DEPARTMENT OF DIGITAL SYSTEMS

NCSR DEMOKRITOS
INSTITUTE OF INFORMATICS AND
TELECOMMUNICATIONS

Grad-CAM vs HiResCAM: A comparative study via quantitative evaluation metrics

by

Vaggelis Lamprou

Submitted

in partial fulfilment of the requirements for the degree of

Master of Artificial Intelligence

at the

UNIVERSITY OF PIRAEUS

May 2023

Author . . . . . Vaggelis Lamprou . . . . . .

II-MSc “Artificial Intelligence”
May 30, 2023

Certified by. . . . . .

Ilias Maglogiannis
Professor
Thesis Supervisor

Certified by. . . . . .

Michael Filippakis
Professor
Member of Examination Committee

Certified by. . . . . .

Orestis Telelis
Assistant Professor
Member of Examination Committee

Grad-CAM vs HiResCAM: A comparative study via quantitative evaluation metrics

By

Vaggelis Lamprou

Submitted to the II-MSc "Artificial Intelligence" on
May 30, 2023,
in partial fulfillment of the
requirements for the MSc degree

Abstract

In this study we utilize the Grad-CAM and HiResCAM attribution map methods and consider a setting where the HiResCAM algorithm provably produces faithful explanations while Grad-CAM does not. This theoretical result motivates us to investigate the quality of their attribution maps in terms of quantitative evaluation metrics and examine if faithfulness aligns with the metrics results. Our evaluation scheme implements the well-established AOPC and Max-Sensitivity scores along with the recently introduced HAAS score and utilizes ResNet and VGG pre-trained architectures trained on four medical image datasets. The experimental results suggest that Max-Sensitivity and AOPC favour faithfulness. On the other hand, HAAS does not contribute meaningful values to our comparison, but rather inspires further study about its nature.

Thesis Supervisor: Professor Ilias Maglogiannis
Title: Grad-CAM vs HiResCAM: A comparative study via quantitative evaluation metrics

Acknowledgments

I want to express my gratitude to Professor Ilias Maglogiannis, the members of the examinations Committee, Professor Michael Filippakis and Assistant Professor Orestis Telelis and the PhD candidate Athanasios Kallipolitis from the Department of Digital Systems at the University of Piraeus, for their support and supervision towards the completion of my thesis. Additionally, I am thankful to my parents and friends for their unwavering support, understanding, and encouragement throughout the past year.

Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the «funding body» or the view of University of Piraeus and Inst. of Informatics and Telecom. of NCSR "Demokritos".

Contents

List of Figures

List of Tables

1 Introduction

In recent years, Convolutional Neural Networks (CNN) have gained popularity as a powerful tool and have revolutionized the field of Deep Learning (DL). These models are capable of processing large amounts of image data and have been successfully applied in fields such as speech recognition, computer vision and autonomous vehicles, among others, with impressive predictive accuracy. One of the reasons for this progress is their ability to automatically learn feature representations from raw data, through a process called *convolution*, reducing the need and risks of manual feature engineering that traditional Machine Learning (ML) techniques require.

However, this success usually comes at the cost of increased model complexity. In order to learn vast amounts of data, it is common to build architectures with thousands or millions of parameters that are difficult to understand and explain the reasoning behind their decisions. These models, may suffer from a *trade-off* between their performance and ability to generate explainable predictions. From a human perspective, it is crucial to strike a balance between the two to maintain the model's usability.

1.1 Explainable AI

Developing a model that can generalize well is an essential attribute that all models should possess. It refers to the ability of the trained model to perform well on unseen data and it is an indication that the model has learned the underlying patterns and relationships in the data, rather than just memorizing the training examples. However, even if the model produces accurate predictions, is it acceptable for its predictions to remain unexplainable? The lack of transparency in AI systems can lead to a significant *lack of trust*, especially when it comes to modelling situations that involve moral and ethical considerations [1]. For instance, in [2] the authors consider a scenario where a model is trained on data that describe the financial status of people who have applied for a bank loan. One of the features included in the dataset is the *sex* of the applicant and therefore it is crucial to ensure that the learned model does not discriminate on whether the applicant is male or female to approve or reject their application. Furthermore, in critical sectors such as healthcare and self-driving cars, errors in a model's reasoning can endanger human lives and damage public opinion about AI. Failure to detect cancer or anticipate a potential car accident [3] are examples of such extreme cases. In order to address these concerns, it is important to develop explainable systems that can provide insights into how a model reaches its decisions. This will enable stakeholders to understand the model's reasoning and increase their trust in AI systems.

Apart from the aforementioned use cases, seeking explanations for a model's decisions can significantly benefit Machine Learning practitioners at the pre-production level.

Understanding a system's behaviour can prevent potential future problems and identify vulnerabilities that otherwise may go unnoticed. Moreover, it results in gathering information, acquiring new knowledge and broadening their understanding of the topic of interest.

Such kinds of concerns and benefits led to the rise of the *eXplainable Artificial Intelligence* (XAI) field, which aims at assisting Machine Learning engineers create trustworthy, fair, robust, and high-performing AI models for real-world applications. As explained in [4], the term was initially introduced in 2004, where the authors attempted to *explain* a model's behaviour in a simulation game, while a few years later, in 2016, one of the most well-known XAI algorithms was defined: LIME [5]. LIME (Local Interpretable Model-Agnostic Explanations) provides a framework for explaining the model predictions by approximating its decision boundary in the local vicinity of a specific input. The paper also includes two very instructive applications that showcase how LIME's explanations did actually help in real-life problems, apart from the theoretical point of view. The first model that is explained is an SVM which is trained to separate "Christianity" from "Atheism" related texts and has a 94% testing accuracy. When examined on specific text samples thought, researchers found out that it relies on words like "Posting" and "Host" to make predictions, which undoubtedly have nothing to do with the classes of interest. The second model is an Inception CNN where LIME successfully produces detailed explanations on how the model distinguishes between similar classes, such as an acoustic and an electric guitar. In addition, although not as popular as LIME, in 2015 the LRP [6] algorithm was introduced as a method to decipher CNN models. We briefly sketch it later in this chapter.

Ever since a lot of new algorithms have emerged and the field has sparked the interest of the scientific community in multiple cases. As we see in Figure 1, taken from [4], the Google searches under "Explainable Artificial Intelligence" have drastically increased since 2016.

Figure 1: "Explainable Artificial Intelligence" Google trends ([4])

While looking for *Explainability* information on-line, one will sooner or later encounter the term *Interpretability* as well. Until now there are not concrete mathematical formulations to describe none of them, resulting in a vague relationship where their usage

varies depending on the author. As explained in [7], many researchers use the terms *interchangeably* while there are also many that distinguish between them. In regards to the latter ones, *Interpretability* can be though as "*the degree to which an observer can understand the cause of a decision*" (Miller [8]), relating the term with the intuition behind the outputs of the model. On the other hand, in [7], we see that *Explainability* is mostly related to the understanding humans get about the *internal mechanisms of the model*. Consequently, as referenced in [9], "*explainable models are interpretable by default, but the reverse is not always true*". The focus of this study does not lie in an area where the two terms need to be handled differently and as a result one may you use them *interchangeably*.

Currently, there are numerous algorithms that comprise the field of XAI. In [4], [10], [11], [12], [13], [7], [14] one may find taxonomies organized from different scopes that aim to present a view of the field. In all of them though, there are some general taxonomy criteria that are the same and define a common ground to describe XAI algorithms. That being said, an algorithm might explain either a particular sample point or the entire model, called *local* or *global* respectively; and it might depend on a particular type of model or not; called *model-specific* or *model-agnostic*. In addition, there are models whose outcomes are *intrinsically (inherently)* interpretable due to their architecture and models whose explanations are computed *post-hoc*, i.e. after the training is completed. The former case includes *white box* models such as Decision Trees and Regression (see [15]) and the latter *black box* models such as Random Forests, SVMs and all kinds of ANNs. Last but not least, in between, it is important to mention that there are algorithms which yield *Interpretability by design* explanations, in the sense that they incorporate interpretability during the training process.

Attribution maps

In this study we focus on a special sub-category of XAI algorithms called *attribution map* methods. It consists of local and model-agnostic algorithms which take as input a test image and a learned CNN classifier and calculate pixel-level explanations in the form of a 2-dimensional attribution map. Each attribution pixel describes the image pixel importance to the predicted class. Following [12], one might sub-categorize attribution map methods into *gradient based*, *perturbation based*, *trainable attention* and the unique Layerwise Relevance Propagation (LPR) [6] and Deep Learning Important FeaTures (DeepLIFT) [16] algorithms.

In short, *gradient based* methods utilize class score gradients with respect to feature maps to identify the important parts of the input image. Some common instances include Gradient * Input [17], SmoothGrad [18] and the family of CAM-based algorithms which we are going to explore in this study. *Perturbation based* methods -as the name suggests- apply perturbations to the image parts in order to identify the locations that

are most important for the model's prediction. Examples of this sub-category include LIME [5], SHAP [19] and Occlusion [20] and they are all characterized by the fact that when applied on images they are time-demanding to implement as they require multiple iterations per image. On the other hand, the gradient based methods are usually fast to compute. Further to the above post-hoc algorithms, attribution maps can also be computed by *trainable attention* methods which enforce Interpretability by design. This special kind of algorithms are an integral part of the model architecture and are used to guide the learning process so that the model develops attention mechanisms while training. In [21] the GAIN (Guided Attention Inference Network) algorithm is introduced, while in [22] the authors explain how to incorporate attention gates to any standard CNN architecture.

Finally, honourable mention should be made to the famous LRP [6] and DeepLIFT [16] algorithms which have been used in numerous applications. On the one hand, LRP assigns relevance scores to each neuron in the output layer, and then propagates these scores backwards through the network to the input layer using a set of propagation rules. These rules ensure that the relevance scores are distributed appropriately among the input features or neurons in each layer, taking into account the weights and biases of the connections between the neurons. On the other hand, DeepLIFT compares the activation of each neuron to a reference activation, and then assigns a contribution score according to neuron differences. As explained in [16], adopting a difference-to-reference approach instead of using gradients allows the algorithm to propagate the signal even in cases where the gradient is zero. As a result, the authors show that the algorithm does not suffer from the *saturation* and *threshold* problems, in contrast to many gradient based methods.

1.2 Evaluation metrics

Extracting explanations is a powerful tool for machine learning practitioners as it provides a visual representation of a model's perspective and establishes a connection between what happens inside the model and the real world. A challenge arises though when applying two different XAI algorithms to calculate attributions for a given model and test image, resulting in two distinct attribution maps for the same image and model. In such cases, the practitioners must decide which attribution map is *better*. Similarly, training two different models on the same dataset and using the same XAI algorithm to extract attributions of a given test image can lead to the production of different attribution maps for each model. Ultimately, the practitioners must *choose* between these two attribution maps.

One may consider two types of measures for evaluating explanations: objective measures and subjective measures. The majority of evaluations for explanations have been

subjective measures as explanations are primarily designed for human understanding. These measures include displaying the explanation to a person, ideally a field expert, or crowd-sourced evaluations of human satisfaction showcasing their ability to comprehend the model. However, there are many reasons suggesting that it is important to consider objective measures as well. At first, establishing metrics puts the problem into a sound theoretical foundation which in turn makes it accessible to the entire scientific community and favours further studies and development. Secondly, from a practical and time perspective, it is impossible to always have a field expert (ex. a doctor) nearby and ever worse to ask them go through thousands of explanations one by one. Finally, objective metrics could provide Machine Learning practitioners with tools that allow them to work independently of field experts and make scientific contribution stemming from their own experience and point of view as well.

An objective evaluation can take the form of a general axiom or property that is either satisfied or not or the form of a quantitative evaluation metric that assigns a value to the produced explanation. The first category includes *properties* such as: Sensitivity [23], Implementation Invariance [23], Continuity [24], Selectivity [24], Sensitivity-N [25] and Summation to Delta [16]. On the other hand, for the purposes of this study, we will use metrics that can *quantify* the quality of attribution maps, as introduced in section 1.1. They are presented in detail in chapter 3. This area of study is currently growing and one may find a couple of metrics that serve this purpose. Among them, the most used one is the Area Over Perturbation Curve (AOPC) score [26], which is a technique developed in the setting of heatmap evaluation in 2015, while, in an attempt to address AOPC limitations, in 2022, the HAAS [27] metric was introduced. Analogously, in [28] the authors introduce two additional quantitative measures: Max-Sensitivity and (In)Fidelity which in contrast to the aforementioned metrics can be applied to tabular data as well.

1.3 The purpose of this study

In this study we consider two gradient based XAI algorithms of the CAM-family: Grad-CAM [29] and HiResCAM [30]. HiResCAM is an adaptation of Grad-CAM that addresses the limitations present in the Grad-CAM's Gradient Averaging Step. Unlike Grad-CAM, which averages feature map gradients, HiResCAM preserves the gradient effect on the pixel level aiming to produce high-resolution attribution mappings.

For the purposes of this study, an attribution map method will be considered *faithful* to a model if the sum of the attribution map values reflect the class score calculation. Based on theory included in [30] and described in detail in chapter 3, when the CNN architecture is of the form *Conv - Flatten - Class Scores* and the XAI algorithm class gradients are computed with respect to the last convolutional layer of the network, then one can prove that HiResCAM is faithful to the model (see formula (20) of chapter 3).

On the other hand, Grad-CAM's attribution maps do not exhibit analogous behaviour. This fact means that HiResCAM attribution maps faithfully highlight the locations the model identifies the class.

Motivated from this theoretical result, we want to quantify the quality of the Grad-CAM and HiResCAM attribution maps in the above setting and examine if the AOPC, Max-Sensitivity and HAAS metrics favour the HiResCAM attribution map.

Last but not least, we highlight that deriving a strict mathematical formulation which connects Grad-CAM with HiResCAM attribution maps in the context of any of the above metrics seems rather unlikely. As a result, the only way to approach such a problem is via case specific applications where we test a variety of datasets and models and analyse the results. For our experiments, we employed the widely recognized CRC (colon tissues), Covid-19 Database (X-rays), HAM10000 (skin tissues), and BreakHis (breast tissues) medical image datasets and customized pre-trained ResNet and VGG architectures to the aforementioned setting.

2 Background

In this chapter, we first dive into the class of CAM-based attribution map algorithms in section 2.1, by describing their definitions, advantages and limitations. The subsequent section 2.2 focuses on presenting evaluation metrics which quantify the quality of the attribution maps: in sections 2.2.1 and 2.2.2 we describe the well-established AOPC and Max-Sensitivity scores respectively, while in section 2.2.3 we meet the recently introduced HAAS score. Finally, section 2.3 is dedicated to scientific works which are related to the topics we consider in this study.

2.1 The CAM-based algorithms

The algorithms that belong to the CAM-based class of explainability algorithms produce class-specific explanations. Given a trained model and a test image the algorithm calculates a 2-dimensional class *attribution map* with the same shape as the test image whose pixel values indicate how each pixel affects the model's score. A positive pixel value means that this pixel increases the model's confidence for the class while a negative one that it decreases it.

2.1.1 CAM algorithm

The idea of generating Class Activation Maps (CAM) was firstly introduced in [31] in 2016 and it applies to CNNs of the *CAM architecture*. This means that the convolutional part of the network is followed by a Global Average Pooling (GAP) layer and a dense layer of the raw class scores.

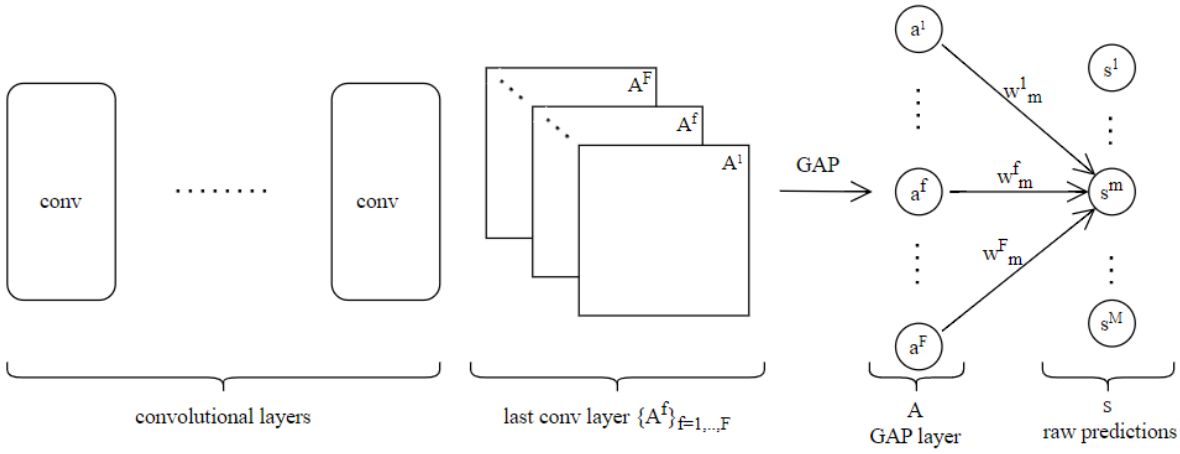


Figure 2: The CAM architecture

In this setting, let $\{A^f\}_{f=1,..,F}$ the feature maps of the last convolutional layer of the network, where $A^f \in \mathbb{R}^{D_1 * D_2}$ for $D_1, D_2 \in \mathbb{N}$ the pixel length and height of the feature map. The GAP layer node defined by A^f is given by

$$a^f = \frac{1}{D_1 D_2} \sum_{i=1}^{D_1} \sum_{j=1}^{D_2} A_{ij}^f \tag{1}$$

Definition 1 (CAM). *For class $m = 1, 2, .., M$, the CAM attribution map of the class is given by*

$$\mathcal{A}_m^{CAM} = \sum_{f=1}^{F} w_m^f A^f \tag{2}$$

By looking at the definition, we note that the feature map A^f is weighted by the weight w_m^f which can be thought as an importance score capturing the connection between A^f and the target score s^m.

The great advantage of the CAM attribution map, as proved in section 3.1 below, is that the sum of its scores directly contribute to the calculated raw class score, allowing for confident explanations of the locations the model used for its prediction. Some examples of well-known pre-trained CNNs that are of the CAM architecture include ResNet [32], DenseNet [33], GoogLeNet [34] and EfficientNet [35].

2.1.2 GradCAM algorithm

As one may easily notice though, the CAM algorithm applies to a narrow class of models since it cannot produce explanations in cases where a multi-layer dense classifier part is present or a Flatten layer is used instead of the GAP layer. In addition, it allows insights derived only from the last convolutional layer of the network.

As a result, in 2019, in an attempt to address these limitations and produce explanations for a quite larger class of CNNs, the Grad-CAM algorithm was introduced. As explained in [29], it is an extension of CAM applicable to any CNN with a differentiable network between the final convolutional layer and the prediction layer and produces class explanations with respect to any convolutional layer of the network.

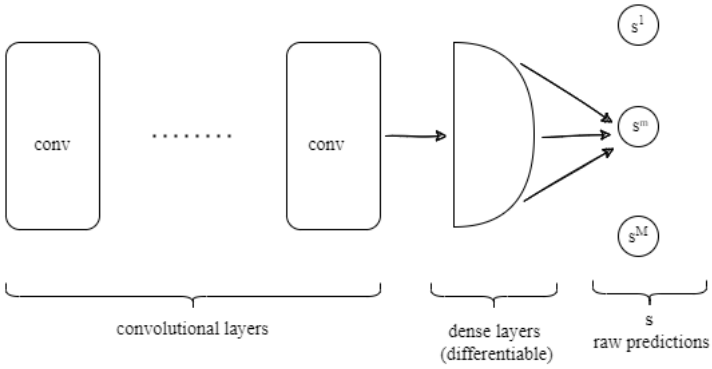


Figure 3: CNN architecture that Grad-CAM and HiResCAM are applicable

Definition 2 (Grad-CAM). *We denote by* $\{A^f\}_{f=1,..,F}$ *a convolutional layer (not necessarily the last one), where* $A^f \in \mathbb{R}^{D_1 * D_2}$, $D_1, D_2 \in \mathbb{N}$ *the pixel length and height of the feature map. For class* $m = 1, 2, .., M$, *the Grad-CAM attribution map with respect to* $\{A^f\}_{f=1,..,F}$ *is given by*

$$\mathcal{A}_m^{\textit{Grad-CAM}} = \textit{ReLU}\left(\sum_{f=1}^{F} a_m^f A^f\right) \tag{3}$$

where

$$a_m^f = \frac{1}{D_1 D_2} \sum_{i=1}^{D1} \sum_{j=1}^{D_2} \frac{\partial s^m}{\partial A_{ij}^f} \quad \textit{(Gradient Averaging step)} \tag{4}$$

By looking at the above definition, we observe that the main idea behind the extension to a larger set of networks is to incorporate the GAP calculation of the CAM architecture into the Gradient Averaging step of Grad-CAM. In that way, a_m^f of (4) becomes an overall importance score of A^f capturing the connection between A^f and s^m and replacing the weight of (2). We finally note that the calculation of the gradients in (4) is achieved via back-propagation; justifying the need for differentiable dense part between the convolutional and the raw score layer of the network.

Grad-CAM's vast network area of application has laid the ground for numerous references in scientific studies and publications. At the same time, although its success is amplified with the remarkable results in the sanity checks of [36] it is also coupled with recent reliability issues ([30], [37]), as by construction when applied outside of the CAM architecture networks it does not guarantee explanations directly related to the model's class score.

2.1.3 HiResCAM algorithm

HiResCAM was introduced as a Grad-CAM alternative in 2020, in [30], aiming to make up for the lack of faithful Grad-CAM explanations in regards to the calculated class score, when the GAP layer of the CAM architecture is replaced by a Flatten one (see also Figure 7 in section 3.2).

Definition 3 (HiResCAM). *We denote by* $\{A^f\}_{f=1,..,F}$ *a convolutional layer (not necessarily the last one), where* $A^f \in \mathbb{R}^{D_1 * D_2}$, $D_1, D_2 \in \mathbb{N}$ *the pixel length and height of the feature map. For class* $m = 1, 2, .., M$, *the HiResCAM attribution map is given by*

$$\mathcal{A}_m^{HiResCAM} = ReLU\left(\sum_{f=1}^{F} \frac{\partial s^m}{\partial A^f} \odot A^f\right) \tag{5}$$

where $\odot$ *stands for the Hadamard product.*

Replacing the Grad-CAM's Gradient Averaging step with the Hadamard product (element-wise multiplication) enables HiResCAM to preserve the pixel-level gradient effect in its wholeness; considering both its value and sign. This becomes more apparent by looking at the graph of Figure 4 and the following quote, both taken from [30] page 4.

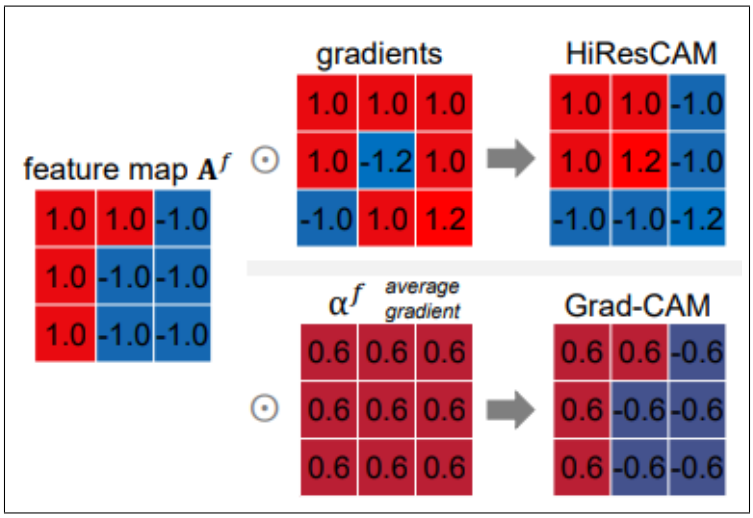


Figure 4: How HiResCAM and Grad-CAM handle the Gradients

"*...The Grad-CAM explanation matches the relative magnitudes and positivenegative pattern of the original feature map (the "inverted red L shape" here), even though the gradients suggest that some elements should be re-scaled and/or change sign. HiResCAM does not average over the gradients and instead element-wise multiplies the feature map with the gradients directly, thereby producing attention that reflects*

the model's computations and emphasizes the most important locations for a particular prediction."

As further explained in chapter 3, treating gradients in that way allows faithful explanations when the network ends in one fully connected layer and the gradients are calculated with respect to the last convolutional layer. On the other hand, Grad-CAM fails to present analogous behaviour.

2.2 Quantitative evaluation metrics

2.2.1 AOPC score

The AOPC (Area Over Perturbation Curve) score, [26], was introduced in 2016 in the context of heatmap evaluation and relies on a technique called MoRF (Most Relevant First). It is the oldest and most used metric for evaluating the quality of attribution maps and in order to be applied to our setting it requires transforming the pixel-level attribution map to a region-level map (a.k.a. *heatmap*) via Average Pooling

For the purposes of this section we denote by x a test image, f a learned mode and $H(x, f, r)$ the corresponing heatmap restricted at the pixel region r.

MoRF looks at the heatmap as a decreasing sequence of L importance regions

$$\mathcal{O} = \{r_1, r_2, .., r_L\}$$

where

$$i < j \Leftrightarrow H(x, f, r_i) > H(x, f, r_j)$$

and then performs an iterative procedure that measures how much the confidence for predicting a designated class decreases when we progressively apply perturbations to the most relevant regions, according to the order given in $\mathcal{O}$.

From a mathematical point of view, it produces:

- A *sequence of perturbated images* $x_{MoRF} = \{x^{(0)}_{MoRF}, x^{(1)}_{MoRF}, .., x^{(L)}_{MoRF}\}$, given by:

$$x^{(0)}_{MoRF} = x$$
$$x^{(k)}_{MoRF} = g(x^{(k-1)}_{MoRF}, r_k), \quad k = 1, 2, .., L,$$

  where g is a function that "removes" information from the image $x^{(k-1)}_{MoRF}$ at region r_k

- The *"MoRF Perturbation Curve"*, defined by the set of points:

$$\{(k, f(x^{(k)}_{MoRF})), \quad k = 0, 1, .., L\}.$$

The method's main idea lies in the observation that the Area Over the resulting Perturbation Curve can be used as a good reference of a heatmap's quality in the sense that an *area of greater importance* as provided by the heatmap corresponds to early and steep decreases in the graph which in turn means that the model's predictions (probability values) have essential changes only after the first few iterations. This suggests that the heatmap's most sensitive regions are accumulated in the first positions of $\mathcal{O}$ which is a desired feature of a *"good" heatmap* as it means that the heatmap can focus on the most important regions of the image.

Definition 4 (AOPC)**.** *For a test point x and a learned model f, the Area Over the Perturbation Curve can be controlled (approximated) by the quantity*

$$\sum_{k=1}^{L}[f(x^{(0)}_{MoRF}) - f(x^{(k)}_{MoRF})] \tag{6}$$

Extending to a test dataset, then (6) becomes:

$$AOPC = \frac{1}{L+1} < \sum_{k=1}^{L}[f(x^{(0)}_{MoRF}) - f(x^{(k)}_{MoRF})] > \tag{7}$$

where $< \cdot >$ denotes the average over all images in the test dataset.

An overview of the sequence of perturbated images from the MoRF procedure and the calculation of the AOPC score in terms of the difference in the class probability at each perturbation step can be found in Figure 5, which is taken from [27] page 3. The original image $x = x^0_{MoRF}$ comes from the MNIST dataset.

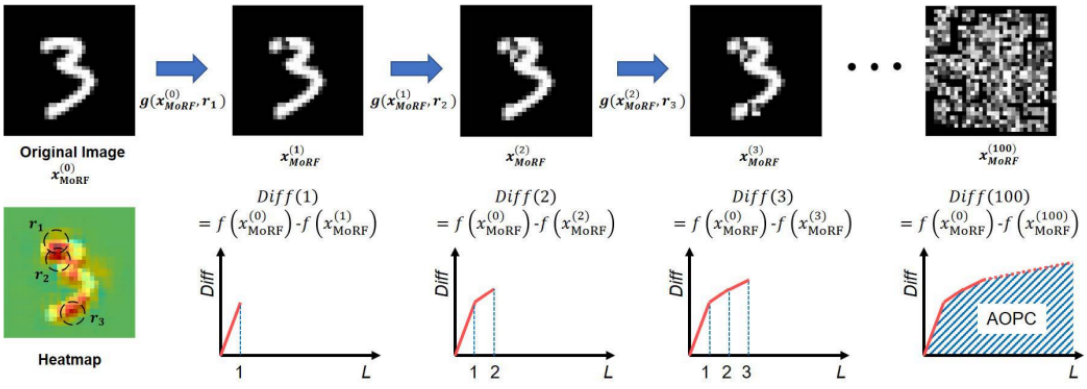


Figure 5: x_{MoRF} sequence and AOPC calculation ([27])

2.2.2 Max-Sensitivity score

The Max-Sensitivity score [28] was introduced in 2019 in an attempt to measure the degree to which the class explanation is affected by small perturbations in the test point within a certain radius. We note that it is natural to desire explanations with *low sensitivity*, as this indicates that the explanation method is more robust to small changes in the input image and produces similar explanations for similar inputs. On the other hand, *higher* values suggest that minor variations in the input lead to quite different explanations, which can make us distrust the model and the produced attribution maps.

In the following definition, the Max-Sensitivity score is calculated as the maximum distance between the class explanations of the original image and all other images within the given radius.

Definition 5 (Max-Sensitivity)**.** *Given a test image $x \in \mathbb{R}^D$, a black-box model $f \in \mathcal{F}$ such that $f : \mathbb{R}^D \to \mathbb{R}^M$, an explanation method $\Phi : \mathcal{F} \times \mathbb{R}^D \to \mathbb{R}^D$ and a radius r, we define the Max-Sensitivity for explanation as:*

$$SENS_{MAX}(\Phi, f, x, r) = \max_{y:\|x-y\|_\infty \leq r} \|\Phi(f, x) - \Phi(f, y)\|_F \tag{8}$$

where $\|\cdot\|_\infty$ is the maximum norm and $\|\cdot\|_F$ is the Frobenius (or Euclidean) norm.

Further to the above, we first note that both $\Phi(f, x)$ and $\Phi(f, y)$ should be computed for the class predicted by the model f for image x. In addition, as highlighted by the authors, it is essential to say that the formula can be robustly estimated via Monte-Carlo sampling, as implemented in the public available code [76].

2.2.3 HAAS score

Finally, we describe the HAAS (Heatmap Assisted Accuracy Score) score [27], which was presented in 2022. The main idea behind HAAS is that if an attribution map gives an *accurate* explanation then tuning the image pixels *properly* according to the attribution map should improve the predictive power of the model and reduce the number of misclassifications.

Once the image and attribution map pixels are normalized to $[-1, 1]$, the authors modify the original image by emphasizing the value of those pixels which have positive influence to the class prediction (i.e. positive pixel value becomes more positive and negative pixel value becomes more negative) and de-emphasizing the value of those pixels which have negative influence to the prediction (i.e. positive pixel value becomes less positive and negative pixel value becomes less negative). In other words, one may think of this procedure as increasing the intensity of important pixels and neutralizing the intensity

of the non-important ones. The resulting image is called the HA image and by definition it has values in $[-1, 1]$.

Definition 6 (HAAS). *Given a test set of N images x, a black-box model f and an attribution map a, we denote by x_{norm} and a_{norm} the normalized versions to $[-1, 1]$ and define the $HA = HA(x, a)$ image by:*

$$HA(x, a) = \max\{-1, \min\{1, x_{norm}(1 + a_{norm}\}\} \tag{9}$$

and the HAAS score by:

$$HAAS = \frac{Acc(f(HA(x, a)))_N}{Acc(f(x))_N} \tag{10}$$

Interpretation of HAAS score: When HAAS is *greater than 1*, the HA images improve the accuracy of the classification model, suggesting that *the attribution maps explain the features' importance well.* On the other hand, if HAAS is *less than 1*, then the accuracy of the model deteriorates, implying that *the attribution maps fail to bring out the features' importance for the model.*

As suggested by the definition, after the prediction of a classifier is explained via an attribution map, the input images are transformed as per formula (9) and are fed into the model to compute their accuracy score. Then as per formula (10) the HAAS score is the accuracy ratio of the HA over original images. The work-flow of this procedure is described in Figure 6, which is taken from [27] page 5.

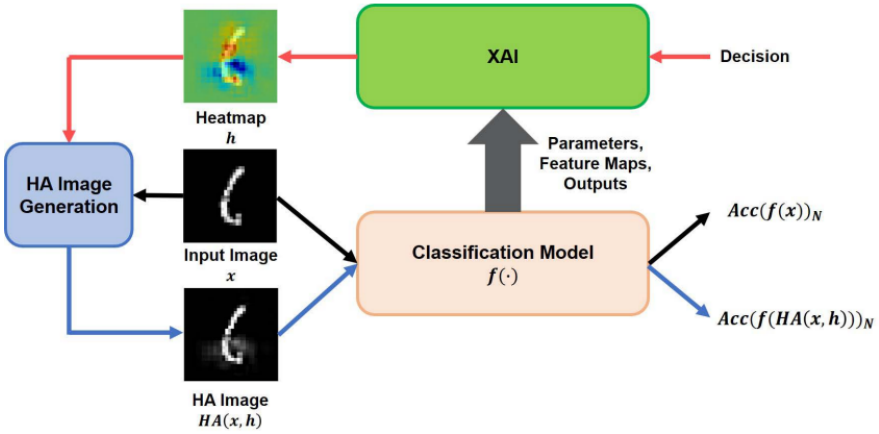


Figure 6: Architecture of HAAS evaluation scheme

An example Finally, in at attempt to get a better insight of the pixel transformations

during the HA image construction in (9), we consider the below simplified scenario where a positive and a negative pixel are transformed in cases of positive and negative influence according to the attribution map score.

For pixel x, we denote by a_x the value of attribution map at the location of pixel x and HA_x the value of HA image at pixel x.

Positive influence $a_x = \frac{1}{2}$ / Emphasizing pixel values:

- If $x = \frac{1}{2}$, then $HA_x = \max\{-1, \min\{1, \frac{3}{4}\}\} = \max\{-1, \frac{3}{4}\} = \frac{3}{4}$
  i.e. positive pixel value ($\frac{1}{2}$) becomes more positive ($\frac{3}{4}$)
- If $x = -\frac{1}{2}$, then $HA_x = \max\{-1, \min\{1, -\frac{3}{4}\}\} = \max\{-1, -\frac{3}{4}\} = -\frac{3}{4}$
  i.e. negative pixel value ($-\frac{1}{2}$) becomes more negative ($-\frac{3}{4}$)

Negative influence $a_x = -\frac{1}{2}$ / De-Emphasizing pixel values:

- If $x = \frac{1}{2}$, then $HA_x = \max\{-1, \min\{1, \frac{1}{4}\}\} = \max\{-1, \frac{1}{4}\} = \frac{1}{4}$
  i.e. positive pixel value ($\frac{1}{2}$) becomes less positive ($\frac{1}{4}$)
- If $x = -\frac{1}{2}$, then $HA_x = \max\{-1, \min\{1, -\frac{1}{4}\}\} = \max\{-1, -\frac{1}{4}\} = -\frac{1}{4}$
  i.e. negative pixel value ($-\frac{1}{2}$) becomes less negative ($-\frac{1}{4}$)

2.3 Related work

This section is a compilation of scientific publications related to the topics we address in this study. The first section, mentions applications of XAI algorithms in domains like Finance, Autonomous-driving, Network Security and especially Medicine. In the end, it presents publications that address concerns about attribution maps methods. In the second section, we refer to works, that utilize XAI evaluation metrics for post-test model analysis. These works, share similarities but also distinct differences with our study. Most importantly, unlike our experiments, they do not incorporate the recently developed HiResCAM XAI algorithm and the HAAS evaluation metric and they do not refer to theory-based faithfulness results. However, among them, there seems to be a strong link between [38] and this study, as its results can be analysed in the context of our posed question as well. More specifically, in [38], though considering a different setting, the most faithful to the model algorithm yields the best metric results in almost all experiments and consequently suggests results that align well with the expected outcome of section 1.3 question.

XAI applications

XAI can bring large benefits to many domains that rely on AI black box systems. In the financial sector, in [39] the authors utilize SHAP values in order to analyse auditing

practices while in [40] both LIME and SHAP are used in real-time financial fraud detection tasks. Similarly, in the field of autonomous-driving the role of XAI is crucial. Based on a survey of the American National Highway Traffic Safety Administration (NHTSA) almost 94% of road accidents are due to human-related mistakes which in turn raises an urgent need for enhancing safety in such systems via XAI. This issue is examined in detail in [41] where the authors have compiled an overview of the current issues and future research directions. Likewise, XAI applications can be found in cybersecurity [42] and 6G-networks [43], [44] publications under the umbrella of providing security and privacy mechanisms to large project architectures.

As already mentioned in this study we are going to focus on medical scenarios. The majority of the literature uses image data and takes place in a transfer learning setting of pre-trained architectures combined with attribution based methods due to their ease of use. For publications that consider tabular data, one may indicatively consider: [45] using XGBoost and SHAP for Covid prediction and [46] using Random Forest and SHAP for Acute Myocardial Infarction (AMI) prediction. However, in the interest of presenting content closer to the content of this study we may not discuss them here. A detailed review for XAI applications on this kind of data can be found at [47]. On the other hand, when it comes to image data XAI applications the interested reader might refer, in the first place, at [10] where the authors list in Table 2 multiple research publications according to the anatomical location described in the data images. The locations are Bladder Brain, Breast, Cardiovascular, Chest, Dental Eye, Female reproductive system, Gastrointestinal, Lymph nodes, Musculoskeletal, Prostate, Skin, Skull and Thyroid. Each publication falls into one of the aforementioned location categories and is accompanied with the image modality (X-ray, MRI, Histology, Computed Tomography, Optical Coherence Tomography, Dermatoscopy, Endoscopy Ultrasound, Fundus Photography) and the main attribution map method which is used. After going through the list one immediately observes that CAM [31] and Grad-CAM are the most common choices that researchers use for their analysis. This showcases the popularity of the CAM-based algorithms in this field but also highlights the need for refining their usability and applicability as much as possible.

HiResCAM was introduced in 2022 and as a result it is not included in the above references. Until now, among the publications utilizing the algorithm, two of them come from the medical field and in both the implemented model has the structure *Conv - Flatten - Class Scores* which is optimal for HiResCAM as explained in section 1.3. In [37], the HiResCAM authors use a model named AxialNet to classify multiple abnormality types in 3D chest CT volumes while in [48] the algorithm is used in the context of Endoscopic images.

In the end of this short subsection, we believe that it would be beneficial to discuss works that focus on the inevitable concerns that come with attribution map methods

and examine how they can be manipulated. In [49], the authors focus on the importance of the *input invariance* property and demonstrate examples where it is possible to slightly shift the model input by a constant value such that the prediction and learned weights are not affected while the attribution map considerably changes. Based on their experiments, methods like Integrated Gradients, Gradient * Input and SmoothGrad are prone to this behaviour while Grad-CAM was not tested. Likewise, the Sanity Checks of [36] propose two artificially randomized experiments in order to address reliability issues for attribution maps. One the one hand, the *model parameter randomization test* focuses on the connection between the learned weights and the attribution maps by comparing the attribution maps of a trained and a randomly initialized network of the same architecture. On the other hand, the *data randomization test* focuses on the connection between the data labels and the attribution maps by comparing the attribution maps of a model trained on the correctly labelled data with those of a model of the same architecture trained on a copy of the original dataset with randomly permuted labels. Ideally, a trustworthy attribution method should be *sensitive* to both tests. In their experiments the authors implement various methods and rely on specific image examples to reach conclusions (there is no evaluation metric included). Grad-CAM shows promising consistency in both sanity tests while Integrated Gradients, Gradient * Input and SmoothGrad yield vague and less promising results.

XAI Evaluation metrics

The AOPC and Max-Sensitivity scores have been used in many real world applications in order to describe the quality of an explanation in terms of a fixed value. In the medical setting, one could refer to [50] to see an example of AOPC applied to multiple attribution maps produced by pre-trained models and ultrasound images of fetal heads. It is worth mentioning that for the purposes of this study the authors had to adapt the class-defined AOPC formula to the regression setting as well. Likewise, in [51], the authors employ a DL ensemble architecture trained on tabular datasets, such as BCW [52] and MIMIC-V [53], and compare the quality of SHAP explanations for both the baseline models and the aggregated one, in terms of the Max-Sensitivity score. On the other hand, in [54], the authors train image input models to facilitate the automated fiber placement production for tasks related to the aviation domain. They use Max-Sensitivity to quantify attributions and argue in favor of the Smooth IG method as the XAI algorithm to explain their model. This task, though not so typical as the rest described in this chapter, showcases the vast application ground of XAI and evaluation metrics and the plethora of benefits they can provide us with.

As far as the HAAS score is concerned, this metric is introduced in [27] as an alternative to the classic AOPC score. Inevitably, the experiments presented in the paper focus on a comparison between the two techniques over multiple classic datasets such as MNIST

[55], Cifar-10 [56], STL-10 [57] and ImageNet [58]. Furthermore, in their research, the authors argue in favour of creating evaluation schemes that are machine-centric and detect the best performing XAI algorithm regardless of the dataset. HAAS passes their tests towards this direction while AOPC does not.

Last but most importantly, we highlight the experiments conducted in [38]. In this paper, the authors use Earth image data collected by remote satellite sensors (BigEarthNet [59] and SEN12MS [60] datasets) and train DenseNet121 [33] and ResNet50 [32] models for their classification tasks. They implement many attribution map methods and quantify the maps via both the Max-Sensitivity and Area Under the MoRF curve (which is analogous to AOPC - Area Over the MoRF curve) scores. In Table 3 of the paper, they summarize the metric results and one can immediately observe that Grad-CAM is the optimal for both datasets with respect to Max-Sensitivity and the optimal for SEN12MS and second optimal for BigEarthNet with respect to the Area Under the MoRF curve. This is important for our study because, as we prove in chapter 3, if the models have the standard *Conv - GAP - Class Scores* architecture (as DenseNets and ResNets do) and gradients are computed with respect to the last convolutional layer, the Grad-CAM attributions directly reflect the class score and consequently are faithful to the model. In other words, among all Table 3 attribution map methods, the XAI metrics favour the attribution method that is faithful to the model (i.e. Grad-CAM). Likewise, our study conducts experiments testing an analogous case, as presented in section 1.3.

3 Methodology

In this chapter we adopt a mathematical perspective to formulate the relationships between Grad-CAM and HiResCAM in different CNN settings. The main focus of this chapter lies in Proposition 4 where we prove that if the network structure is *CNN - Flatten - Class Scores* and the gradients are computed with respect to the last convolutional layer then only HiResCAM produces faithful attribution maps, as it directly reflects the class score calculation. This theoretical result sets the motivational ground for our study and is investigated in chapter 4 by calculating the quality of attribution maps via the evaluation metrics of section 2.2 over multiple medical datasets. Finally, Table 1 summarizes the relationship between Grad-CAM and HiResCAM in the different CNN settings.

3.1 CNNs of the CAM architecture

In this section we consider CNNs of the CAM architecture, like the ones described in Figure 2, and first show that the CAM algorithm produces faithful explanations that directly contribute to the class score. We further prove that Grad-CAM and HiResCAM when applied in this setting collapse to CAM and as a result produce faithful explanations as well.

We denote:

- $\{A^f\}_{f=1,2,..,F}$ the feature maps of the last convolutional layer, where $A^f \in \mathbb{R}^{D_1 * D_2}$, $D_1, D_2 \in \mathbb{N}$ the pixel length and height of the feature map
- s the fully connected layer, given by:

  $$s = WA + b$$

  where $s \in \mathbb{R}^M$ are the raw scores of the M classes, $W \in \mathbb{R}^{M*F}$ is the weight matrix and $A \in \mathbb{R}^{F*1}$ is the GAP layer with values as in equation (1).

In this setting, the class scores become:

$$\begin{aligned} s^m &= \sum_f w_m^f a^f + b^m \\ &\overset{(1)}{=} \sum_f w_m^f \left(\frac{1}{D_1 D_2} \sum_{ij} A_{ij}^f \right) + b^m \end{aligned} \tag{11}$$

Proposition 1. *For CNNs of the CAM architecture the CAM explanations reflect the class score calculation.*

Proof. By calculating the class score we have:

$$\begin{aligned} s^m &\overset{(11)}{=} \frac{1}{D_1 D_2} \sum_f \sum_{ij} w_m^f A_{ij}^f + b^m \\ &= \frac{1}{D_1 D_2} \sum_{ij} \left(\sum_f w_m^f A_{ij}^f \right) + b^m \\ &= \frac{1}{D_1 D_2} \sum_{ij} \left(\sum_f w_m^f A^f \right)_{ij} + b^m \\ &\overset{(2)}{=} \frac{1}{D_1 D_2} \sum_{ij} \left(\mathcal{A}_m^{\text{CAM}} \right)_{ij} + b^m \end{aligned}$$

□

By taking derivatives in (11) we compute:

$$\frac{\partial s^m}{\partial A_{ij}^f} = w_m^f \frac{1}{D_1 D_2} \qquad \text{for } i = 1, 2, .., D_1 \text{ and } j = 1, 2, .., D_2 \tag{12}$$

$$\Leftrightarrow \quad \frac{\partial s^m}{\partial A^f} = w_m^f \frac{1}{D_1 D_2} \mathbb{1}_{D_1 * D_2} \qquad \text{in matrix form} \tag{13}$$

which are used in the Propositions 2 and 3 respectively. We note that $\mathbb{1}_{D_1 * D_2}$ stands for the 2-dimensional $D_1 * D_2$ matrix with the value 1 in all entries.

Proposition 2. *Grad-CAM is a generalization of CAM*

Proof.

$$\begin{aligned} \mathcal{A}_m^{\text{Grad-CAM}} &\overset{(3)}{\underset{(4)}{=}} \sum_f \left(\frac{1}{D_1 D_2} \sum_{ij} \frac{\partial s^m}{\partial A_{ij}^f} \right) A^f \\ &\overset{(12)}{=} \frac{1}{D_1 D_2} \sum_f \left(\sum_{ij} w_m^f \frac{1}{D_1 D_2} \right) A^f \\ &= \frac{1}{D_1 D_2} \sum_f \left(w_m^f \frac{1}{D_1 D_2} D_1 D_2 \right) A^f \end{aligned}$$

$$= \frac{1}{D_1 D_2} \sum_f w_m^f A^f$$
$$\overset{(2)}{=} \frac{1}{D_1 D_2} \mathcal{A}_m^{\text{CAM}}$$

Thus both methods yield identical explanations. The constant $\frac{1}{D_1 D_2}$ disappears in a subsequent normalization step. □

Proposition 3. *HiResCAM is a generalization of CAM*

Proof.

$$\mathcal{A}_m^{\text{HiResCAM}} \overset{(5)}{=} \sum_f \frac{\partial s^m}{\partial A^f} \odot A^f$$
$$\overset{(13)}{=} \sum_f \left(w_m^f \frac{1}{D_1 D_2} \mathbb{1}_{D_1 * D_2} \right) \odot A^f$$
$$= \frac{1}{D_1 D_2} \sum_f w_m^f A^f$$
$$\overset{(2)}{=} \frac{1}{D_1 D_2} \mathcal{A}_m^{\text{CAM}}$$

Thus both methods yield identical explanations. The constant $\frac{1}{D_1 D_2}$ disappears in a subsequent normalization step. □

3.2 CNNS ending in one fully connected layer

We consider CNN architectures ending in one fully connected layer (there is no GAP layer), as per Figure 7 below. In this setting Grad-CAM and HiResCAM no longer collapse to CAM and in Proposition 4 we present a mathematical explanation of how HiResCAM highlights locations (pixels) that increase the class score while Grad-CAM fails to do so, when we consider explanations with respect to the last convolutional layer. This property suggests that HiResCAM calculates maps that are more faithful to the model in the sense that they describe more accurately the locations that the model identifies the class.

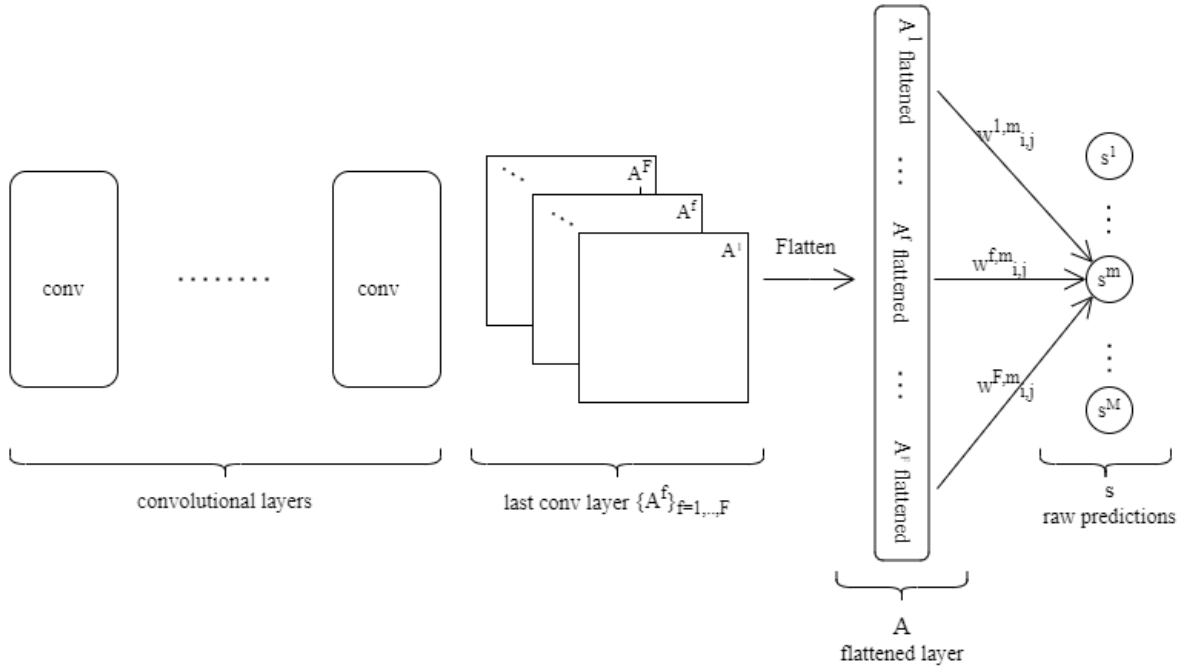


Figure 7: CNN ending in one fully connected layer

Proposition 4. *For CNNs ending in one fully connected layer if we compute the gradients with respect to the last convolutional layer, then HiResCAM is faithful to the model and highlights the locations that increase the class score. On the other hand, Grad-CAM fails to guarantee analogous behaviour.*

Proof. We denote:

- $\{A^f\}_{f=1,2,..,F}$ the feature maps of the last convolutional layer, where $A^f \in \mathbb{R}^{D_1 * D_2}$, $D_1, D_2 \in \mathbb{N}$ the pixel length and height of the feature map
- s the fully connected layer, given by:

$$s = WA + b$$

where $s \in \mathbb{R}^M$ are the raw scores of the M classes, $W \in \mathbb{R}^{M * FD_1D_2}$ is the weight matrix and $A \in \mathbb{R}^{FD_1D_2 * 1}$ is the flattened version of the last convolutional layer $\{A^f\}_{f=1,2,..,F}$.

In this setting, the class scores become:

$$
\begin{aligned}
s^m &= W^m A + b^m \qquad \text{,where } W^m \in \mathbb{R}^{1*FD_1D_2} \\
&= \sum_{f,i,j} W_{ij}^{f,m} A_{ij}^f + b^m \qquad \text{,where } W^{f,m} \in \mathbb{R}^{D_1*D_2} \qquad (14) \\
&= \sum_{i,j} \left(\sum_f W_{ij}^{f,m} A_{ij}^f \right) + b^m \\
&= \sum_{i,j} \left(\sum_f W^{f,m} \odot A^f \right)_{ij} + b^m \qquad (15)
\end{aligned}
$$

and by taking derivatives in (15) we compute:

$$
\frac{\partial s^m}{\partial A_{ij}^f} = W_{ij}^{f,m}, \quad \text{for } i = 1, 2, .., D_1 \text{ and } j = 1, 2, .., D_2 \qquad (16)
$$

$$
\Leftrightarrow \quad \frac{\partial s^m}{\partial A^f} = W^{f,m} \quad \text{in matrix form} \qquad (17)
$$

As a result, the Grad-CAM and HiResCAM algorithms take the following form in terms of the weights:

$$
\begin{aligned}
\mathcal{A}_m^{\text{Grad-CAM}} &\overset{(3)}{=} \sum_f \left(\frac{1}{D_1 D_2} \sum_{ij} \frac{\partial s^m}{\partial A_{ij}^f} \right) A^f \\
&\overset{(16)}{=} \sum_f \left(\frac{1}{D_1 D_2} \sum_{ij} W_{ij}^{f,m} \right) A^f \qquad (18)
\end{aligned}
$$

and

$$
\begin{aligned}
\mathcal{A}_m^{\text{HiResCAM}} &\overset{(5)}{=} \sum_f \frac{\partial s^m}{\partial A^f} \odot A^f \\
&\overset{(17)}{=} \sum_f W^{f,m} \odot A^f \qquad (19)
\end{aligned}
$$

Finally, combining (15) and (19) we get that

$$
s^m = \sum_{i,j} \left(\mathcal{A}_m^{\text{HiResCAM}} \right)_{i,j} + b^m \qquad (20)
$$

which shows that HiResCAM highlights the image locations which increase the class scores.

On the other hand, we note that it is not possible to plug equation (15) into (18) in a similar way, as quantities

$$\left(\frac{1}{D_1 D_2} \sum_{ij} W_{ij}^{f,m} \right) A^f \quad \& \quad W^{f,m} \odot A^f$$

are not equal in principle. This shows that Grad-CAM does not guarantee to highlight the class important locations.

□

Finally, for this setting, we note that if the gradients are calculated with respect to some other layer then there is no direct contribution to the class score for HiResCAM as well. This case along with the rest possible ones is included in Table 1 below, which shows a summary of the theory we have seen so far and shows the motivation for the topic of this study.

Table 1: Grad-CAM vs HiResCAM - Theory summary

CNN structure	Gradients wrt last Conv layer	Grad-CAM vs HiResCAM
Conv - GAP - Class Scores	yes	Equivalent & Both contribute to the class score
	no	Not equivalent & None contributes to the class score
Conv - Flatten - Class Scores	yes	Not equivalent. Only HiResCAM contributes to class score
	no	Not equivalent & None contributes to the class score
Conv - GAP/Flatten - Dense - Class Scores	yes	Not equivalent & None contributes to the class score
	no	

4 Experimental Results

This chapter focuses on the experiments designed to quantify the difference between Grad-CAM and HiResCAM attribution maps. For the purposes of our study we consider attribution maps coming from ResNet and VGG models which are trained on medical datasets. A brief introduction to each dataset is included in section 4.1, while in 4.2 we describe all steps leading to the construction of the final models; such as data preprocessing, customizing the models to the *Conv - Flatten - Class scores* architecture, training and testing. The next three sections of the chapter are dedicated to the attribution map evaluation metrics results: section 4.3 for AOPC, section 4.4 for Max Sensitivity and section 4.5 for HAAS, while finally section 4.6 contains plots that describe how the evaluation metrics results vary with respect to the models' balanced accuracy scores.

4.1 Datasets

In this section we present the medical datasets used in this study: CRC [61], Covid-19 Database [62], HAM10000 [63] and BreakHis [64]. All datasets are publically available and the links are given in the reference section.

We note that they all present class imbalanced distributions which is quite common in medical datasets. This phenomenon is more apparent in HAM10000 and BreakHis datasets and results in making the learning procedure difficult, especially compared to the other two datasets. For this reason, the BreakHis dataset is treated as a binary problem as this allows to build more robust models. On the other hand, dealing with HAM10000 as binary does not significantly improve the learned classifiers because the classes remain quite imbalanced. Thus this dataset is addressed via its multi-class original version.

CRC

The CRC dataset, [61], was introduced in 2018 and has been widely used in cancer research to investigate the patterns characterizing colorectal cancer (CRC). It comes into the forms of "NCT-CRC-HE-100K" and "CRC-VAL-HE-7K" datasets which consist of 100,000 and 7,180 non-overlapping samples and are used as training and test sets respectively. The tissue classes are: Adipose (ADI), background (BACK), debris (DEB), lymphocytes (LYM), mucus (MUC), smooth muscle (MUS), normal colon mucosa (NORM), cancer-associated stroma (STR), colorectal adenocarcinoma epithelium (TUM).

The class distribution along with data samples are presented in the Figures 8 and 9 below:

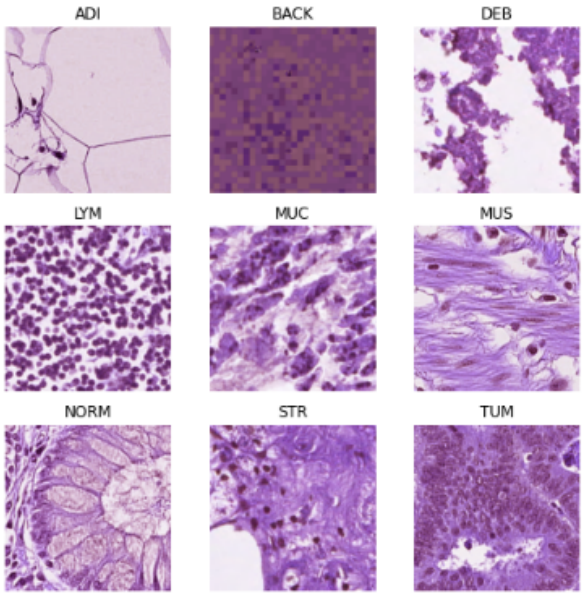
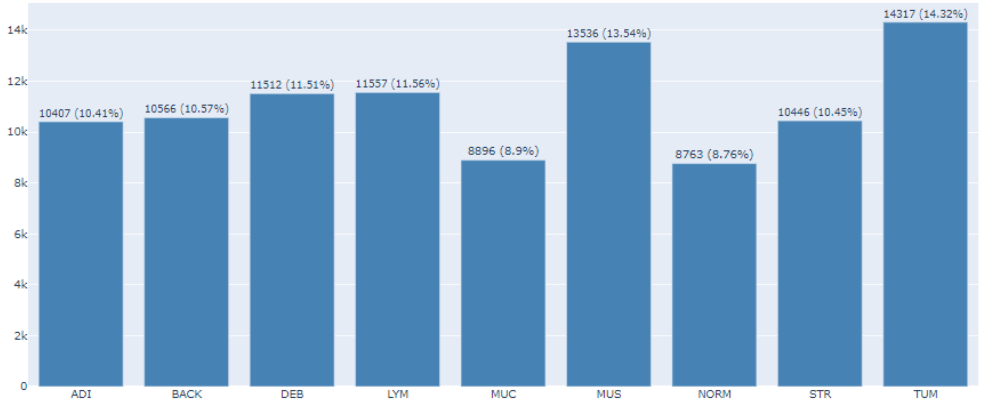
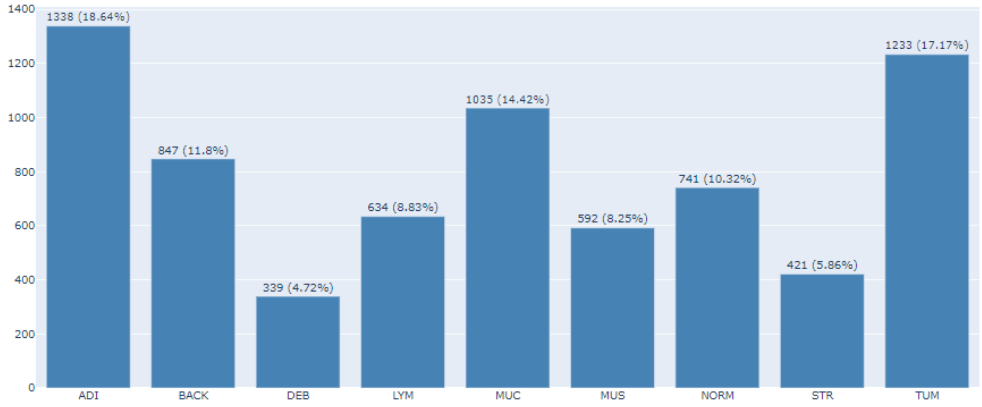


Figure 8: CRC - Data samples



(a) NCT-CRC-HE-100K



(b) CRC-VAL-HE-7K

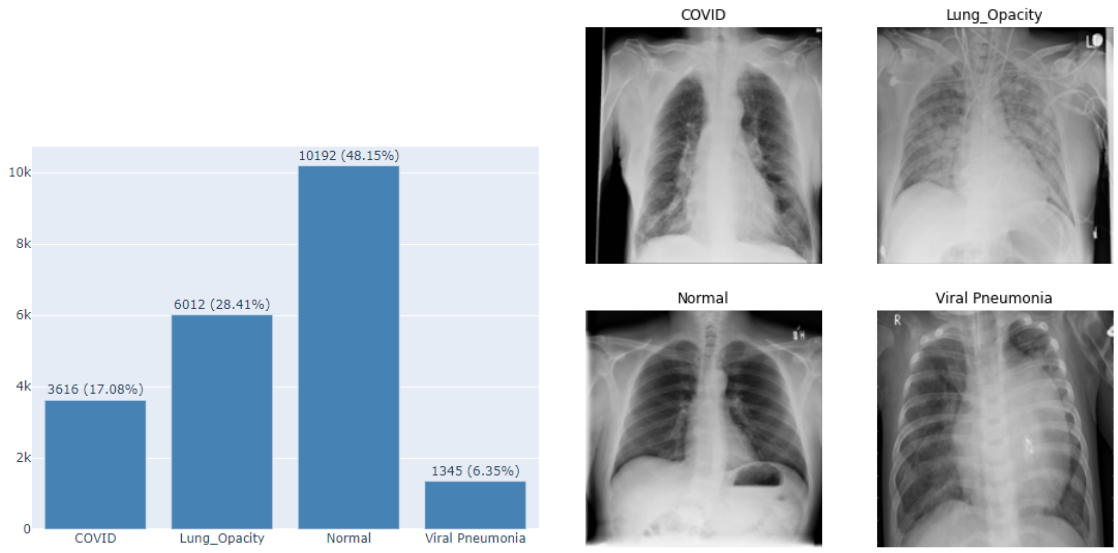
Figure 9: CRC - Class distribution

The interested reader might find many publications focusing on the classification task. In [65] and [66] the authors use single pre-trained architectures while in [67] they build an ensemble of pre-trained models. At the same time, in [68] the semantic segmentation task is addressed by using UNet and SegNet models.

Covid-19 Radiography Database

The Covid-19 Radiography Database, [62], was created in an attempt to introduce the Machine Learning community to this recent disease and build architectures that could separate it from other well-known chest related diseases such as Lung Opacity and Viral Pneumonia. The authors combined several public X-Ray databases and collected separate images from published articles to launch the first version of the dataset in 2020. Since then it has been enriched multiple times and has been used in many scientific publications. Some famous examples include [69] and [70] where pre-trained models are used to build image classifiers and semantic segmentation models respectively.

For the purposes of this thesis we have used the most up-to-date release. The class distribution is described in the Figure 10 along with a sample case per class.



(a) Class distribution (b) Data samples

Figure 10: Covid-19 Radiography Database

HAM10000

HAM10000, [63], is a dataset of 10,015 dermatoscopic images of pigmented skin lesions developed to facilitate the research in computer-aided diagnosis. The images are

labelled by dermatologists into one of the following categories: Melanocytic nevi (nv), Melanoma (mel), Benign keratosis-like lesions (bkl), Basal cell carcinoma (bcc), Actinic keratoses (akiec), Vascular lesions (vasc) and Dermatofibroma (df). The class distribution along with data samples are presented in the Figures 11 and 12 below:

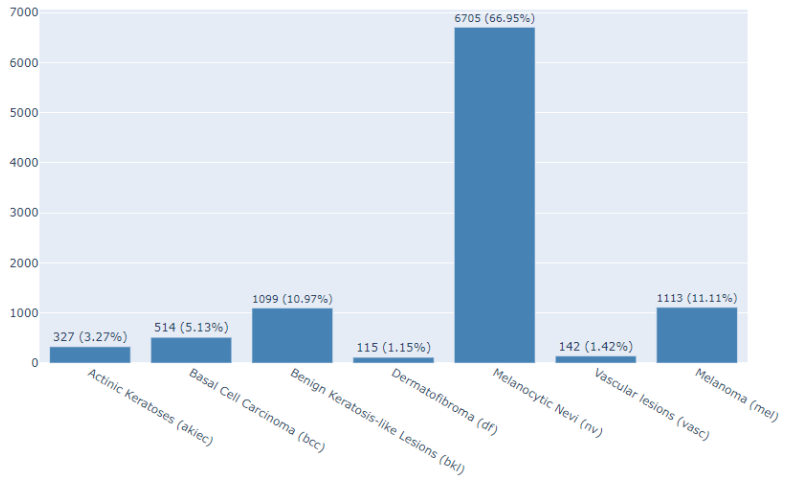


Figure 11: HAM10000 Dataset - Class distribution

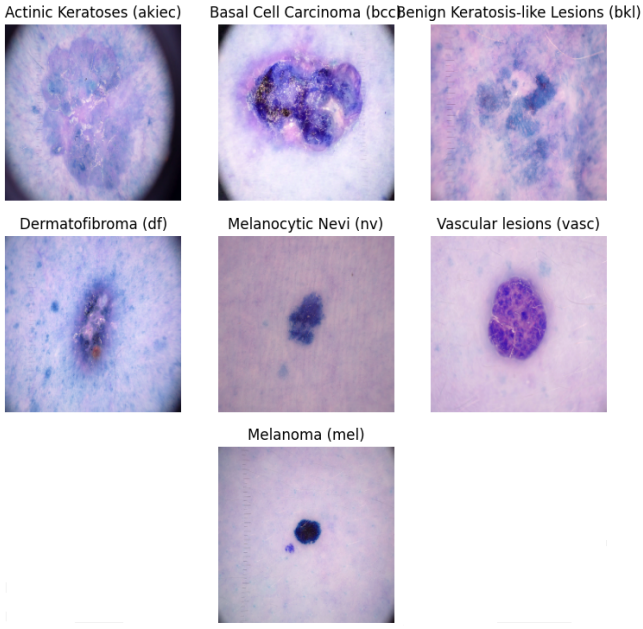


Figure 12: HAM10000 Dataset - Data samples

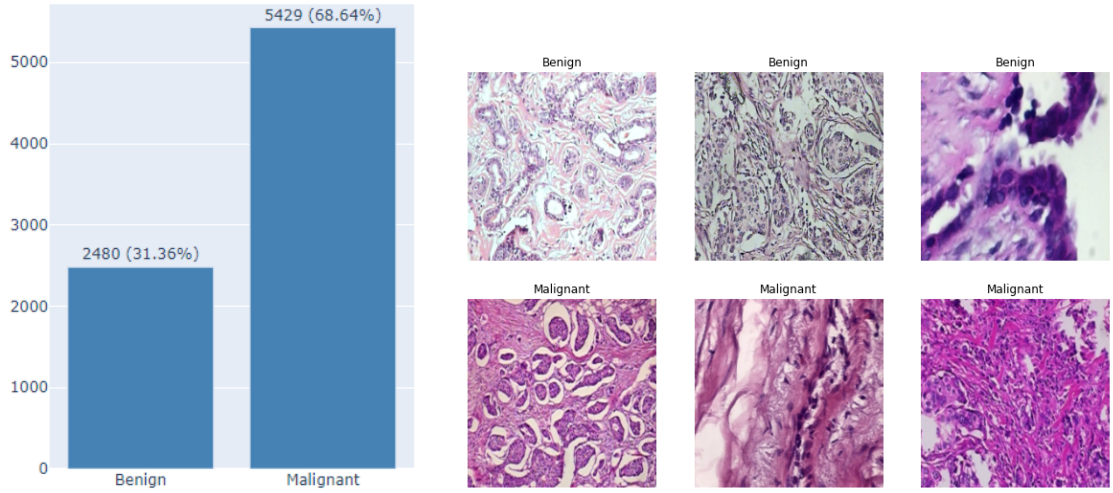
The dataset is also known for the ISIC (International Skin Imaging Collaboration) 2018 challenge, which was hosted on Kaggle. The competition aimed to encourage the devel-

opment of automated algorithms for the classification of skin lesions, using the HAM10000 dataset as the training and validation dataset for the sub-tasks of multi-class and binary (benign vs malignant) problems. The results of the challenge are summarized in [71].

BreakHis

The Breast Cancer Histopathological Image Classification (a.k.a. BreakHis) dataset, [64] and [72], consists of 7909 microscopic images of breast tumor tissue collected from 82 patients on different magnifying factors (40X, 100X, 200X, and 400X). It contains 2,480 benign samples classified as adenosis (A), fibroadenoma (F), phyllodes tumor (PT), and tubular adenona (TA) and 5,429 malignant samples (breast tumor) classified as carcinoma (DC), lobular carcinoma (LC), mucinous carcinoma (MC) and papillary carcinoma (PC).

For the purposes of this thesis we consider the binary problem (benign vs malignant) whose class distribution and samples are summarized in Figure 13 below:



(a) Class distribution (b) Data samples

Figure 13: BreakHis Dataset

The interested reader might refer at first to [73] where the authors present a summary of the different approaches used for modelling this dataset until 2020; such as magnification level independent and specific approaches for both the binary and the multi-class problems. Further to this, in [67] and [74] an ensemle approach is adopted for dealing with the complexity of the data.

4.2 Models

Data preprocessing

In this section we firstly discuss the issue of *rescaled* samples which are present in two of the datasets. The first one is the HAM10000 dataset and we already know from section 4.1 that it consists of 10,015 images. These images come from 7,470 unique skin lesions some of which are included in the dataset in different rescaled versions and in total add up to 10,015 different images. Once we download the dataset we have access to the HAM10000_metadata.csv file which allows us to connect each lesion to the image(s) it contributes to the dataset. The second one is the BreakHis dataset where as explained in 4.1 a breast tissue image contributes samples at 40X, 100X, 200X, and 400X magnification factors. The data is downloaded in a folder format that directly separates the images among the different factors.

The rescaled images issue suggests the need of adopting a skin-lesion and breast-image *independent* approach when constructing the training-validation-test sets of HAM10000 and BreakHis datasets respectively. In that way we ensure that the model performance will not be accessed on rescaled variations of training datapoints and as a result we avoid introducing extra *bias* to the final model.

Table 2 contains a summary of the training-validation-test datasets constructed for each medical dataset. We note that in CRC, Covid-19 Database and HAM10000 a common *stratified* sampling approach is implemented. On the other hand, in BreakHis we allocated the datapoints manually to the training, validation and test datasets and as a result the allocated class percentages are not exactly the same for each class. As we see in the last line of the table they are quite close though.

Table 2: Training - Validation - Test sets

Dataset	Training - Validation - Test set
CRC	80% of NCT-100K - 20% of NCT-100K - CRC-VAL-HE-7K
Covid-19 Database	80% - 10% - 10%
HAM10000*	60% - 20% - 20%
BreakHis**	Benign 58% - 20% - 22%, Malignant 63% - 18% - 19%

Note: * skin lesion independent sets, ** tissue image independent sets

Finally we note that all data were resized to the pre-trained models recommended input size, $224 * 224$, and normalized to $[-1, 1]$ valued tensors. In addition, the training datasets were augmented with vertical and horizontal flips and random rotations.

Adjusting the pre-trained models to the required architecture

For the purposes of our experiments we used pre-trained ResNet34, ResNet50 and VGG19 architectures. All networks were customized to the *Conv - Flatten - Class Scores* structure of Figure 7 (chapter 3) such that HiResCAM has faithful behaviour when calculated with respect to the last convolutional layer (recall result of Proposition 4).

The models were imported by PyTorch's *torchvision* class. In all of them, the convolutional part does not end in a Conv2d layer and is followed by a GAP layer. Thus in order to transform them in the aforementioned structure one has to act as follows:

- VGG19: remove ReLU and MaxPool2d layers at the end of the convolutional part
- ResNet34: remove BatchNorm2d at the end of the convolutional part
- ResNet50: remove BatchNorm2d and ReLU at the end of the convolutional part
- All three networks: replace GAP by Flatten and adjust the number of nodes

Training and Testing

For each dataset a ResNet and a VGG19 custom model was trained, resulting in 8 models in total. Each model was trained with a weighted Cross Entropy loss function to address the class imbalance issue and Adam optimizer. The *trainable (unfrozen) layers* for each pre-trained architecture were us follows, by using the *torchvision layer names*:

- CRC dataset - ResNet34: *layer3*, *layer4*
- CRC dataset - VGG19: all *feature* layers numbered from *19* to *34*
- Covid-19 dataset - ResNet34: *layer1*, *layer2*, *layer3*, *layer4*
- Covid-19 dataset - VGG19: all *feature* layers numbered from *21* to *34*
- HAM10000 dataset - ResNet50: *layer4*
- HAM10000 dataset - VGG19: all *feature* layers numbered from *12* to *34*
- BreakHis dataset - ResNet50: *None*
- BreakHis dataset - VGG19: *None*

Furthermore, Table 3 below shows the training configuration that eventually prevailed for each model among many others that were tested. It is worth to mention that along with PyTorch's build-in regularizers, Weight decay (L2 norm) and Learning Rate Scheduler, we also implemented a custom written Early Stopping technique to control training when performance was no longer improving. As we observe in the table, it is controlled by the *patience* parameter and stops training as long as validation loss is no longer decreasing and the mean validation recall of non-normal (i.e. usually "disease" related) classes is no longer increasing after *patience* number of epochs.

The choice of allowing the user to control the classes whose recall will be monitored during training is motivated by the fact that the Covid-19 dataset includes a *Normal* class which is not as important as the rest "disease" related classes. It gives the option to focus only on the classes that are important for the human life. In this study *we have monitored all diseases in the CRC, HAM10000 and BreakHis datasets, regardless if they are fatal or not*. We have skipped monitoring only the training of the *Normal* class of the Covid-19 dataset. However, is it essential to note that one could consider monitoring only fatal diseases. If this is desired, then the interested user is encouraged to go through the documentation of the supporting *training_loop.py* file and use the attribute *labels_of_normal_classes* accordingly.

Table 3: Training Configurations

	CRC		Covid-19		HAM10000		BreakHis	
	ResNet34	VGG19	ResNet34	VGG19	ResNet50	VGG19	ResNet50	VGG19
Trainable layers	layer3-layer4	19-34	layer1-layer4	21-34	layer4	12-34	None	None
Learning rate	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-5}	10^{-4}	10^{-4}	10^{-4}
Weight decay (L2)	None	None	None	None	10^{-1}	10^{-1}	None	None
Scheduler	None	None	None	None	None	step=1, γ=0.9	step=1, γ=0.5	step=1, γ=0.5
Batch size	256	256	64	128	256	128	16	16
Training Epochs	22	24	25	21	26	22	12	12
Patience	7	7	20	20	20	20	None	None

Finally, Table 4 summarizes the testing results for all produced models. We observe that the CRC and Covid-19 datasets yield well performing models while BreakHis decent results indicate that there is still room for improvement. HAM10000 seems to be the most challenging among four justifying its presence in the ISIC 2018 competition. As explained in [75], the models' performance in the competition was tested via the Mean AUC score and the top 10 performing scores were between 0.9461 and 0.949.

Table 4: Testing Results

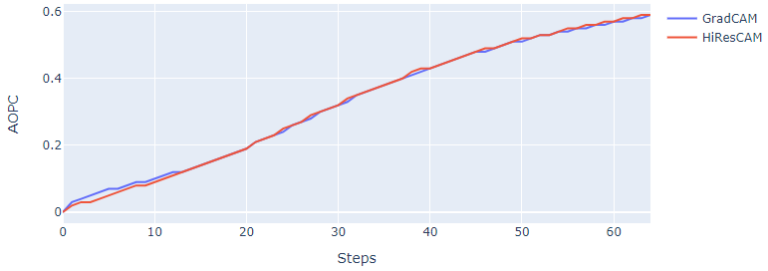
	CRC		Covid-19		HAM10000		BreakHis	
	ResNet34	VGG19	ResNet34	VGG19	ResNet50	VGG19	ResNet50	VGG19
Balanced Accuracy	0.89	0.94	0.97	0.95	0.69	0.73	0.87	0.84
Precision (weighted)	0.93	0.95	0.95	0.93	0.8	0.8	0.87	0.85
Recall (weighted)	0.91	0.94	0.95	0.93	0.71	0.67	0.85	0.81
F1 (weighted)	0.92	0.94	0.95	0.93	0.73	0.71	0.85	0.81
Mean AUC	0.993	0.997	0.995	0.992	0.934	0.938	0.942	0.932

4.3 AOPC results

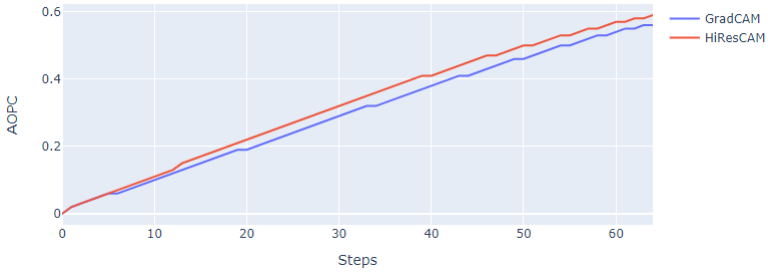
The implementation of a post-test metric such as the AOPC score can be proven a valuable tool to get insights on how well a learned model understands the problem's classes. One may refer to chapter 2 where we refer to cases that the use of AOPC enhanced the model's credibility. However, the implementation of the MoRF technique is also coupled with many critical decisions that should be made about its hyper-parameters. As theory of section 2.2.1 suggests, the heatmap H can be created at many different sizes, via different baseline functions g and different ways of perturbations.

In our experiments, in order to cover a variery of different scenarios, the $224*224$ Grad-CAM and HiResCAM attribution maps are perturbated by regions of size $56*56$, $28*28$, $21*21$ and $16*16$ resulting in heatmaps of size $4*4$, $8*8$, $11*11$ and $14*14$ respectively. In addition, per perturbation step, we replace the image pixels with re-sampled uniform noise g coming from the range $[-1, 1]$.

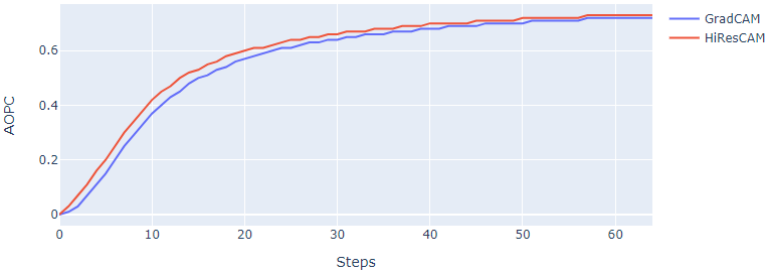
A summary of our findings is presented in the numeric scores of Table 5 and the AOPC graphs of Figure 14. As per analysis of section 2.2.1, we recall that large AOPC values suggest heatmaps of better quality. In the interest of space we have included only the $8*8$ heatmap graphs in this document. The rest graphs are included in the accompanying code material.



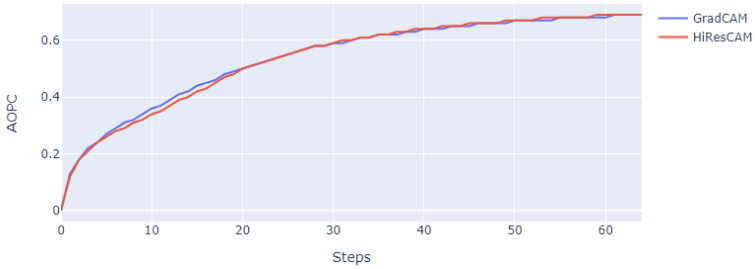
(a) CRC - ResNet34



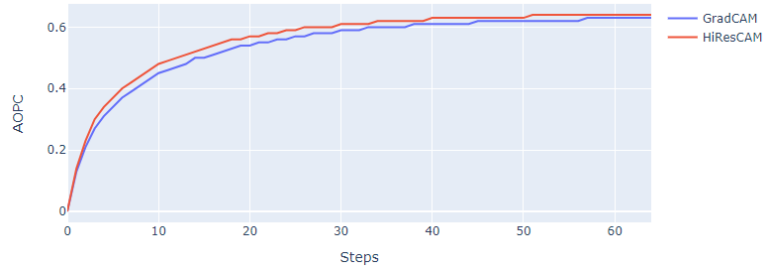
(b) CRC - VGG19



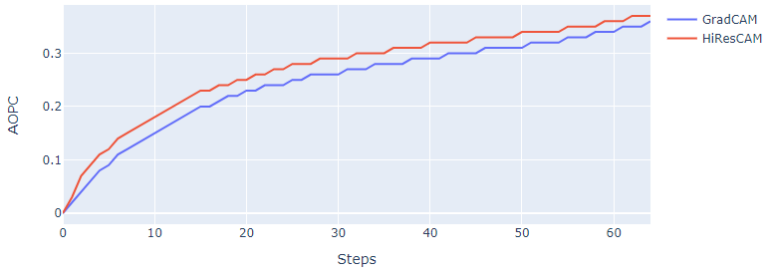
(c) Covid-19 - ResNet34



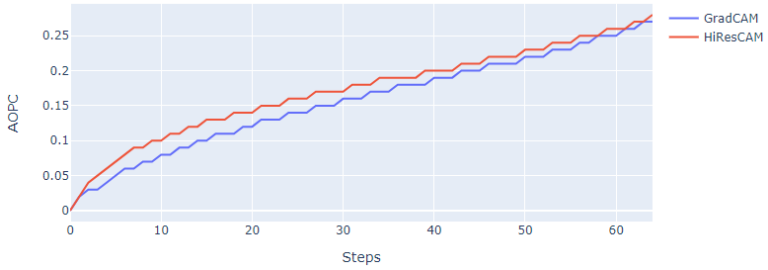
(d) Covid-19 - VGG19



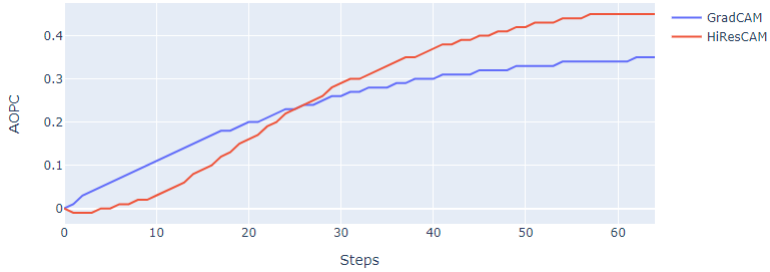
(e) HAM10000 - ResNet50



(f) HAM10000 - VGG19



(g) BreakHis - ResNet50



(h) BreakHis - VGG19

Figure 14: AOPC Graphs for Heatmaps 8*8 size

Table 5: AOPC Scores

			CRC		Covid-19		HAM10000		BreakHis	
			ResNet34	VGG19	ResNet34	VGG19	ResNet50	VGG19	ResNet50	VGG19
Heatmap size	4×4	Grad-CAM	0.57	0.5	0.71	0.64	0.57	0.35	0.32	0.35
		HiResCAM	0.57	**0.52**	**0.73**	**0.65**	**0.58**	**0.37**	**0.34**	**0.42**
	8×8	Grad-CAM	0.59	0.56	0.72	0.69	0.63	0.36	0.27	0.35
		HiResCAM	0.59	**0.59**	**0.73**	0.69	**0.64**	**0.37**	**0.28**	**0.45**
	11×11	Grad-CAM	0.6	0.59	0.73	0.71	0.65	0.37	**0.28**	0.35
		HiResCAM	0.6	**0.62**	**0.74**	0.71	**0.67**	**0.39**	0.27	**0.45**
	14×14	Grad-CAM	0.58	0.6	0.73	**0.7**	0.66	0.36	0.27	0.35
		HiResCAM	**0.6**	**0.62**	**0.74**	0.69	**0.68**	0.36	0.27	**0.48**

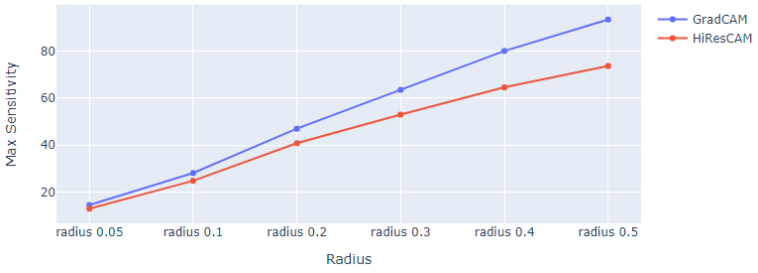
4.4 Max-Sensitivity results

The implementation of Max-Sensitivity score was based on formula (8) of section 2.2.2 and similarly to the AOPC score it is governed by a set of hyper-parameters. Following the online code provided by the authors in [76], one has to decide about the value of radius r which controls the noise amount that is added to the image pixels and the number of perturbed images that will be produced. Then by uniformly sampling random points in this radius they produce a set of perturbed versions of the original image whose attribution map features should be as close as possible to the features of the original image attribution map.

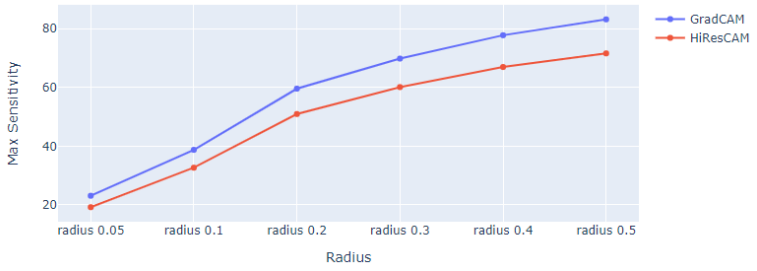
In our experiments we implemented the metric for different radius values and number of perturbated instances (also referred as "iterations" in the code file). The experiment configurations are summarized in Table 6 while in Figure 15 we have plotted the calculated score plots for all tested scenarios. As per analysis of section 2.2.2, we recall that low Max-Sensitivity values suggest heatmaps of better quality. The number of iterations was chosen based on the calculation time complexity and the available hardware resources. It seems natural to consider increasing iterations per increasing radii.

Table 6: Max-Sensitivity experiment configurations

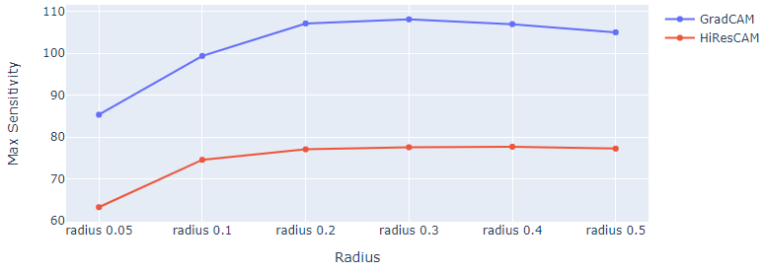
Radius	Iterations
0.05	20
0.1	20
0.2	30
0.3	30
0.4	40
0.5	40



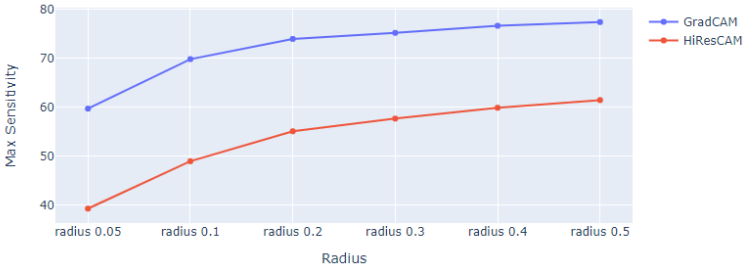
(a) CRC - ResNet34



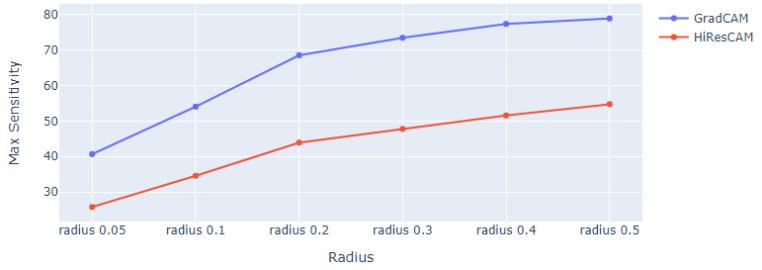
(b) CRC - VGG19



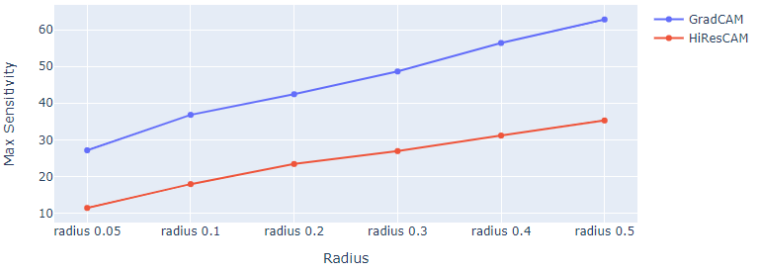
(c) Covid-19 - ResNet34



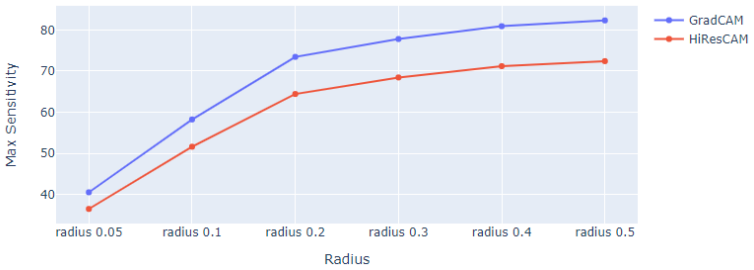
(d) Covid-19 - VGG19



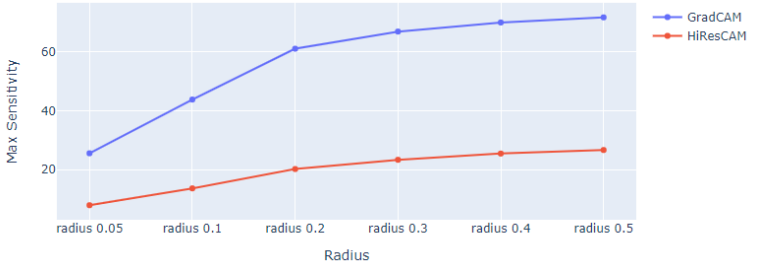
(e) HAM10000 - ResNet50



(f) HAM10000 - VGG19



(g) BreakHis - ResNet50



(h) BreakHis - VGG19

Figure 15: Max Sensitivity Results

4.5 HAAS results

The last part of our experiments is about the HAAS score. Being available only for a few months, there are no published experiments at the moment that utilize it. Thus our knowledge and expectations are limited to the theoretical setting presented in 2.2.3. We also recall

In contrast to AOPC and Max-Sensitivity, HAAS score is free of hyper-parameters. The only requirement for its implementation is to ensure that the test images are scaled to $[-1, 1]$. In our study, as explained in section 4.2, this step took place during the model pre-training steps for all datasets.

Table 7: HAAS Scores (Medical datasets)

	CRC		Covid-19		HAM10000		BreakHis	
	ResNet34	VGG19	ResNet34	VGG19	ResNet50	VGG19	ResNet50	VGG19
Grad-CAM	0.47	0.76	**0.86**	**0.89**	**0.831**	0.714	0.927	0.985
HiResCAM	**0.53**	**0.8**	0.67	0.84	0.83	**0.834**	**0.936**	**1.081**

Table 7 summarizes the results of our experiments. For this metric, we recall that ideally the score should be above 1 and as large as possible. However, our calculations over medical datasets suggested that almost all the times the metric did not value the models' explanations and could not yield helpful results to test our case between Grad-CAM and HiResCAM. In order to further explore if this is related to the nature of the dataset or not we also implemented it for the non-medical datasets used in the original HAAS publication, [27]. Per dataset, we trained 16 VGG19 architectures for different values of batch size, learning rate, scheduler and weight decay, as per Table 8, and recorded the findings for the HiResCAM attribution maps in Table 9. The content of both tables is extensively discussed in the next chapter. Finally, as suggested in Figure 4 of [27], the VGG19 architectures for Cifar-10 and STL-10 are customized to the input image size. Thus the network is cut such that the image does not collapse to a single number.

Table 8: HAAS - VGG19 configurations for non-medical datasets

	Cifar-10	STL-10	Imagenette*
Batch size	$32, 64, 128, 256$	$32, 128$	$32, 64, 128, 256$
Learning rate	10^{-4}, 10^{-6}	10^{-4}, 10^{-6}	10^{-4}, 10^{-6}
Scheduler	-	None, step=2 &γ=0.5	None, step=2 & γ=0.5
Weight decay (L2)	None, 10^{-2}	None, 10^{-1}	-

Note: * Used Imagenette instead of ImageNet for size reasons

Table 9: HAAS Scores (Non Medical datasets)

		Cifar-10	STL-10	Imagenette
		VGG19*	VGG19*	VGG19*
HiResCAM	Max HAAS Score	1.009	1.034	1.002
	Mean AUC	0.981	0.966	0.995
	Min HAAS Score	0.970	0.978	0.986
	Mean AUC	0.969	0.899	0.889

Note: * loop of 16 models for different batch size, learning rate, scheduler and weight decay

Finally, in Table 10 below, we put together the testing results of section 4.2 (Table 4) and XAI evaluation metrics results of sections 4.3, 4.4 and 4.5 (Table 5, Figure 15 and Table 7 respectively). Although they will be discussed in detail in chapter 5, at this point it is worth noticing that AOPC and Max-Sensitivity favour HiResCAM over Grad-CAM while HAAS gives scores that do not favour any algorithm. In the following calculations, note that for AOPC and Max-Sensitivity we calculate the *mean values* over the heatmap sizes and radii tested respectively.

Table 10: Summary of Test and Evaluation Metrics Results

		Bal. Accuracy	Mean AUC	Mean AOPC		Mean Max-Sens		HAAS	
				GradCAM	HiResCAM	GradCAM	HiResCAM	GradCAM	HiResCAM
CRC	ResNet34	0.89	0.993	0.585	**0.59**	54.37	**44.91**	0.47	**0.53**
	VGG19	0.94	0.997	0.563	**0.588**	58.7	**50.24**	0.76	**0.8**
Covid-19	ResNet34	0.97	0.995	0.723	**0.735**	101.99	**74.6**	**0.86**	0.67
	VGG19	0.95	0.992	0.685	0.685	72.14	**53.72**	**0.89**	0.84
HAM10k	ResNet50	0.69	0.934	0.628	**0.643**	65.51	**43.04**	**0.831**	0.83
	VGG19	0.73	0.938	0.36	**0.373**	45.68	**24.33**	0.714	**0.834**
BreakHis	ResNet50	0.87	0.942	0.285	**0.29**	68.88	**60.78**	0.927	**0.936**
	VGG19	0.84	0.932	0.35	**0.45**	56.46	**19.52**	0.985	**1.081**

4.6 Connecting accuracy and explainability results

In this short section we look at the extracted results from a slightly different perspective. We get motivation from the fact that HiResCAM yields attribution maps of better quality (as per Table 10 summary) and plot all HiResCAM expainability metrics results with respect to the models' balanced accuracy in order to discuss in section 5.5 whether models with better explainability properties have better accuracy as well. Our findings are presented in Figures 16, 17 and 18 below. We note that in Figures 16 and 17 there are model cases where the plotted points are less than the number of heatmap sizes and radii tested respectively; for instance BreakHis ResNet50 in Figure 16 and Covid19 ResNet34 in Figure 17. This is due to the fact that there are values that are either equal or close to each other and as a result they are not discernible in the scatter plot.

Figure 16: Balanced Accuracy vs AOPC

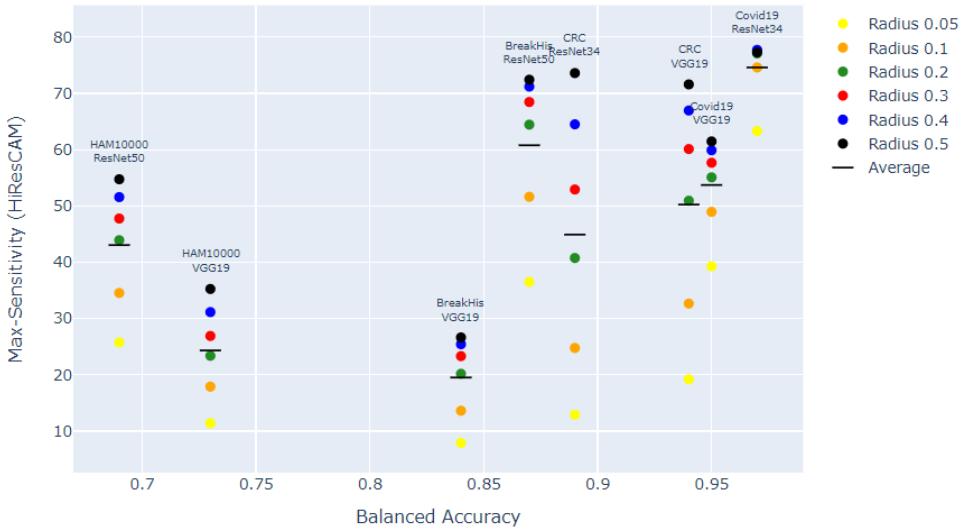


Figure 17: Balanced Accuracy vs Max-Sensitivity

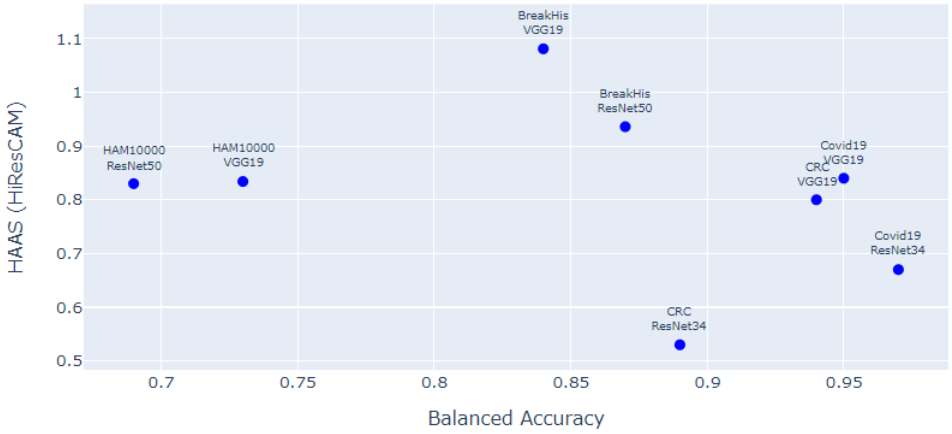


Figure 18: Balanced Accuracy vs HAAS

5 Discussion

The purpose of this chapter is to discuss the experimental results of chapter 4, in order to reach conclusions that enlighten the relationship between the Grad-CAM and HiResCAM in the setting of *Conv - Flatten - Class scores* architectures and while attributions are calculated with respect to the last convolutional layer of the network.

As we already know from chapter 3, the HiResCAM attribution maps preserve the effect of the gradients on a pixel level and directly contribute to the calculated class score. On the other hand, Grad-CAM considers average gradient effects for all feature map pixels and fails to produce maps that ensure analogous behaviour. As a result, based on theory, the HiResCAM attribution maps are more faithful to the model's actual viewpoint. This fact encourages us to investigate the relationship between Grad-CAM and HiResCAM attribution maps on a practical level and see how it is depicted through metrics that quantify the quality of attribution maps.

For this reason we designed experiments on real-world medical datasets and calculated metrics such as the well-known AOPC score along with the Max-Sensitivity and HAAS scores for Grad-CAM and HiResCAM attribution maps in many different scenarios. These metrics are designed to evaluate the quality of an attribution map, for a good performing model, with respect to many *logical* perspectives. For instance, the AOPC score favors maps which localize the model's viewpoint in a *small* area of the image while Max-Sensitivity praises maps which *do not deviate* a lot when the tested image is subject to small perturbations. Finally, the HAAS score suggests that an *accurate* attribution map should be able to tune the image pixels based on their calculated importance in such a way that correct predictions are preserved and incorrect ones are classified correctly.

5.1 Evaluation metrics comparison

Before discussing the experimental results, it is essential to consider some technical differences between the AOPC, Max-Sensitivity and HAAS scores *in general*. Notably, HAAS is the only metric among the three that is independent of noise or any baseline. On the other hand, AOPC requires tuning for the g function, which can be either random noise or a constant baseline, while Max-Sensitivity is estimated by drawing random noise samples for each image pixel. Additionally, as previously mentioned in section 2.2.3, HAAS is free of hyper-parameters and establishes a machine-centric deterministic score that is efficient to implement, making it a desirable metric from a theoretical perspective.

Further to the above, it is important to note that Max-Sensitivity and HAAS directly utilize all the attribution map pixel values into the score calculation. On the other hand,

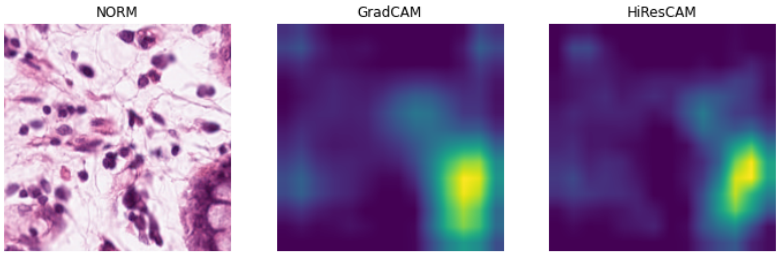
MoRF requires the pixel attribution maps to be transformed to a region-level map and as a result the value of an attribution pixel is *camouflaged* into the averaged region value.

Finally, time-wise speaking, based on our experiments, calculating HAAS score is quite faster than calculating AOPC and Max-Sensitivity score; regardless of the hyper-params configuration chosen by the user for the latter two metrics. Indeed for each test image, HAAS requires a simple transformation (to construct the HA image variant) and two forward passes (predictions). On the other hand, AOPC has to select random noise and calculate a forward pass for as many times as the chosen perturbation steps, while Max-Sensitivity computes multiple slightly perturbated variants of the image and computes the distances of the respective class attribution maps. In the supporting Google Colab notebooks, all experiments are timed and the results are printed in seconds. However, due the fact that Google Colab does not provide the same GPU every time we will not report the exact raw results. Instead, on an estimation basis, we can confidently say that HAAS is faster than AOPC and Max-Sensitivity.

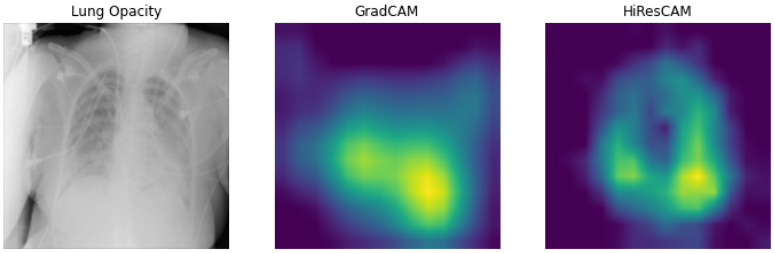
5.2 AOPC and Max-Sensitivity results

In this subsection we focus on the AOPC and Max-Sensitivity results as described in Table 5 (section 4.3) and Figure 15 (section 4.4) respectively. A summary with respect to their mean scores is also included in the end of section 4.5. They are discussed together because their analysis is rooted to the same reasoning.

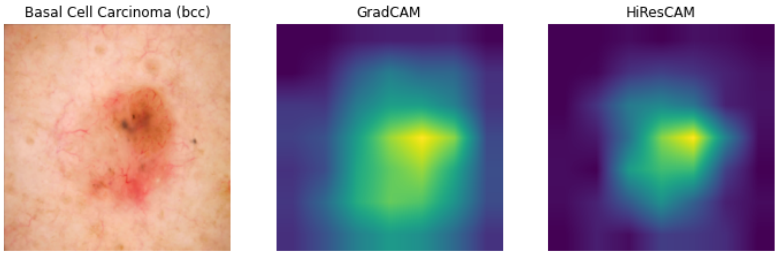
In regards to the AOPC results, we first observe that heatmaps with large region size $4*4$ favour HiResCAM in 7 out of 8 cases while as the size decreases the effect slightly fades. Indeed, heatmaps of $8 * 8$ size favour HiResCAM in 6 out of 8 cases while heatmaps of $11 * 11$ and $14 * 14$ in 5 out of 8 cases. Furthermore, for each model tested and taking into account all heatmap sizes, one observes that the metric either favors HiResCAM (ResNet34 CRC, VGG19 CRC, ResNet34 Covid-19 Database, ResNet50 HAM10000, VGG19 HAM10000, ResNet50 BreakHis, VGG19 BreakHis) or gives similar measurements among Grad-CAM and HiResCAM (VGG19 Covid-19 Database). Based on this, we conclude that HiResCAM attribution maps are at least as good (informative) as the Grad-CAM attribution maps. At the same time, in regards to the Max-Sensitivity results, one immediately observes that HiResCAM has always lower score for all models and radii tested.



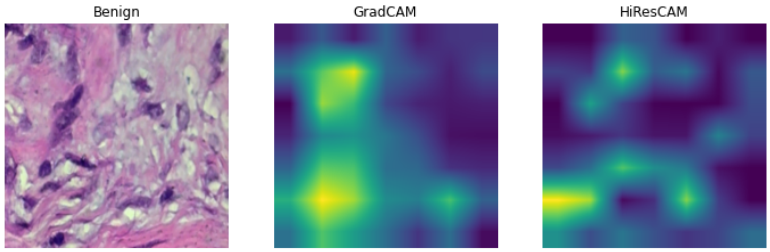
(a) CRC dataset - VGG19



(b) Covid-19 dataset - VGG19



(c) HAM10000 dataset - ResNet50



(d) BreakHis dataset - ResNet50

Figure 19: Grad-CAM and HiResCAM attribution maps

In an attempt to detect the factors that lead to the results in favour of HiResCAM it is logical to revisit the definitions of Grad-CAM (Definition 2) and HiResCAM (Definition 3). We observe that the two algorithms treat the gradients in a different way, as on the one hand Grad-CAM considers Gradient Averages (formula (4)) to weight each feature map while on the other hand HiResCAM weights each feature map on a pixel level by the respective gradient. This approach allows HiResCAM to directly contribute to the

calculated class score and also preserve the gradient effect, in terms of value and sign, on a pixel level. In turn, this leads to fine-grained high resolution HiResCAM attribution maps, in comparison to those produced by Grad-CAM where the attention area is larger and smoother due to the Gradient Averaging step.

From a visual perspective, this becomes apparent from the examples of Figure 19 which are taken from the experiments datasets. In all cases, even though the explanations look similar in terms of the regions of interest, the Grad-CAM ones occupy larger part of the image accompanied by smoother boundary.

As a result, when it comes to the AOPC score one may argue that the high resolution HiResCAM attribution map might capture better the details of the input image and provide more precise localization of the most discriminative regions, leading eventually to higher AOPC scores. Similarly, considering the Max-Sensitivity results, the high resolution HiResCAM maps present a resilient behaviour to small perturbations in the input image.

5.3 HAAS results

As far as the HAAS score is concerned, Table 7 (section 4.5) gives a summary of the values obtained on medical datasets. One can immediately see that it was quite common (15 out of 16 times) to get values below 1, which at first raises doubts about the quality of the Grad-CAM and HiResCAM attribution maps; implying that they cannot highlight the feature's importance for the model.

In an attempt to look deeper into these values one could make a few remarks that could motivate some further study:

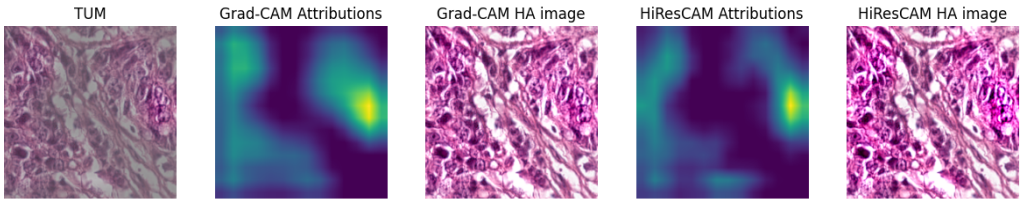
In the first place, one might notice that, so far, HAAS has been tested, in [27], only on datasets whose classes are strongly shape dependent. Cifar-10, STL-10 and ImageNet consist of objects coming from the classes like airplane, car, bird, cat, horse, garbage truck, golf ball, cassette player etc.. which can be recognized mainly based on their shape and without focusing much on small variations in the colour. On the other hand, medical images are more complex. The classes are in principle more densely populated and could have a stronger colour dependency. As a result, changing the intensity of a pixel according to an attribution map might not necessarily assist the model to predict better, but it could rather confuse it. Figure 20 includes representative cases for this argument; as the produced HA images have visible colour differences compared to the original images. Secondly, one might explain a low HAAS score due to insufficiently performing models instead of the explainability algorithms. This approach however does not align well with the testing results of Table 4 where we see that the pool of models used for our experiments consists of both very well performing models (CRC

and Covid-19 datasets) and well performing ones (HAM10000 and BreakHis datasets).

In order to test a potential incompatibility issue with medical datasets we further implemented the technique on the following non-medical datasets: Cifar-10, STL-10 and Imagenette, which were used in the original HAAS paper. For each dataset we trained a loop of 16 *random* VGG19 models for a few epochs in order to see the HAAS scores produced by HiResCAM. In Table 8 of section 4.5 one may find the different hyperparameter training configurations used for each dataset and in Table 9 the HAAS and Mean AUC results of the best and worst performing models, in terms of the HAAS score. Based on the latter table, we see that for every dataset we achieved HAAS score above 1 for a very well performing model while at the same time the worst case scenario included a well performing model of a HAAS score quite close to 1. In other words, it was possible to extract meaningful HAAS scores when we considered the non-medical datasets and not optimally trained models, which comes in contrast to our study with medical datasets and models with more extensive training.

These findings suggest that the HAAS metric might be more sensitive to medical data and it is not always possible to yield robust and helpful results. Nevertheless, this unexpected behaviour is not decisive for future use as it is derived from a very small testing sample of cases.

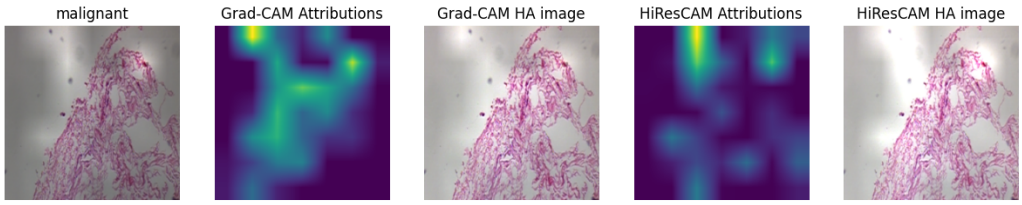
In addition, motivated by the calculations example at the end of section 2.2.3, we observe that the HA image pixels can have *considerable* value difference when compared to the original image pixels. For instance, as calculated in the example, a pixel with value $\frac{1}{2}$ is transformed into a HA pixel with value $\frac{3}{4}$ under $\frac{1}{2}$ attribution value and $\frac{1}{4}$ under $-\frac{1}{2}$ attribution value. Thus, when testing the model on the HA image one has to take into account that the HA image might be *far* from the distribution that the model was trained on. This observation thought not investigated extensively in this study should also be mentioned. It applies to all kinds of datasets, since even if in the above loops of the non-medical datasets we achieved HAAS scores above 1, there were also multiple models of very good performance with HAAS score below 1.



(a) CRC dataset - VGG19

(b) Covid-19 dataset - VGG19

(c) HAM10000 dataset - ResNet50



(d) BreakHis dataset - ResNet50

Figure 20: Examples of HA images.

5.4 An answer to the initial question

Overall, recalling Table 10 results, we see that Max-Sensitivity is the metric that clearly favors HiResCAM attribution maps in all cases while AOPC score either slightly favors HiResCAM (7 out of 8 experiments) or sees no real difference (1 out of 8 experiments) in the attribution maps. As per section 5.2, this was attributed to the fine-grained high resolution HiResCAM attribution maps that seem to distinguish better between salient and non-salient regions of the image, resulting eventually in more informative and resilient maps. Furthermore, it is essential to say that the calculated results align with the results of [38], as presented in chapter 2, despite the fact that the setting is different. Indeed, Max-Sensitivity always favours the faithful algorithm and as a result suggests a

good metric for capturing this *feature* of attribution maps. In addition, the AOPC score most of the times favours the faithful algorithm but not on the level of certainty that Max-Sensitivity does.

On the other hand, the HAAS score does not directly contribute to our comparison study; it rather inspires study to further investigate it. As discussed above, this could be related to the nature of the datasets or the fact that the HA images might be far from the model's learned distribution. However, this is an issue that is still vague and definitely requires more study and experiments.

5.5 Does accuracy align with the evaluation metrics results?

The objective of this section is to analyse the graphs presented in section 4.6 and extend the study on HiResCAM attribution maps. Within the group of all created models, we will be examining if models that have a high balanced accuracy score are associated with *good* evaluation metric results, and whether models that have a mediocre balanced accuracy score are linked with *worse* evaluation metric results.

Starting from Figure 16, one immediately notices that the top four accuracy performing models, those of Covid19 and CRC datasets, form an *almost* monotonous increasing relationship between the balanced accuracy value and the average AOPC value. This pattern is amplified by the VGG19 models of BreakHis and HAM10000. On the hand, the two remaining ResNet50 models deviate from this as they either yield high balanced accuracy and low AOPC scores (BreakHis dataset) or the opposite (HAM10000 dataset).

In regards to the Max-Sensitivity results of Figure 17, we observe that the top performing models yield high Max Sensitivity values, while lower performing models yield lower values as well. This suggests that there is no pattern to connect well performing models with good (i.e. low) Max-Sensitivity results.

Finally, Figure 18 plots the HAAS values of Table 7. As already discussed in section 5.3, this metric did not yield meaningful results and by arranging its values with respect to the balanced accuracy does not seem to provide insights either.

To sum up, plotting the metrics values in the order of the models' balanced accuracy we see that only AOPC has a *robust* behaviour for *well* performing models. On the other hand, as the models' predictive power drops the graph becomes less informative. On top of this, we should highlight the VGG19 behaviour for the AOPC metric where as balanced accuracy increases the mean AOPC value increases as well.

6 Future Work

In this study we addressed the problem of quantifying the quality of attribution maps in a setting where HiResCAM produces attributions faithful to the model decisions while Grad-CAM does not. For this purpose, we considered medical data and a transfer learning approach to investigate if this theoretical result is coupled with superior attribution map behaviour in terms of the evaluation metrics.

During this long course of actions we encountered many obstacles which undoubtedly set many difficulties but also brought out remarks and ideas which could motivate further study. This short chapter aims to draw a sketch for some of them.

Starting with the AOPC score, we should highlight that the problem of choosing a proper function g that efficiently *removes* class information is in general quite blurry. In order to implement the technique one has to take decisions on questions that do not always have an obvious answer and most of the time require extensive experimentation. For example, in the first place, we should decide if we will use constant baseline colour or random noise. Then which constant colour or what kind of random noise ? Further to this, if random noise is chosen, how do we apply it to the image? One option could be to replace the image pixels with the noise pixels while another could be to add noise pixels to the image pixels. All these questions are dataset dependent and as a result could be proven very critical for successfully utilizing the AOPC score. In addition, before we move to the next metric, it may be worth for future reference to shortly describe an approach tested during our experiments that did not yield though quite satisfying results: inspired from the MoRF procedure, one may consider exploring variations of the standard AOPC calculation where, per perturbation step, instead of tracking the class probability, the class attribution map is recalculated. This would lead to a series of attribution maps whose top regions and/or their intensity could be used for further analysis.

On the other hand, regarding the HAAS score, as our calculated results suggest, conducting more experiments on medical data seems a reasonable next step that could most certainly add useful knowledge. One may implement more pre-trained structures or even custom written models and utilize more types of medical data, apart from the MRIs and the colon, skin and breast tissues used in this study. In addition, as discussed in chapter 5, in order to further dive into the relationship between HAAS and the colour sensitive classes, experimenting on datasets with strongly colour-only dependent classes could be useful. For example, what would be the HAAS values if the dataset consists of objects of the same shape and only different colours?

From the models' perspective, one may consider forming ensembles of models and investigate their behaviour in the context of the Grad-CAM vs HiResCAM comparison

via the three evaluation metrics implemented in this study. In this approach one has to decide about the ensemble's aggregation rule and most importantly about the attribution maps' aggregation rule in order to define the attribution map of the ensemble. This is a non-trivial decision and one may come up with different ways to combine the attribution maps. However, once the ensemble setting is *well defined*, then one could focus their investigation on whether the Max-Sensitivity results are preserved in the ensemble setting as well, whether AOPC still exhibits indecisive results in some cases and whether the HAAS score can consistently yield higher scores, hopefully above 1.

To sum up, this study could motivate further experimentation towards directions like the ones described above. In the hope that the gained insights may contribute to a better understanding of the underlying mechanisms we encourage to conduct further investigation into these research directions.

Bibliography

[1] Adrienne Yapo and Joseph Weiss. “Ethical implications of bias in machine learning”. In: (2018).

[2] Adnan Khashman. “Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes”. In: *Expert Systems with Applications* 37.9 (2010), pp. 6233–6239.

[3] Victoria A Banks, Katherine L Plant, and Neville A Stanton. “Driver error or designer error: Using the Perceptual Cycle Model to explore the circumstances surrounding the fatal Tesla crash on 7th May 2016”. In: *Safety science* 108 (2018), pp. 278–285.

[4] Amina Adadi and Mohammed Berrada. “Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)”. In: *IEEE access* 6 (2018), pp. 52138–52160.

[5] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “” Why should i trust you?” Explaining the predictions of any classifier”. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[6] Sebastian Bach et al. “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation”. In: *PloS one* 10.7 (2015), e0130140.

[7] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. “Explainable ai: A review of machine learning interpretability methods”. In: *Entropy* 23.1 (2020), p. 18.

[8] Tim Miller. “Explanation in artificial intelligence: Insights from the social sciences”. In: *Artificial intelligence* 267 (2019), pp. 1–38.

[9] Leilani H Gilpin et al. “Explaining explanations: An overview of interpretability of machine learning”. In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE. 2018, pp. 80–89.

[10] Bas HM Van der Velden et al. “Explainable artificial intelligence (XAI) in deep learning-based medical image analysis”. In: *Medical Image Analysis* (2022), p. 102470.

[11] Amitojdeep Singh, Sourya Sengupta, and Vasudevan Lakshminarayanan. “Explainable deep learning models in medical image analysis”. In: *Journal of Imaging* 6.6 (2020), p. 52.

[12] Zohaib Salahuddin et al. “Transparency of deep neural networks for medical image analysis: A review of interpretability methods”. In: *Computers in biology and medicine* 140 (2022), p. 105111.

[13] Wojciech Samek et al. “Explaining deep neural networks and beyond: A review of methods and applications”. In: *Proceedings of the IEEE* 109.3 (2021), pp. 247–278.

[14] Vaishak Belle and Ioannis Papantonis. “Principles and practice of explainable machine learning”. In: *Frontiers in big Data* (2021), p. 39.
[15] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2nd ed. 2022. URL: https://christophm.github.io/interpretable-ml-book.
[16] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. “Learning important features through propagating activation differences”. In: *International conference on machine learning*. PMLR. 2017, pp. 3145–3153.
[17] Marco Ancona et al. “Towards better understanding of gradient-based attribution methods for deep neural networks”. In: *arXiv preprint arXiv:1711.06104* (2017).
[18] Daniel Smilkov et al. “Smoothgrad: removing noise by adding noise”. In: *arXiv preprint arXiv:1706.03825* (2017).
[19] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions”. In: *Advances in neural information processing systems* 30 (2017).
[20] Matthew D Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks”. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer. 2014, pp. 818–833.
[21] Kunpeng Li et al. “Tell me where to look: Guided attention inference network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 9215–9223.
[22] Jo Schlemper et al. “Attention gated networks: Learning to leverage salient regions in medical images”. In: *Medical image analysis* 53 (2019), pp. 197–207.
[23] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. “Axiomatic attribution for deep networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 3319–3328.
[24] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. “Methods for interpreting and understanding deep neural networks”. In: *Digital signal processing* 73 (2018), pp. 1–15.
[25] Marco Ancona et al. “Towards better understanding of gradient-based attribution methods for deep neural networks”. In: *arXiv preprint arXiv:1711.06104* (2017).
[26] Wojciech Samek et al. “Evaluating the visualization of what a deep neural network has learned”. In: *IEEE transactions on neural networks and learning systems* 28.11 (2016), pp. 2660–2673.
[27] Junhee Lee et al. “Heatmap Assisted Accuracy Score Evaluation Method for Machine-Centric Explainable Deep Neural Networks”. In: *IEEE Access* 10 (2022), pp. 64832–64849.

[28] Chih-Kuan Yeh et al. “On the (in) fidelity and sensitivity of explanations”. In: *Advances in Neural Information Processing Systems* 32 (2019).
[29] Ramprasaath R Selvaraju et al. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.
[30] Rachel Lea Draelos and Lawrence Carin. “Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks”. In: *arXiv e-prints* (2020), arXiv–2011.
[31] Bolei Zhou et al. “Learning deep features for discriminative localization”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.
[32] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
[33] Gao Huang et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.
[34] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.
[35] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.
[36] Julius Adebayo et al. “Sanity checks for saliency maps”. In: *Advances in neural information processing systems* 31 (2018).
[37] Rachel Lea Draelos and Lawrence Carin. “Explainable multiple abnormality classification of chest CT volumes”. In: *Artificial Intelligence in Medicine* 132 (2022), p. 102372.
[38] Ioannis Kakogeorgiou and Konstantinos Karantzalos. “Evaluating explainable artificial intelligence methods for multi-label deep learning classification tasks in remote sensing”. In: *International Journal of Applied Earth Observation and Geoinformation* 103 (2021), p. 102520.
[39] Maria Mora-Rodriguez et al. “Incorporating Explainable Ai into the auditing practices”. In: (2021). URL: https://xxicongreso.aeca.es/wp-content/uploads/2021/09/129w3.pdf.
[40] Ismini Psychoula et al. “Explainable machine learning for fraud detection”. In: *Computer* 54.10 (2021), pp. 49–59.
[41] Shahin Atakishiyev et al. “Explainable artificial intelligence for autonomous driving: a comprehensive overview and field guide for future research directions”. In: *arXiv preprint arXiv:2112.11561* (2021).

[42] Zhibo Zhang et al. “Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research”. In: *arXiv preprint arXiv:2208.14937* (2022).
[43] Weisi Guo. “Explainable artificial intelligence for 6G: Improving trust between human and machine”. In: *IEEE Communications Magazine* 58.6 (2020), pp. 39–45.
[44] Shen Wang et al. “Explainable AI for B5G/6G: technical aspects, use cases, and research challenges”. In: *arXiv e-prints* (2021), arXiv–2112.
[45] Lujain Ibrahim et al. “Explainable prediction of acute myocardial infarction using machine learning and shapley values”. In: *Ieee Access* 8 (2020), pp. 210410–210417.
[46] Carson K Leung et al. “Explainable data analytics for disease and healthcare informatics”. In: *Proceedings of the 25th International Database Engineering & Applications Symposium*. 2021, pp. 65–74.
[47] Flavio Di Martino and Franca Delmastro. “Explainable AI for clinical and remote health applications: a survey on tabular and time series data”. In: *Artificial Intelligence Review* (2022), pp. 1–55.
[48] Doniyorjon Mukhtorov et al. “Endoscopic Image Classification Based on Explainable Deep Learning”. In: *Sensors* 23.6 (2023), p. 3176.
[49] Pieter-Jan Kindermans et al. “The (un) reliability of saliency methods”. In: *Explainable AI: Interpreting, explaining and visualizing deep learning* (2019), pp. 267–280.
[50] Jing Zhang et al. “Explainability for regression CNN in fetal head circumference estimation from ultrasound images”. In: *Interpretable and Annotation-Efficient Learning for Medical Image Computing: Third International Workshop, iMIMIC 2020, Second International Workshop, MIL3ID 2020, and 5th International Workshop, LABELS 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4–8, 2020, Proceedings 3*. Springer. 2020, pp. 73–82.
[51] Matthew Watson, Bashar Awwad Shiekh Hasan, and Noura Al Moubayed. “Using model explanations to guide deep learning models towards consistent explanations for EHR data”. In: *Scientific Reports* 12.1 (2022), p. 19899.
[52] “Breast Cancer Wisconsin (BCW) Dataset”. In: (2016). URL: https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data.
[53] “MIMIC-V Database”. In: (2016). URL: https://archive.physionet.org/physiobank/database/mimicdb/.
[54] Sebastian Meister et al. “Investigations on Explainable Artificial Intelligence methods for the deep learning classification of fibre layup defect in the automated composite manufacturing”. In: *Composites Part B: Engineering* 224 (2021), p. 109160.
[55] “MNIST Dataset”. In: (2019). URL: https://www.kaggle.com/datasets/hojjatk/mnist-dataset.

[56] "Cifar-10 Dataset". In: (2014). URL: https://www.kaggle.com/c/cifar-10/data.
[57] "STL-10 Dataset". In: (2018). URL: https://www.kaggle.com/datasets/jessicali9530/stl10.
[58] "ImageNet Dataset". In: (2011). URL: https://www.image-net.org/.
[59] "BigEarthNet Dataset". In: (2019). URL: https://bigearth.net/.
[60] "SEN12MS Dataset". In: (2019). URL: https://mediatum.ub.tum.de/1474000.
[61] Jakob Nikolas Kather, Niels Halama, and Alexander Marx. *100000 histological images of human colorectal cancer and healthy tissue*. 2018. URL: https://zenodo.org/record/1214456#.ZBsjCHZBxPY.
[62] *COVID-19 Radiography Database*. 2020. URL: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database.
[63] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. "The HAM10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions". In: *Scientific data* 5 (2018), p. 180161. URL: https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000.
[64] Fabio A Spanhol et al. "A dataset for breast cancer histopathological image classification". In: *Ieee transactions on biomedical engineering* 63.7 (2015), pp. 1455–1462.
[65] Jakob Nikolas Kather et al. "Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study". In: *PLoS medicine* 16.1 (2019), e1002730.
[66] Radwan Al. Shawesh and Yi Xiang Chen. "Enhancing Histopathological Colorectal Cancer Image Classification by using Convolutional Neural Network". In: *medRxiv* (2021), pp. 2021–03.
[67] Athanasios Kallipolitis, Kyriakos Revelos, and Ilias Maglogiannis. "Ensembling EfficientNets for the classification and interpretation of histopathology images". In: *Algorithms* 14.10 (2021), p. 278.
[68] A Ben Hamida et al. "Deep learning for colon cancer histopathological images analysis". In: *Computers in Biology and Medicine* 136 (2021), p. 104730.
[69] Muhammad EH Chowdhury et al. "Can AI help in screening viral and COVID-19 pneumonia?" In: *Ieee Access* 8 (2020), pp. 132665–132676.
[70] Tawsifur Rahman et al. "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images". In: *Computers in biology and medicine* 132 (2021), p. 104319.
[71] Noel Codella et al. "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)". In: (2019).
[72] Fabio A Spanhol et al. "Breast cancer histopathological image classification using convolutional neural networks". In: *Journal of Digital Imaging* 30.3 (2017), pp. 385–395. URL: https://www.kaggle.com/datasets/ambarish/breakhis.

[73] Yassir Benhammou et al. “BreakHis based breast cancer automatic diagnosis using deep learning: Taxonomy, survey and insights”. In: *Neurocomputing* 375 (2020), pp. 9–24.
[74] Sara Hosseinzadeh Kassani et al. “Classification of histopathological biopsy images using ensemble of deep learning networks”. In: *arXiv preprint arXiv:1909.11870* (2019).
[75] “SIIM-ISIC Melanoma Classification”. In: (2018). URL: https://www.kaggle.com/c/siim-isic-melanoma-classification/leaderboard.
[76] “Max Sensitivity GitHub”. In: (2019). URL: https://github.com/chihkuanyeh/saliency_evaluation.