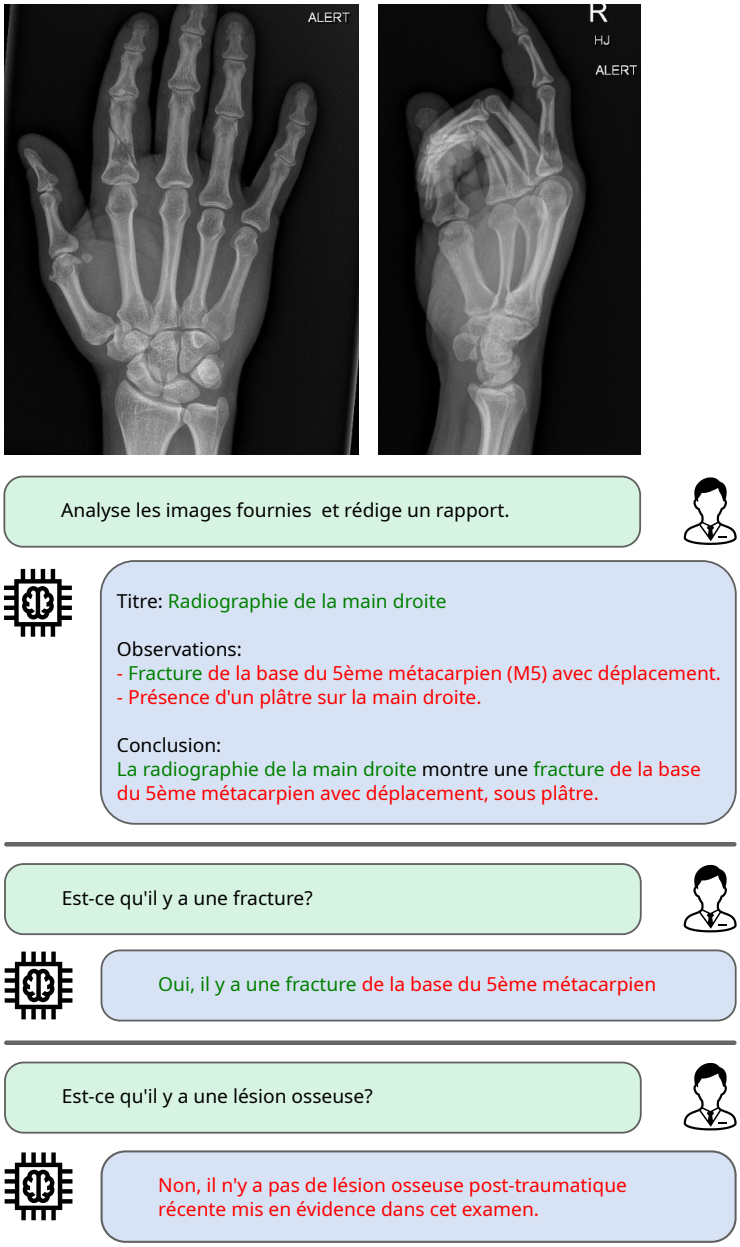


Fig. 9.4 This figure demonstrates preliminary results from our vision-language model. The model processes a right hand radiograph. The primary outcomes reveal an erroneous detection of a fracture in the 5th metacarpal instead of the proximal phalanx of the index finger. Additionally, the model incorrectly states the absence of any bone lesion post-trauma when asked. Note: Images sourced from `https://radiopaedia.org/` for illustrative purposes, to maintain patient confidentiality and GDPR compliance on our data.

fracture, it is more conventional to use the term "bone lesion" (classically used like “Absence de lésion osseuse post-traumatique" in French), whereas the term “fracture" is typically more used when a fracture is present. As the Llama 3 model tends to reuse terms from the original report to generate questions, this introduces bias into the generated questions which translates into a bias in the trained model.

Given the recent release of the Llama 3.1 model, which demonstrates superior performance, it could also be worthwhile to reprocess our dataset using this updated model. Our initial trials showed significant improvements when transitioning from Mixtral 8x7B [68] to newer Llama 3, emphasize the importance of the performance of the model used for data preprocessing. While GPT-4 is not viable due to GDPR constraints, exploring open-source models like Llama 3.1 remains a promising avenue.

A critical next step is the creation of a comprehensive benchmark for our dataset. This benchmark will enable us to quantify the model’s performance and provide a clearer picture of its capabilities and areas needing improvement.

Additionally, we need to iterate over the training process. While QLoRA has proven effective, it may not be sufficient on its own. Alternative strategies, such as using different training strategies for different parts of the network could be explored (e.g. promote a better representation of the image encoder by keeping the text model frozen during all or part of the training session,...).

Lastly, Idefics 2, based on the outdated Llama 2, might not be the optimal choice for our tasks. Open-source vision-language models are still catching up, but future releases based on more advanced models could potentially change the game.

In conclusion, while our preliminary findings are promising, there is substantial work ahead to refine our approach and fully realize the potential of vision-language models in medical imaging tasks.

10

Conclusion

10.1 Main findings

This thesis set out to explore the transformative potential of artificial intelligence (AI) in healthcare, with a specific focus on explainability and vision-language self-supervision in the context of bone radiography. Throughout the chapters, we have traversed from the theoretical underpinnings of deep learning and explainable AI (XAI) to the practical implementation and evaluation of novel vision-language self-supervised methodologies that push the boundaries of current AI applications in medicine.

10.1.1 Addressing Research Questions

Explainability

Research Question #1

How can we develop explainability methods that provide insights into the decision-making processes of artificial intelligence models ?

In Chapter 4 we introduced Poly-CAM, a novel method for generating high-resolution saliency maps for Convolutional Neural Networks (CNNs) without relying on gradient backpropagation. Our experiments demonstrated that the method excels in faithfulness insertion-deletion metrics and outperforms existing techniques in terms of visualization precision. Additionally, the Transformer Input Sampling (TIS) method was developed in Chapter 5, offering an alternative to attention- and gradient-based accounts of visual transformers. The versatility

of TIS makes it a promising approach for future multimodal applications with transformers.

Research Question #2

Can explainable methods uncover and help to mitigate biases in artificial intelligence model training ?

We applied the Poly-CAM method to bone radiographs in Chapter 6, discovering biases in the model's predictions, specifically with cast images. We demonstrated that these biases can be mitigated by incorporating additional image crops from cast images, thereby improving the robustness of the AI models.

Self-Supervision and Vision-Language Models

Research Question #3

Can self-supervision techniques be adapted to utilize the inherent supervision within bone radiographic data and associated French reports ?

In Chapter 8 we demonstrated the potential of self-supervised learning to train robust AI models for medical imaging without extensive manual annotation. Using raw radiological images and associated French reports, we developed a pipeline that significantly reduces reliance on annotated data.

Research Question #4

How can these methods be optimized to reduce the need for costly annotations in medical imaging ?

Our exploration into vision-language pretraining and the automatic generation of pseudo-labels in Chapter 8 demonstrated methods to reduce the need for costly annotations. Additionally, generating standardized report and question-answer pairs in Chapter 9 laid the groundwork for training a vision-language model, thereby optimizing data utilization in medical imaging.

10.1.2 Summary of contributions

Explainability

- Development of Poly-CAM for high-resolution saliency maps in CNNs.
- Introduction of Transformer Input Sampling (TIS) for visual transformers.
- Application of Poly-CAM to identify and mitigate biases in bone radiographs.

Self-Supervision and Vision-Language Models

- Development of a French adaptation of the DEDUCE [107] method for pseudonymization of medical reports in French.

- Creation of a comprehensive pipeline to leverage medical data from a single hospital for self-supervised learning using radiological images and French reports.

- Exploration of vision-language pretraining and pseudo-label generation to reduce annotation needs.

10.2 Future Directions

This thesis suggests several promising areas for future research. Some of the most compelling ones include:

- **More reliable evaluation of explainability methods:** Evaluating XAI (Explainable Artificial Intelligence) methods is challenging due to the lack of clear and universally accepted metrics. Currently, no single metric can effectively rank methods reliably, as the evaluation of XAI is often subjective and context-dependent. Establishing more definitive ways to measure the effectiveness of explainability methods will be crucial for future advancements.

- **Exploring other paradigms of explainability:** Reducing explanations to 2D saliency maps has limitations. Investigating alternative methods, such as narrative explanations or model-driven explanations, could provide richer and more intuitive insights.

- **Advanced Vision-Language Models:** The development of robust vision-language models that seamlessly integrate visual and textual data could lead to significant improvements in AI-assisted medical imaging analysis. These models can enhance the understanding and integration of radiographic images with clinical data, facilitating richer reasoning and support for treatment and follow-up.

- **Exploring Generative Text from Images:** Future work could focus on developing systems that generate descriptive text from medical images and answer visual questions. This would aid clinical documentation and enhance AI explainability by providing natural language justifications for predictions, thereby improving model interpretability and trustworthiness.

10.3 Final Thoughts

The AI revolution is poised to redefine modern medicine, promising significant improvements in diagnosis, treatment, and patient care. This thesis makes a modest contribution to this transformation by developing and validating techniques for explainability and self-supervision. For the future, the integration of these methodologies with advanced vision-language models and explainability tools holds great potential. By continuing to push the boundaries of AI research and addressing the challenges specific to medical applications, we can pave the way for a more transparent, reliable, and efficient healthcare system where AI supports rather than replaces medical professionals.

In conclusion, this thesis highlights the importance of interdisciplinary collaboration and the need to keep pace with the frenetic evolution of technology.

In the end, the most important contribution of this thesis is probably not its scientific contribution, but the intellectual and human experience it represented for me, a small medical doctor with a big passion for technology.

Bibliography

[1] Abedeen, I, Rahman, MA, Prottyasha, FZ, Ahmed, T, Chowdhury, TM, et al. (2023). Fracatlas: A dataset for fracture classification, localization and segmentation of musculoskeletal radiographs. *Scientific Data*, 10(1):521.

[2] Abnar, S and Zuidema, W (2020). Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197.

[3] Adadi, A and Berrada, M (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access*, 6:52138–52160.

[4] Adebayo, J, Gilmer, J, Muelly, M, Goodfellow, I, Hardt, M, et al. (2018). Sanity checks for saliency maps. In Bengio, S, Wallach, H, Larochelle, H, Grauman, K, Cesa-Bianchi, N, et al., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[5] Adebayo, J, Gilmer, J, Muelly, M, Goodfellow, I, Hardt, M, et al. (2018). Sanity checks for saliency maps. *Advances in neural information processing systems*, 31.

[6] Akiba, T, Sano, S, Yanase, T, Ohta, T, and Koyama, M (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

[7] Alayrac, JB, Donahue, J, Luc, P, Miech, A, Barr, I, et al. (2022). Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

[8] Alsentzer, E, Murphy, J, Boag, W, Weng, WH, Jindi, D, et al. (2019). Publicly available clinical bert embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.

[9] Alvarez-Melis, D and Jaakkola, TS (2018). On the robustness of interpretability method. In *Proceedings of the 2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*.

[10] Alvarez-Melis, D and Jaakkola, TS (2018). Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 7786–7795. Curran Associates Inc., Red Hook, NY, USA.

[11] Arya, V, Bellamy, RK, Chen, PY, Dhurandhar, A, Hind, M, et al. (2021). One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. In *INFORMS Annual Meeting*.

[12] Bach, S, Binder, A, Montavon, G, Klauschen, F, Müller, KR, et al. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

[13] Bajorath, J, Kearnes, S, Walters, WP, Meanwell, NA, Georg, GI, et al. (2020). Artificial intelligence in drug discovery: Into the great wide open. *Journal of Medicinal Chemistry*, 63(16):8651–8652. doi: 10.1021/acs.jmedchem.0c01077. PMID: 32639156, `arXiv:https://doi.org/10.1021/acs.jmedchem.0c01077`.

[14] Beyer, L, Izmailov, P, Kolesnikov, A, Caron, M, Kornblith, S, et al. (2023). Flexivit: One model for all patch sizes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14496–14506.

[15] Bhatt, U, Weller, A, and Moura, JM (2021). Evaluating and aggregating feature-based model explanations. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3016–3022.

[16] Brown, N, Ertl, P, Lewis, R, Luksch, T, Reker, D, et al. (2020). Artificial intelligence in chemistry and drug design. *Journal of Computer-Aided Molecular Design*, 34:709–715.

[17] Brown, T, Mann, B, Ryder, N, Subbiah, M, Kaplan, JD, et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[18] Caron, M, Touvron, H, Misra, I, Jégou, H, Mairal, J, et al. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660.

[19] Carrell, D, Malin, B, Aberdeen, J, Bayer, S, Clark, C, et al. (2013). Hiding in plain sight: use of realistic surrogates to reduce exposure of protected

health information in clinical text. *Journal of the American Medical Informatics Association*, 20(2):342–348.

[20] Chalasani, P, Chen, J, Chowdhury, AR, Wu, X, and Jha, S (2020). Concise explanations of neural networks using adversarial training. In *International Conference on Machine Learning*, pages 1383–1391. PMLR.

[21] Chattopadhay, A, Sarkar, A, Howlader, P, and Balasubramanian, VN (2018). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE.

[22] Chefer, H, Gur, S, and Wolf, L (2021). Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 397–406.

[23] Chefer, H, Gur, S, and Wolf, L (2021). Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791.

[24] Chen, J, Li, X, Yu, L, Dou, D, and Xiong, H (2022). Beyond intuition: Rethinking token attributions inside transformers. *Transactions on Machine Learning Research*.

[25] Chen, T, Kornblith, S, Norouzi, M, and Hinton, G (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

[26] Chen, X, Liang, C, Huang, D, Real, E, Wang, K, et al. (2024). Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36.

[27] Cheng, P, Lin, L, Lyu, J, Huang, Y, Luo, W, et al. (2023). Prior: Prototype representation joint learning from medical images and reports. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21361–21371.

[28] Clevert, DA, Unterthiner, T, and Hochreiter, S (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

[29] Conneau, A, Khandelwal, K, Goyal, N, Chaudhary, V, Wenzek, G, et al. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

[30] Conneau, A and Lample, G (2019). *Cross-lingual language model pretraining*. Curran Associates Inc., Red Hook, NY, USA.

[31] Dahl, GE, Sainath, TN, and Hinton, GE (2013). Improving deep neural networks for LVCSR using rectified linear units and dropout. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE.

[32] Desai, K and Johnson, J (2021). Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11162–11173.

[33] Dettmers, T, Lewis, M, Shleifer, S, and Zettlemoyer, L (2022). 8-bit optimizers via block-wise quantization. In *International Conference on Learning Representations*.

[34] Dettmers, T, Pagnoni, A, Holtzman, A, and Zettlemoyer, L (2024). Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

[35] Devlin, J, Chang, MW, Lee, K, and Toutanova, K (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[36] Dosovitskiy, A, Beyer, L, Kolesnikov, A, Weissenborn, D, Zhai, X, et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[37] Duchi, J, Hazan, E, and Singer, Y (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

[38] Elfwing, S, Uchibe, E, and Doya, K (2018). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11. doi: https://doi.org/10.1016/j.neunet.2017.12.012. Special issue on deep reinforcement learning.

[39] Ellingrud, K, Sanghvi, S, Madgavkar, A, Dandona, GS, Chui, M, et al. (2023). Generative ai and the future of work in america.

[40] Englebert, A, Cornu, O, and de Vleeschouwer, C (2022). Backward recursive class activation map refinement for high resolution saliency map. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 2444–2450. IEEE.

[41] Falkner, S, Klein, A, and Hutter, F (2018). BOHB: Robust and efficient hyperparameter optimization at scale. In Dy, J and Krause, A, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1437–1446. PMLR.

[42] Fernandez, FG (2020). Torchcam: class activation explorer. https://github.com/frgfm/torch-cam.

[43] Frantar, E, Ashkboos, S, Hoefler, T, and Alistarh, D (2023). Gptq: Accurate post-training quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*.

[44] Fukushima, K (1969). Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics*, 5(4):322–333. doi: 10.1109/TSSC.1969.300225.

[45] Fukushima, K (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.

[46] Gal, Y and Ghahramani, Z (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

[47] Ghorbani, A, Abid, A, and Zou, J (2019). Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3681–3688.

[48] Gu, Y, Tinn, R, Cheng, H, Lucas, M, Usuyama, N, et al. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

[49] Halabi, SS, Prevedello, LM, Kalpathy-Cramer, J, Mamonov, AB, Bilbily, A, et al. (2019). The rsna pediatric bone age machine learning challenge. *Radiology*, 290(2):498–503.

[50] He, K, Chen, X, Xie, S, Li, Y, Dollár, P, et al. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

[51] He, K, Zhang, X, Ren, S, and Sun, J (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[52] Hedström, A, Weber, L, Krakowczyk, D, Bareeva, D, Motzkus, F, et al. (2023). Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research*, 24(34):1–11.

[53] Hendrycks, D and Gimpel, K (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

[54] Heydon, P, Egan, C, Bolter, L, Chambers, R, Anderson, J, et al. (2021). Prospective evaluation of an artificial intelligence-enabled algorithm for automated diabetic retinopathy screening of 30 000 patients. *British Journal of Ophthalmology*, 105(5):723–728.

[55] Hirasa, Y (1991). Onboard navigation system operating via gps. Technical report, SAE Technical Paper.

[56] Hochreiter, S and Schmidhuber, J (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[57] Holtzman, A, Buys, J, Du, L, Forbes, M, and Choi, Y (2020). The curious case of neural text degeneration. In *International Conference on Learning Representations*.

[58] Hopfield, JJ (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

[59] Hu, EJ, Wallis, P, Allen-Zhu, Z, Li, Y, Wang, S, et al. (2022). Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

[60] Huang, SC, Shen, L, Lungren, MP, and Yeung, S (2021). Gloria: A multimodal global-local representation learning framework for label-efficient medical image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3942–3951.

[61] Huber, PJ (1964). Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35:492–518.

[62] Irvin, J, Rajpurkar, P, Ko, M, Yu, Y, Ciurea-Ilcus, S, et al. (2019). Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597.

[63] Jain, S, Agrawal, A, Saporta, A, Truong, S, Bui, T, et al. (2021). Radgraph: Extracting clinical entities and relations from radiology reports. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

[64] Jalwana, MA, Akhtar, N, Bennamoun, M, and Mian, A (2021). Cameras: Enhanced resolution and sanity preserving class activation mapping for image saliency. In *CVPR*.

[65] Jia, C, Yang, Y, Xia, Y, Chen, YT, Parekh, Z, et al. (2021). Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.

[66] Jia, X, Ren, L, and Cai, J (2020). Clinical implementation of ai technologies will require interpretable ai models. *Medical physics*, (1):1–4.

[67] Jiang, AQ, Sablayrolles, A, Mensch, A, Bamford, C, Chaplot, DS, et al. (2023). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

[68] Jiang, AQ, Sablayrolles, A, Roux, A, Mensch, A, Savary, B, et al. (2024). Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

[69] Jiang, PT, Zhang, CB, Hou, Q, Cheng, MM, and Wei, Y (2021). Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888.

[70] Johnson, A, Pollard, T, Mark, R, Berkowitz, S, and Horng, S (2019). Mimic-cxr database (version 2.0. 0). physionet.

[71] Johnson, AE, Pollard, TJ, Shen, L, Lehman, LwH, Feng, M, et al. (2016). Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

[72] Jumper, J, Evans, R, Pritzel, A, Green, T, Figurnov, M, et al. (2021). Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589.

[73] Kingma, DP (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[74] Kirubarajan, A, Young, D, Khan, S, Crasto, N, Sobel, M, et al. (2022). Artificial intelligence and surgical education: a systematic scoping review of interventions. *Journal of Surgical Education*, 79(2):500–515.

[75] Kokhlikyan, N, Miglani, V, Martin, M, Wang, E, Alsallakh, B, et al. (2020). Captum: A unified and generic model interpretability library for pytorch. arXiv preprint arXiv:2009.07896. `arXiv:cs.LG/2009.07896`.

[76] Krizhevsky, A, Sutskever, I, and Hinton, GE (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F, Burges, C, Bottou, L, and Weinberger, K, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[77] Labrak, Y, Bazoge, A, Dufour, R, Rouvier, M, Morin, E, et al. (2023). DrBERT: A Robust Pre-trained Model in French for Biomedical and Clinical domains. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL'23), Long Paper*. Association for Computational Linguistics, Toronto, Canada.

[78] Lakshminarayanan, B, Pritzel, A, and Blundell, C (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

[79] Lapuschkin, S, Wäldchen, S, Binder, A, Montavon, G, Samek, W, et al. (2019). Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8.

[80] Laurençon, H, Saulnier, L, Tronchon, L, Bekman, S, Singh, A, et al. (2023). Obelics: an open web-scale filtered dataset of interleaved image-text documents. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 71683–71702.

[81] Laurençon, H, Tronchon, L, Cord, M, and Sanh, V (2024). What matters when building vision-language models? `arXiv:cs.CV/2405.02246`.

[82] LeCun, Y, Boser, B, Denker, JS, Henderson, D, Howard, RE, et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

[83] Lee, J, Yoon, W, Kim, S, Kim, D, Kim, S, et al. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

[84] Lee, S, Youn, J, Kim, M, and Yoon, SH (2023). Cxr-llava: Multimodal large language model for interpreting chest x-ray images. *arXiv preprint arXiv:2310.18341*.

[85] Li, C, Wong, C, Zhang, S, Usuyama, N, Liu, H, et al. (2024). Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36.

[86] Li, H, Li, Z, Ma, R, and Wu, T (2022). Fd-cam: Improving faithfulness and discriminability of visual explanation for cnns. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1300–1306. IEEE.

[87] Li, J, Li, D, Savarese, S, and Hoi, S (2023). Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

[88] Li, J, Selvaraju, R, Gotmare, A, Joty, S, Xiong, C, et al. (2021). Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705.

[89] Li, J, Zhang, Q, Yu, Y, Fu, Q, and Ye, D (2024). More agents is all you need. *arXiv preprint arXiv:2402.05120*.

[90] Li, LH, Yatskar, M, Yin, D, Hsieh, CJ, and Chang, KW (2019). Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

[91] Li, LH, Zhang, P, Zhang, H, Yang, J, Li, C, et al. (2022). Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975.

[92] Li, Y, Liang, F, Zhao, L, Cui, Y, Ouyang, W, et al. (2022). Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *International Conference on Learning Representations*.

[93] Lin, J, Tang, J, Tang, H, Yang, S, Chen, WM, et al. (2024). Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100.

[94] Lin, TY, Maire, M, Belongie, S, Hays, J, Perona, P, et al. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

[95] Liu, F, Vulić, I, Korhonen, A, and Collier, N (2021). Learning domain-specialised representations for cross-lingual biomedical entity linking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 565–574.

[96] Liu, H, Li, C, Li, Y, and Lee, YJ (2023). Improved baselines with visual instruction tuning. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

[97] Liu, H, Li, C, Wu, Q, and Lee, YJ (2023). Visual instruction tuning.

[98] Liu, Pr, Lu, L, Zhang, Jy, Huo, Tt, Liu, Sx, et al. (2021). Application of artificial intelligence in medicine: an overview. *Current medical science*, 41(6):1105–1115.

[99] Liu, Y, Ott, M, Goyal, N, Du, J, Joshi, M, et al. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[100] Liu, Z, Lin, Y, Cao, Y, Hu, H, Wei, Y, et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022.

[101] Loshchilov, I and Hutter, F (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

[102] Lu, J, Batra, D, Parikh, D, and Lee, S (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

[103] Maas, AL (2013). Rectifier nonlinearities improve neural network acoustic models.

[104] Van der Maaten, L and Hinton, G (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

[105] Martin, L, Muller, B, Suarez, PO, Dupont, Y, Romary, L, et al. (2020). Camembert: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219.

[106] McCulloch, WS and Pitts, W (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133.

[107] Menger, V, Scheepers, F, van Wijk, LM, and Spruit, M (2018). Deduce: A pattern matching method for automatic de-identification of dutch medical text. *Telematics and Informatics*, 35(4):727–736.

[108] Montavon, G, Samek, W, and Müller, KR (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15. doi: https://doi.org/10.1016/j.dsp.2017.10.011.

[109] Müller, H and Unay, D (2017). Retrieval from and understanding of large-scale multi-modal medical datasets: a review. *IEEE transactions on multimedia*, 19(9):2093–2104.

[110] Müller, P, Kaissis, G, Zou, C, and Rueckert, D (2022). Joint learning of localized representations from medical images and reports. In *European Conference on Computer Vision*, pages 685–701. Springer.

[111] Murdoch, WJ, Singh, C, Kumbier, K, Abbasi-Asl, R, and Yu, B (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080. doi: 10.1073/pnas.1900654116. arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.1900654116.

[112] Naidu, R, Ghosh, A, Maurya, Y, Kundu, SS, et al. (2020). Is-cam: Integrated score-cam for axiomatic-based explanations. *arXiv preprint arXiv:2010.03023*.

[113] Nevitt, M, Felson, D, and Lester, G (2006). The osteoarthritis initiative. *Protocol for the cohort study*, 1.

[114] Newell, A and Simon, H (1956). The logic theory machine–a complex information processing system. *IRE Transactions on information theory*, 2(3):61–79.

[115] Ng, A (2017). Ai is the new electricity. `https://www.youtube.com/watch?v=21EiKfQYZXc`. Speech at Stanford MSx program.

[116] Nguyen, Ap and Martínez, MR (2020). On quantitative aspects of model interpretability. *arXiv preprint arXiv:2007.07584*.

[117] Omeiza, D, Speakman, S, Cintas, C, and Weldermariam, K (2019). Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv preprint arXiv:1908.01224*.

[118] Paszke, A, Gross, S, Massa, F, Lerer, A, Bradbury, J, et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

[119] Peng, Y, Wang, X, Lu, L, Bagheri, M, Summers, R, et al. (2018). Negbio: a high-performance tool for negation and uncertainty detection in radiology reports. *AMIA Summits on Translational Science Proceedings*, 2018:188.

[120] Petsiuk, V, Das, A, and Saenko, K (2018). Rise: Randomized input sampling for explanation of black-box models. In *British Machine Vision Conference (BMVC)*.

[121] Polyak, BT (1964). Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17.

[122] Poursabzi-Sangdeh, F, Goldstein, DG, Hofman, JM, Wortman Vaughan, JW, and Wallach, H (2021). Manipulating and measuring model interpretability. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.

[123] Pruthi, D, Gupta, M, Dhingra, B, Neubig, G, and Lipton, ZC (2020). Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

[124] Radford, A, Kim, JW, Hallacy, C, Ramesh, A, Goh, G, et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

[125] Radford, A, Narasimhan, K, Salimans, T, Sutskever, I, et al. (2018). Improving language understanding by generative pre-training.

[126] Raghu, M, Unterthiner, T, Kornblith, S, Zhang, C, and Dosovitskiy, A (2021). Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128.

[127] Rajpurkar, P, Irvin, J, Bagul, A, Ding, D, Duan, T, et al. (2017). Mura: Large dataset for abnormality detection in musculoskeletal radiographs. *arXiv preprint arXiv:1712.06957*.

[128] Ramaswamy, HG et al. (2020). Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 983–991.

[129] Ramesh, V, Chi, NA, and Rajpurkar, P (2022). Improving radiology report generation systems by removing hallucinated references to non-existent priors. In *Machine Learning for Health*, pages 456–473. PMLR.

[130] Regulation, P (2016). Regulation (eu) 2016/679 of the european parliament and of the council. *Regulation (eu)*, 679:2016.

[131] Ri, R, Yamada, I, and Tsuruoka, Y (2022). mluke: The power of entity representations in multilingual pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7316–7330.

[132] Robbins, H and Monro, S (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

[133] Rodriguez-Ruiz, A, Lång, K, Gubern-Merida, A, Broeders, M, Gennaro, G, et al. (2019). Stand-alone artificial intelligence for breast cancer detection in mammography: comparison with 101 radiologists. *JNCI: Journal of the National Cancer Institute*, 111(9):916–922.

[134] Rosenblatt, F (1957). *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

[135] Rumelhart, DE, Hinton, GE, and Williams, RJ (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

[136] Russakovsky, O, Deng, J, Su, H, Krause, J, Satheesh, S, et al. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252. doi: 10.1007/s11263-015-0816-y.

[137] Samek, W, Binder, A, Montavon, G, Lapuschkin, S, and Müller, KR (2017). Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28:2660–2673.

[138] Schwab, K (2017). *The fourth industrial revolution*. Crown Currency.

[139] Selvaraju, RR, Cogswell, M, Das, A, Vedantam, R, Parikh, D, et al. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*.

[140] Serag, A, Ion-Margineanu, A, Qureshi, H, McMillan, R, Saint Martin, MJ, et al. (2019). Translational ai and deep learning in diagnostic pathology. *Frontiers in medicine*, 6:185.

[141] Serrano, S and Smith, NA (2019). Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951.

[142] Shannon, CE (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

[143] Shi, X, Khademi, S, Li, Y, and van Gemert, J (2021). Zoom-cam: Generating fine-grained pixel annotations from image labels. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10289–10296. IEEE.

[144] Shrikumar, A, Greenside, P, and Kundaje, A (2017). Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMlR.

[145] Simonyan, K, Vedaldi, A, and Zisserman, A (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. *ArXiv:1312.6034 [Cs]*.

[146] Simonyan, K and Zisserman, A (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[147] Singh, A, Hu, R, Goswami, V, Couairon, G, Galuba, W, et al. (2022). Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650.

[148] Sixt, L, Granz, M, and Landgraf, T (2020). When explanations lie: Why many modified bp attributions fail. In *International Conference on Machine Learning*, pages 9046–9057. PMLR.

[149] Smilkov, D, Thorat, N, Kim, B, Viégas, F, and Wattenberg, M (2017). Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

[150] Sobel, I and Feldman, G (1968). An isotropic 3x3 image gradient operator for image processing. *Mach. Vis. Three–Dimens. Scenes*, (June):376–379.

[151] Squarcina, L, Villa, FM, Nobile, M, Grisan, E, and Brambilla, P (2021). Deep learning for the prediction of treatment response in depression. *Journal of affective disorders*, 281:618–622.

[152] Stassin, S, Englebert, A, Albert, J, Nanfack, G, Versbraegen, N, et al. (2023). An experimental investigation into the evaluation of explainability methods for computer vision. *Communications in Computer and Information Science*.

[153] Sundararajan, M, Taly, A, and Yan, Q (2017). Axiomatic attribution for deep networks. In *International Conference on Machine Learning*. PMLR.

[154] Tajbakhsh, N, Jeyaseelan, L, Li, Q, Chiang, JN, Wu, Z, et al. (2020). Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical image analysis*, 63:101693.

[155] Tieleman, T (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26.

[156] Touchent, R, Romary, L, and de la Clergerie, E (2023). Camembert-bio: a tasty french language model better for your health. `arXiv:cs.CL/2306.15550`.

[157] Touvron, H, Cord, M, Douze, M, Massa, F, Sablayrolles, A, et al. (2021). Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR.

[158] Touvron, H, Lavril, T, Izacard, G, Martinet, X, Lachaux, MA, et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

[159] Tsimpoukelli, M, Menick, JL, Cabi, S, Eslami, S, Vinyals, O, et al. (2021). Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212.

[160] van der Velden, BH, Kuijf, HJ, Gilhuijs, KG, and Viergever, MA (2022). Explainable artificial intelligence (xai) in deep learning-based medical image analysis. *Medical Image Analysis*, 79:102470. doi: https://doi.org/10.1016/j.media.2022.102470.

[161] Vaswani, A, Shazeer, N, Parmar, N, Uszkoreit, J, Jones, L, et al. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[162] Voita, E, Talbot, D, Moiseev, F, Sennrich, R, and Titov, I (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808. ACL Anthology.

[163] Wang, F, Zhou, Y, Wang, S, Vardhanabhuti, V, and Yu, L (2022). Multi-granularity cross-modal alignment for generalized medical visual representation learning. *Advances in Neural Information Processing Systems*, 35:33536–33549.

[164] Wang, G, Li, W, Aertsen, M, Deprest, J, Ourselin, S, et al. (2019). Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45.

[165] Wang, H, Naidu, R, Michael, J, and Kundu, SS (2020). Ss-cam: Smoothed score-cam for sharper visual feature localization. *arXiv preprint arXiv:2006.14255*.

[166] Wang, H, Wang, Z, Du, M, Yang, F, Zhang, Z, et al. (2020). Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, workshop on Fair, Data Efficient and Trusted Computer Vision*.

[167] Wang, W, Han, C, Zhou, T, and Liu, D (2023). Visual recognition with deep nearest centroids. In *International Conference on Learning Representations (ICLR)*.

[168] Wang, Z, Yu, J, Yu, AW, Dai, Z, Tsvetkov, Y, et al. (2022). Simvlm: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations*.

[169] Wenzek, G, Lachaux, MA, Conneau, A, Chaudhary, V, Guzmán, F, et al. (2020). CCNet: Extracting high quality monolingual datasets from web crawl data. In Calzolari, N, Béchet, F, Blache, P, Choukri, K, Cieri, C, et al., editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012. European Language Resources Association, Marseille, France. ISBN 979-10-95546-34-4.

[170] Wightman, R (2019). Pytorch image models. `https://github.com/rwightman/pytorch-image-models`. doi: 10.5281/zenodo.4414861.

[171] Xiao, B, Wu, H, Xu, W, Dai, X, Hu, H, et al. (2024). Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829.

[172] Xie, W, Li, XH, Cao, CC, and Zhang, NL (2023). Vit-cx: causal explanation of vision transformers. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 1569–1577.

[173] Xie, Z, Zhang, Z, Cao, Y, Lin, Y, Bao, J, et al. (2022). Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663.

[174] Xu, K, Ba, J, Kiros, R, Cho, K, Courville, A, et al. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.

[175] Xu, Y, Hosny, A, Zeleznik, R, Parmar, C, Coroller, T, et al. (2019). Deep learning predicts lung cancer treatment response from serial medical imaging. *Clinical Cancer Research*, 25(11):3266–3275.

[176] Yamada, I, Asai, A, Shindo, H, Takeda, H, and Matsumoto, Y (2020). Luke: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454.

[177] Yeh, CK, Hsieh, CY, Suggala, A, Inouye, DI, and Ravikumar, PK (2019). On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32:10967–10978.

[178] Yuan, T, Li, X, Xiong, H, Cao, H, and Dou, D (2021). Explaining information flow inside vision transformers using markov chain. In *eXplainable AI approaches for debugging and diagnosis.*

[179] Zeiler, MD and Fergus, R (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

[180] Zhang, J, Bargal, SA, Lin, Z, Brandt, J, Shen, X, et al. (2018). Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102.

[181] Zhang, Y, Jiang, H, Miura, Y, Manning, CD, and Langlotz, CP (2022). Contrastive learning of medical visual representations from paired images and text. In *Machine Learning for Healthcare Conference*, pages 2–25. PMLR.

[182] Zhou, B, Khosla, A, Lapedriza, A, Oliva, A, and Torralba, A (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.