TOWARDS THE APPLICATION OF ATTENTION MECHANISMS FOR MEDICAL IMAGE CLASSIFICATION

A PREPRINT

Tiago Gonçalves
Faculdade de Engenharia
Universidade do Porto
Porto, Portugal
up201607753@fe.up.pt

June 22, 2021

ABSTRACT

Despite their high performance, deep learning algorithms still work as black boxes and are not capable of explaining their predictions in a human-understandable manner, thus leading to a lack of transparency which may jeopardise the acceptance of these technologies by the healthcare community. Therefore, the topic of explainable artificial intelligence (xAI) appeared to address this issue. Several techniques to assure that models are focusing on the important parts of the images and learning relevant features have been proposed. These attention mechanisms have reported improvements in the predictive performances and the explainability of models. Following this trend, we propose a comparative study of the application of attention mechanisms in state-of-the-art deep learning architectures using two different use cases in the context of medical image classification: breast cancer detection in mammography images and pleural effusion detection in chest X-ray images.

***K*eywords** Deep learning · Medical image classification · Attention mechanisms · Breast cancer · Thorax pathology

1 Introduction

Traditional machine learning methods generally rely on careful data engineering techniques and domain knowledge to achieve meaningful features that could be given to a learning algorithm capable of classifying or detecting patterns in the input data [1]. Hence, one of the major drawbacks of these methods is their limited ability to process data in its raw form. To overcome this issue, deep learning (DL) approaches have been proposed. One of the main concepts behind the functioning of DL algorithms is their capacity of being fed with raw data and to extract the suitable representations needed for the learning task; this is known as *representation learning*. DL algorithms are representation learning methods with multiple levels of representation capable of transforming the representation at one level (*i.e.*, starting at the raw input) into a representation at a more abstract (*i.e.*, semantic) level [1]. With the democratised access to data and the increase of the availability of computational power, DL-based methodologies have been achieving nearly-human performances in several areas of science, business and government.

The medical community is aware of the development and success of DL-based technologies in healthcare-related tasks. However, due to their high complexity and number of parameters, DL algorithms still work as black-boxes and are not capable of explaining their predictions in a human-understandable manner. This lack of transparency may be delaying the implementation of these technologies into the real-world context of healthcare professionals [2]. Hence, it is of utmost importance that the researchers of the machine learning community develop methodologies that promote transparency and explainability of these high performing algorithms to increase confidence by the end-users and speed up their implementation [3].

Recently, attention-based mechanisms for DL algorithms have been explored in several domains. Reported results suggest that this technique may act as a regularisation variable that assures that algorithms will focus on the most important features during training [4]. Following the works of [5] and [6], we performed an exploratory analysis of the

application of several state-of-the-art pre-trained models (*i.e.*, backbones) and the use of attention mechanisms to assess the impact on the predictive performance of these models, as well as their explainability. We used medical images from two different use cases: breast cancer detection in mammography images and pleural effusion detection in chest X-ray images.

The remainder of the paper is organised as follows: section 2 gives an overview of the application of deep learning in medical image classification, explainable artificial intelligence and attention mechanisms; section 3 shows the methodologies and data sets that were used to address this work; section 4 presents the results and provides a brief discussion; and section 5 concludes the paper with the main contributions and proposes future work approaches. The code is available online in a GitHub repository[1].

2 Background

2.1 Deep Learning for Medical Image Classification

In medical image classification, the main task is to output a diagnosis (*e.g.*, presence or absence of a disease) based on one or more input images [7]. Given the high predictive performance rates of CNNs in other computer vision tasks (*e.g.*, natural image recognition), the application of DL algorithms in medical image classification occurred almost naturally. Besides, it benefited mostly from the application of *transfer learning*. Transfer learning consists of the use of pre-trained networks in large image databases (*e.g.*, ImageNet[2]) to speed up the training of DL models in medical image databases. It can be employed essentially in two ways: 1) using the pre-trained network has a feature extractor from the medical data and train a classifier using these features; 2) fine-tuning the pre-trained network in the medical data. However, there are already some approaches that use novel model architectures trained from scratch to address a specific challenge. Image modalities and applications range from brain MRI to retinal imaging and digital pathology to lung computed tomography [7].

Although is it not the aim of this work to describe exhaustively all the state-of-the-art methods in the fields of breast cancer or chest pathology classification, we consider it important to point out some important works in these areas. In the field of breast cancer classification, an international study on the evaluation of an AI system based on DL methods for breast cancer screening using mammography images has been proposed and the authors claimed that their system is capable of surpassing human experts in breast cancer prediction [8]. The field of chest pathology is wider, but, currently, approaches have been dealing mostly with the classification of COVID-19 X-ray images, detection of infectious diseases and abnormal cases [9].

2.2 Explainable Artificial Intelligence

Despite the high performances achieved by DL-based algorithms, their transition into real-world applications is not trivial, due to their complexity (*i.e.*, high-number of parameters) and their black-box behaviour which may jeopardise their acceptance by the clinical community. Therefore, the topic of explainable artificial intelligence (xAI) appeared intending to contribute to a more transparent AI [2, 10]. Although there is no clear distinction between explainability and interpretability [11], one may think of these as a three-stage process: pre-, in- and post-model [12]. Pre-model methods focus on understanding the data distribution before building the model, through exploratory data analysis [13, 14]. This comprehension of the data may contribute to higher confidence with the posterior decisions that a model can provide. In-model approaches seek to integrate interpretability inside the model. To do this, one may follow different strategies: models based in rules [15], models based in cases [16] and the use of regularization techniques during training to obtain sparser or monotonic models [1, 17]. Finally, post-model strategies are related to a posterior analysis of the model predictions. For instance, in medical imaging, this can be done using the gradient information to identify the areas of the image that mostly contribute to the final decision [18, 19], inserting a perturbation and observing the prediction [20], inverting the representations back to the input pixel space [21, 22] or connecting the representations to semantic concepts [23]. In healthcare applications, it is fundamental to assess the quality of these explanations [3, 24], for the sake of transparency, ethics and fairness. Current studies suggest that the use of in-model methods may contribute to more transparent model decisions and better explanations [3, 25].

Having acknowledged that just being able to obtain explanations is not enough [3], it is important to understand what the algorithms are already learning and to evaluate the quality of such explanations (e.g., understand if the algorithms are extracting relevant features for the clinical context). To facilitate trust (and increase transparency) in AI algorithms it is important to ensure *a priori* that these models are interpretable. This can only be achieved if one understands

[1] `https://github.com/TiagoFilipeSousaGoncalves/attention-mechanisms-healthcare`

[2] `https://www.image-net.org/index.php`

first how decisions are made in the clinical context. For instance, to produce a decision, clinicians may use examples that have similarities with the sample under analysis. Contrarily, it is also common to use examples that represent the opposite to understand the decision. This type of prior knowledge can be integrated during the training of the models (e.g., imposing model sparsity through regularization) or before, by building models based on cases [12].

Also, a current trend on AI is the development of ethical and fair algorithms [26] that take into account the intrinsic variability of the datasets to produce models that can be agnostic to several features that should not be relevant (e.g., skin, colour or gender) [3].

2.3 Attention Mechanisms

The intuition behind the application of attention mechanisms in DL algorithms is inspired by the field of psychology, according to which humans tend to selectively concentrate on a part of the information [27]. For instance, the human visual system tends to selectively focus on specific parts of an image while ignoring others [28]. Following this rationale, it is recognised that in AI systems, some parts of the inputs may be more relevant than others (*e.g.*, in automatic translation systems, only a subset of words is relevant) [4].

The use of attention was initially proposed in [29], for the task of neural machine translation. In this work, the authors use a RNN-based encoder-decoder architecture which presented two challenges: 1) the decoder needs to compress all the input information into a single fixed-length vector and pass it to the decoder; 2) ensuring model alignment between input and output sequences was not possible. Hence, it was necessary to develop an attention mechanism that could support the decoder in focusing on the relevant parts of the inputs [4]. Naturally, during the training phase, an extra task is added: the learning of the attention weights. Nevertheless, this approach showed improved results against the state-of-the-art and paved the way for the creation of novel attention-based methodologies.

Following [4], attention models can be classified into one of the following categories, according to:

- **Numbers of Sequences:** This category takes into account the number of input and output sequences. At this level, we may consider the following types of attention: 1) **distinctive**, when the candidate and query states belong to two different input and output sequences respectively; 2) **co-attention**, when we consider multiple input sequences at the same time and the main goal is to jointly learn their attention weights; 3) **self-attention**, when the query and candidate states belong to the same input sequence.
- **Number of Abstraction Levels:** This category takes into account the number of representation/feature levels where the model will learn the attention weights. In this case, we may consider the following types of attention: 1) **single-level**, when the attention weights are computed only for the original input sequence; 2) **multi-level**, when we apply the attention mechanism on multiple levels of abstraction of the input sequence, usually in a sequential manner.
- **Number of Positions:** This category takes into account the number of positions of the input sequence where the attention weights are learned. Here, we may consider the following types of attention: 1) **soft**, which consists of a weighted average of all the hidden states of the input sequence to build the context vector; 2) **hard**, which consists of a stochastic sampling of the hidden states to build the context vector; 3) **global**, which is the term used in the machine-translation field to describe soft attention; 4) **local**, which is also part of the machine-translation field, consists of the detection of an attention point and then use a window around that point to compute local attention weights.
- **Number of Representations:** This category takes into account the number of different feature representations of the input sequence used in the learning task. In this class, we can consider the following types of attention: 1) **single-representational**, which is the most common and consists of learning the attention weights using just a single feature representation; 2) **multi-representational**, which consists of learning the attention weights using different representations (of the same input sequence) and creating a context vector which is the result of a weighted combination of these multiple representations and their attention weights; 3) **multi-dimensional**, which consists of computing the attention weights along several dimensions of the representation of the input sequence, which will result in the computation of the relevance of each dimension to the learning task.

2.3.1 Attention Mechanisms in Medical Image Classification

Recently, a CNN with a multi-level dual-attention mechanism (MLDAM) has been proposed for macular optical coherence tomography classification [5]. The main novelty of this work in the context of medical image classification is the joint application of a *self-attention* and a *multi-level attention* mechanisms that allow the network (the authors use the ResNet-50 [30] as backbone network) to learn relevant features in coarser as well as finer sub-spaces. In their article [5], the authors state that this technique enables the network to utilize the information of coarser features

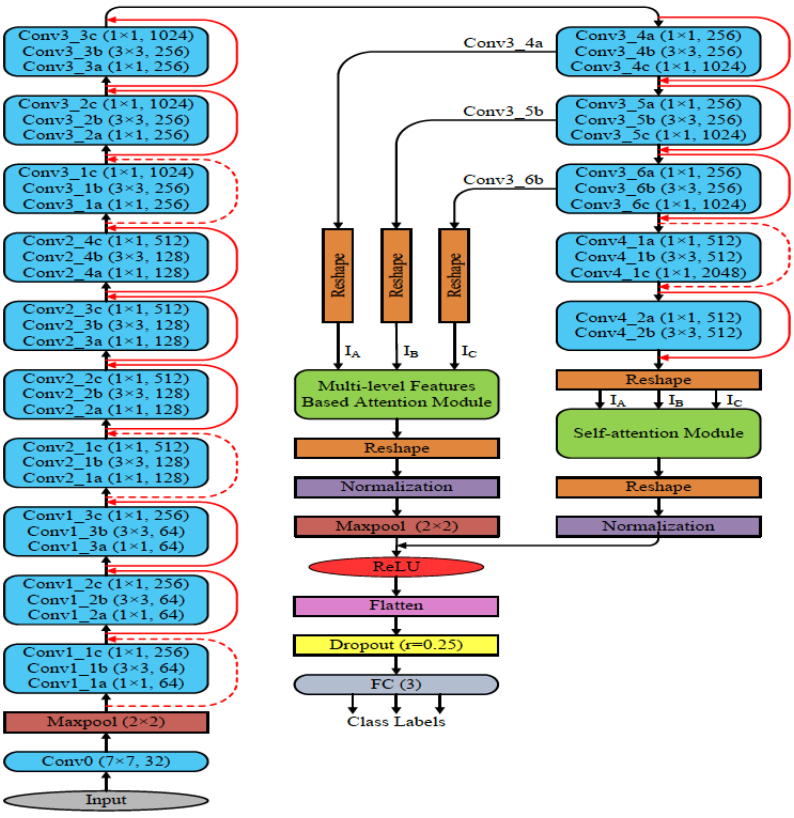

Figure 1: Architecture of the MLDAM, proposed by [5].

preventing loss of any useful information, thus enabling the network to be trained to yield more focused features as input to the classifier leading to better convergence. Figure 1 shows the architecture of the MLDAM, proposed by [5].

Regarding the impact of the application of attention mechanisms in the interpretability of the DL algorithms, we point to the work proposed by [6]. Using iris presentation attack detection as the main use case, this work proposes the jointly use of a position attention module (PAM) and a channel attention module (CAM) to refine the pixel values at spatial and channel levels. These refined features are then fused through an element-wise sum. The authors relate their work with the field of interpretability through an analysis of the saliency maps produced by the gradient-weighted class activation mapping (Grad-CAM) [31] and conclude that the use of attention modules has enabled the network to shift the focus on to the annular iris region. Figure 2 shows the architecture of the PAM and CAM modules, proposed by [6].

3 Methodology

3.1 Data

For this work, we decided to perform experiments on two different use cases using medical images: breast cancer detection in mammography images (see 3.1.1) and pleural effusion detection in chest X-ray images (see 3.1.2).

3.1.1 CBIS-DDSM

CBIS-DDSM[3] (Curated Breast Imaging Subset of DDSM) is an updated and standardized version of the Digital Database for Screening Mammography (DDSM) [32]. In this study, we work with a data set with 2078 training images, 363 validation images and 591 test images. Images are assigned label "0" (*benign*) or label "1" (*malign*).

[3] `https://wiki.cancerimagingarchive.net/display/Public/CBIS-DDSM`

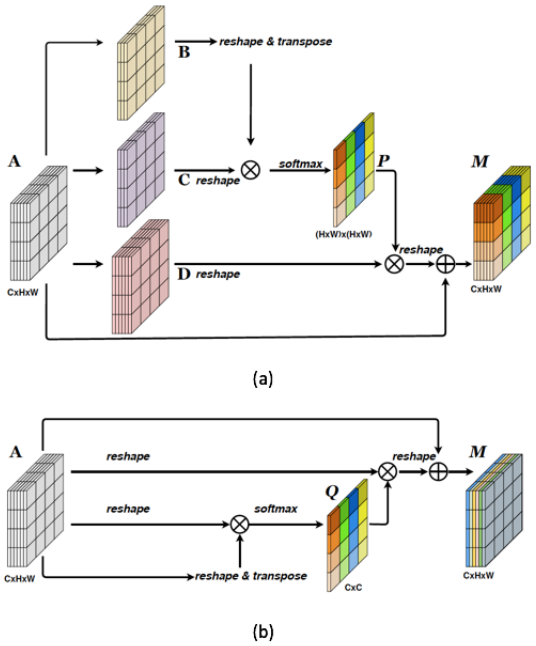

Figure 2: Architecture of the PAM (a) and CAM (b) modules, proposed by [6].

3.1.2 MIMIC-CXR

MIMIC-CXR[4] contains 227,835 imaging studies for 64,588 patients presenting to the Beth Israel Deaconess Medical Center Emergency Department between 2011 - 2016. Each imaging study can contain one or more images, usually a frontal view and a lateral view. In this study, we work with a data set with 61203 training images, 534 validation images and 1072 test images. Images are assigned label "0" (*normal*) or label "1" (*pleural effusion*).

3.2 Implementation

In this work, we performed a comparative study using three state-of-the-art pre-trained DL models as backbones (see 3.2.2). To assess the influence of the use of attention mechanisms, we adapted the MLDAM architecture described in [5] (see 3.2.3) for each of the backbones. We also tested the effect of the presence of a data augmentation strategy (see 3.2.4).

3.2.1 Data Processing

The data processing pipeline in this work is employed as follows:

1. Images are resized to the final size of 224×224;
2. A z-normalisation is applied to each RGB channel. Let in be the input image, out be the output image, $mean$ be the array of means of each channel, std be the array of the standard deviations of each channel and c_i the channel c with index i. Hence, $out[c_i] = \frac{in[c_i] - mean[c_i])}{std[c_i]}$, with $i \in [0, 1, 2]$, $mean = [0.485, 0.456, 0.406]$ and $std = [0.229, 0.224, 0.225]$.

The images of CBIS-DDSM go through a preliminary step before entering the data processing pipeline. Since all the backbones used in this work require the input image to be RGB (see 3.2.2), we have to copy and concatenate each image three times to achieve images with the three channels. The images of MIMIC-CXR are already in this format.

3.2.2 Backbones

We decided to build all the experiments reported in this work on top of three state-of-the-art deep learning models: 1) VGG-16 [33], which was a pioneer in the use of deeper convolutional neural networks architectures for image

[4] `https://mimic.physionet.org/iv/modules/cxr/about/`

classification; 2) ResNet-50 [30], which introduced the *residual learning blocks* to facilitate weight optimisation and improve classification accuracy; 3) DenseNet-121 [34], which proposed the use of densely connected convolutional layers to facilitate training by improving the flow of information and gradients throughout the network. All these backbones are initialised with their ImageNet pre-trained weights[5].

3.2.3 Multi-Level Dual-Attention Mechanism (MLDAM)

To implement the MLDAM mechanism for the three different backbones explained in 3.2.2, we tried to assure diversity in the levels and scales of the features that were used, taking into account their different architectures. Following the notation in [5], let I_A, I_B and I_C be the multi-level features extracted from each backbone:

1. DenseNet-121: in this backbone we extract features from the different dense-blocks resulting in a I_A with shape $[512, 28, 28]$, I_B with shape $[1024, 14, 14]$ and I_C with shape $[1024, 7, 7]$;
2. ResNet-50: in this backbone we extract features from different residual-blocks resulting in a I_A with shape $[512, 28, 28]$, I_B with shape $[1024, 14, 14]$ and I_C with shape $[2048, 7, 7]$;
3. VGG-16: in this backbone we extract features from different convolutional-blocks resulting in a I_A with shape $[256, 28, 28]$, I_B with shape $[512, 14, 14]$ and I_C with shape $[512, 7, 7]$.

3.2.4 Data Augmentation

All the data augmentation functions used in this work are implemented in the *torchvision*[6] library of the PyTorch framework for Python. The data augmentation strategy employed in this work is composed of several random affine transformations:

1. Random Rotations: let α be the angle of rotation in degrees. $\alpha \in [-10, 10]$.
2. Random Translations: let h and v be the horizontal and vertical translation shifts, respectively. $h = 0.05$ and $v = 0.1$.
3. Random Scaling: let s be the scaling factor. $s \in [0.95, 1.05]$
4. Random Horizontal Flip: the horizontal flip is applied with probability $p = 0.5$.

This strategy is applied to both CBIS-DDSM and MIMIC-CXR data sets.

3.2.5 Performance Metrics

To assess the predictive performance of our models we compute the accuracy, precision, recall and F1-score. Considering the binary classification cases presented in this work, let TP be the true positives, TN be the true negatives, FP be the false positives and FN be the false negatives.

The accuracy is the fraction of predictions that the model got right and can be computed according to Equation 1:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

The precision is the fraction of positive identifications that was actually correct and can be computed according to Equation 2:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

The recall is the fraction of actual positives that was identified correctly and can be computed according to Equation 3:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

The F1-score is the harmonic mean of the precision and recall and can be computed according to Equation 4:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

All these performance metrics are upper-bounded by 1 (*i.e.*, the best value) and lower-bounded by 0 (*i.e.*, the worse value).

[5] `https://pytorch.org/vision/stable/models.html`

[6] `https://pytorch.org/vision/stable/index.html`

Table 1: Accuracy results obtained for the test set of the CBIS-DDSM data set.

Model	Weights	Baseline	Baseline with Data Augmentation	MLDAM	MLDAM with Data Augmentation
DenseNet-121	Training	0.6650	0.6142	0.6210	0.5871
	Validation	0.6108	0.5584	0.6396	0.6125
ResNet-50	Training	0.6514	0.6396	-	0.5973
	Validation	0.5956	0.6176	0.5939	0.5854
VGG-16	Training	0.6650	0.6074	0.5939	0.5854
	Validation	0.6244	0.6514	0.6210	0.6041

3.2.6 Saliency Maps

To generate the saliency maps, we applied the algorithm proposed by [35], which returns the gradients with respect to inputs. We used the implementation provided by the Captum library for Python, which is built on top of PyTorch[7].

3.2.7 Training Settings

In this work we performed experiments with four use cases:

1. Baseline, which consists of the training of the backbone models described in 3.2.2, without data augmentation strategies.
2. Baseline with Data Augmentation, which consists of the training of the backbone models described in 3.2.2, with the data augmentation strategies described in 3.2.4.
3. MLDAM, which consists of the training of the backbone models described in 3.2.2 with the MLDAM described in 3.2.3, but without data augmentation strategies.
4. MLDAM with Data Augmentation, which consists of the training of the backbone models described in 3.2.2 with the MLDAM described in 3.2.3 and with the data augmentation strategies described in 3.2.4.

Each model is trained for a maximum of 300 epochs, with binary cross-entropy as the loss function and Adaptive Moment Estimation (Adam) [36] with learning rate 1×10^{-4} as the optimisation algorithm. The batch size varied from 1 to 4, depending on the available GPU memory. We save the best model's parameters in both training and validation sets according to the value of the loss. During training we also monitored the accuracy, precision, recall and F1-score performance metrics (see 3.2.5).

3.2.8 Testing Settings

In the test phase, we tested all the trained models (using the best weights in both training and validation sets) in the test set of each database and computed the accuracy, precision, recall and F1-score. We generated saliency maps for the positive and negative samples of the test set that were correctly predicted by all the use cases (see 3.2.7) of all backbones (see 3.2.2), to assure a fair intra- and inter-comparison. It is important to note that these saliency maps were generated using the models loaded with the best weights in the validation set of each database.

4 Results and Discussion

4.1 CBIS-DDSM

Table 1, Table 2, Table 3 and Table 4 present the accuracy, precision, recall and F1-score results obtained for the test set using the best model parameters for both training and validation sets applied to the use cases studied in this work. There isn't a clear pattern in the results that shows that a given strategy is better than the other, since, for different backbones, different training use cases and different best model parameters, the results change. However, it seems that the baseline models (with or without data augmentation) perform fairly well. We also noticed that the overall results need further improvements. Please note that results for the best model parameters in training for the MLDAM use case of ResNet-50 could not be obtained due to an error in the saved file.

Figure 3, Figure 4 and Figure 5 present the saliency maps obtained for an image with label "0" for all the use cases of the DenseNet-121, ResNet-50 and VGG-16 backbones, respectively.

Figure 6, Figure 7 and Figure 8 present the saliency maps obtained for an image with label "1" for all the use cases of the DenseNet-121, ResNet-50 and VGG-16 backbones, respectively.

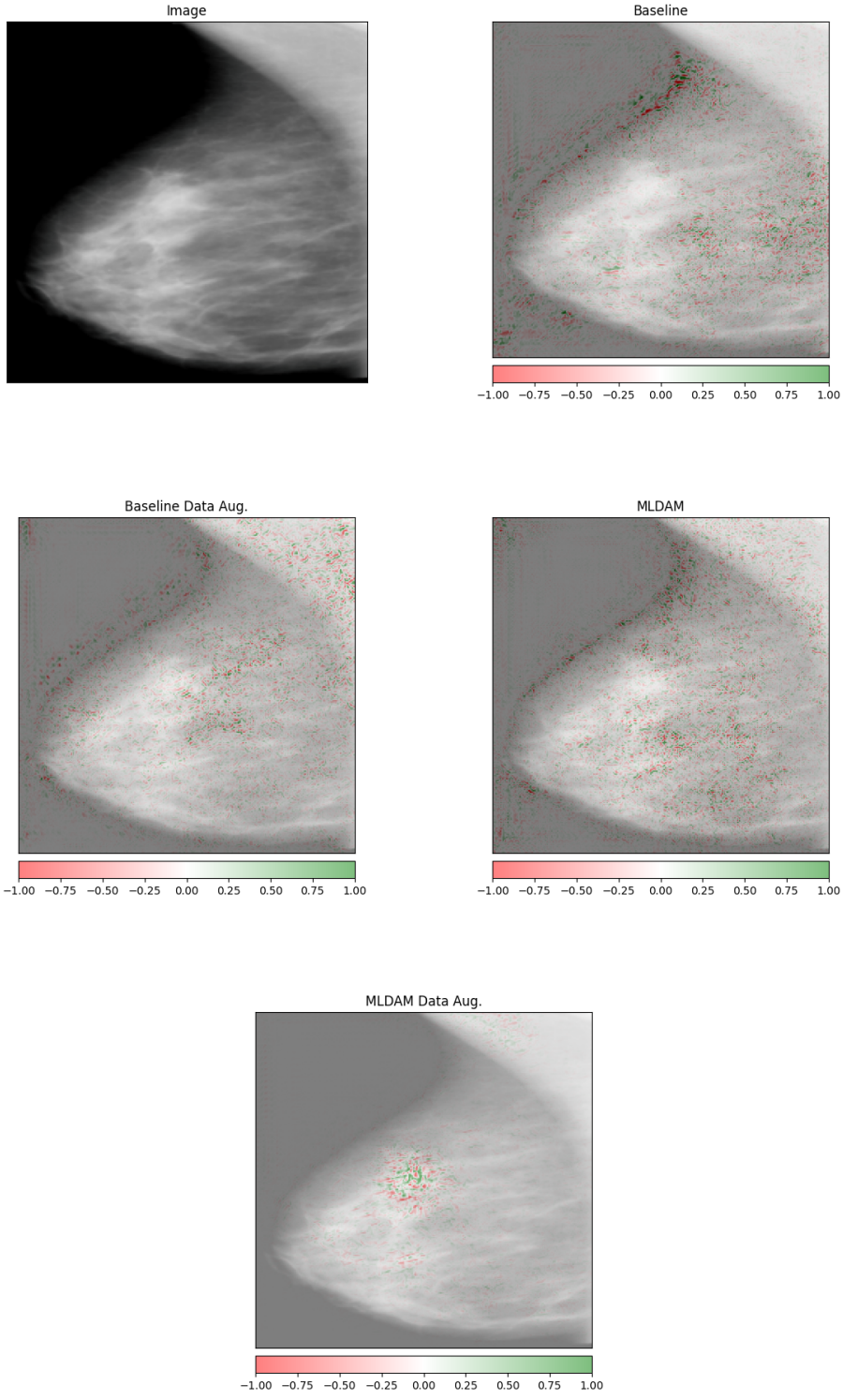

Figure 3: Examples of saliency maps obtained for an image with label "0" (*i.e.*, benign) of the CBIS-DDSM data set, using the DenseNet-121 backbone model.

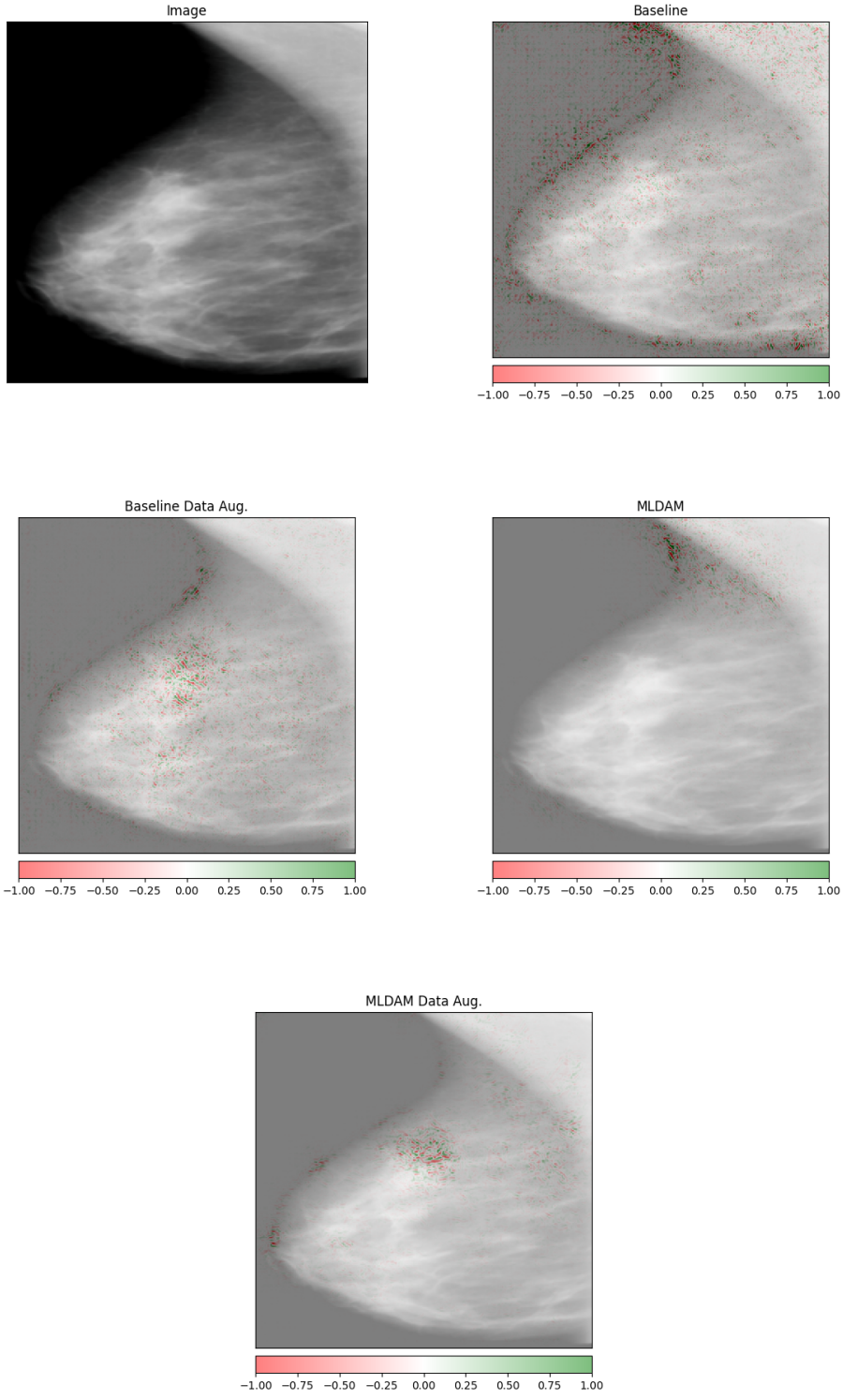

Figure 4: Examples of saliency maps obtained for an image with label "0" (*i.e.*, benign) of the CBIS-DDSM data set, using the ResNet-50 backbone model.

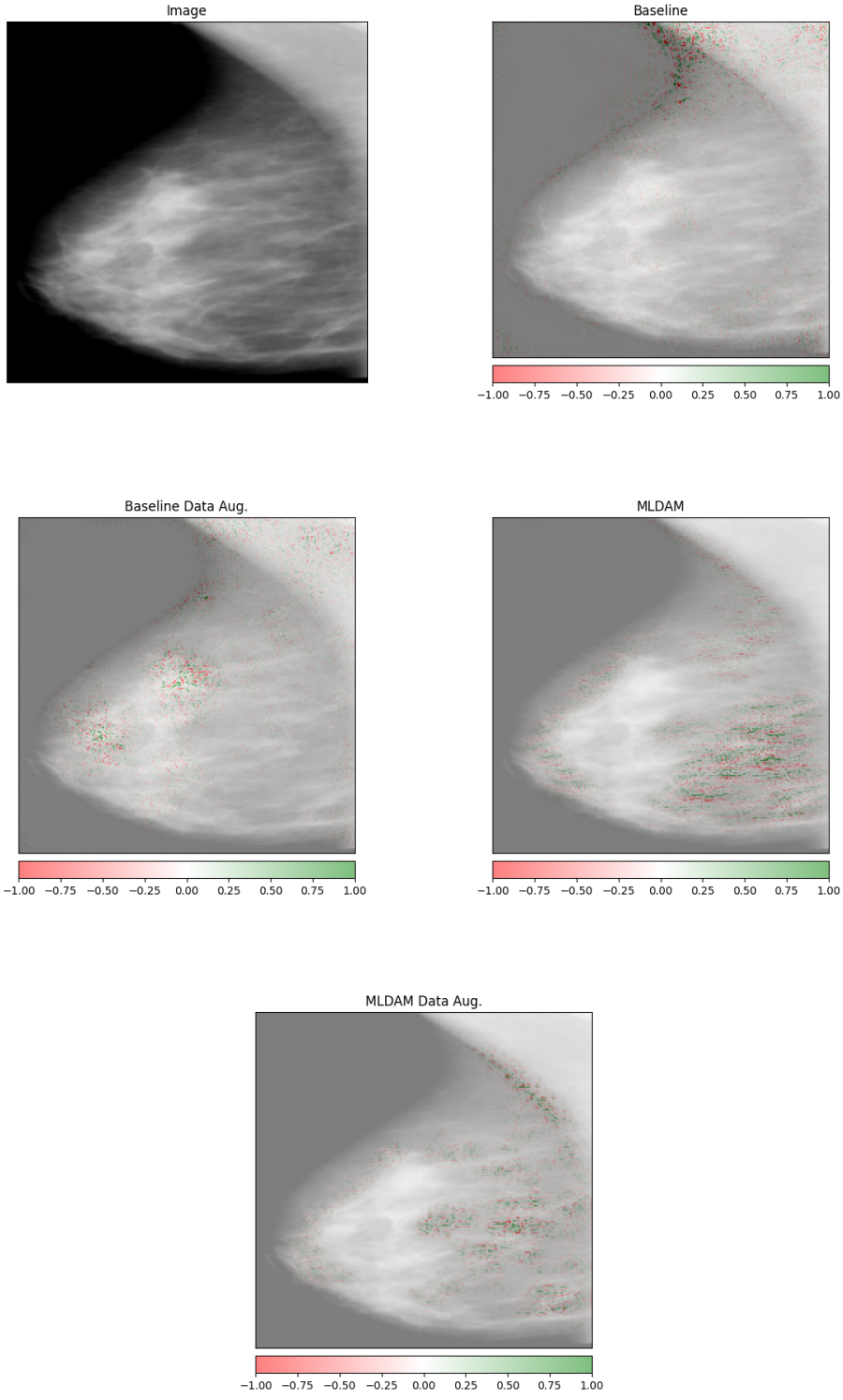

Figure 5: Examples of saliency maps obtained for an image with label "0" (*i.e.*, benign) of the CBIS-DDSM data set, using the VGG-16 backbone model.

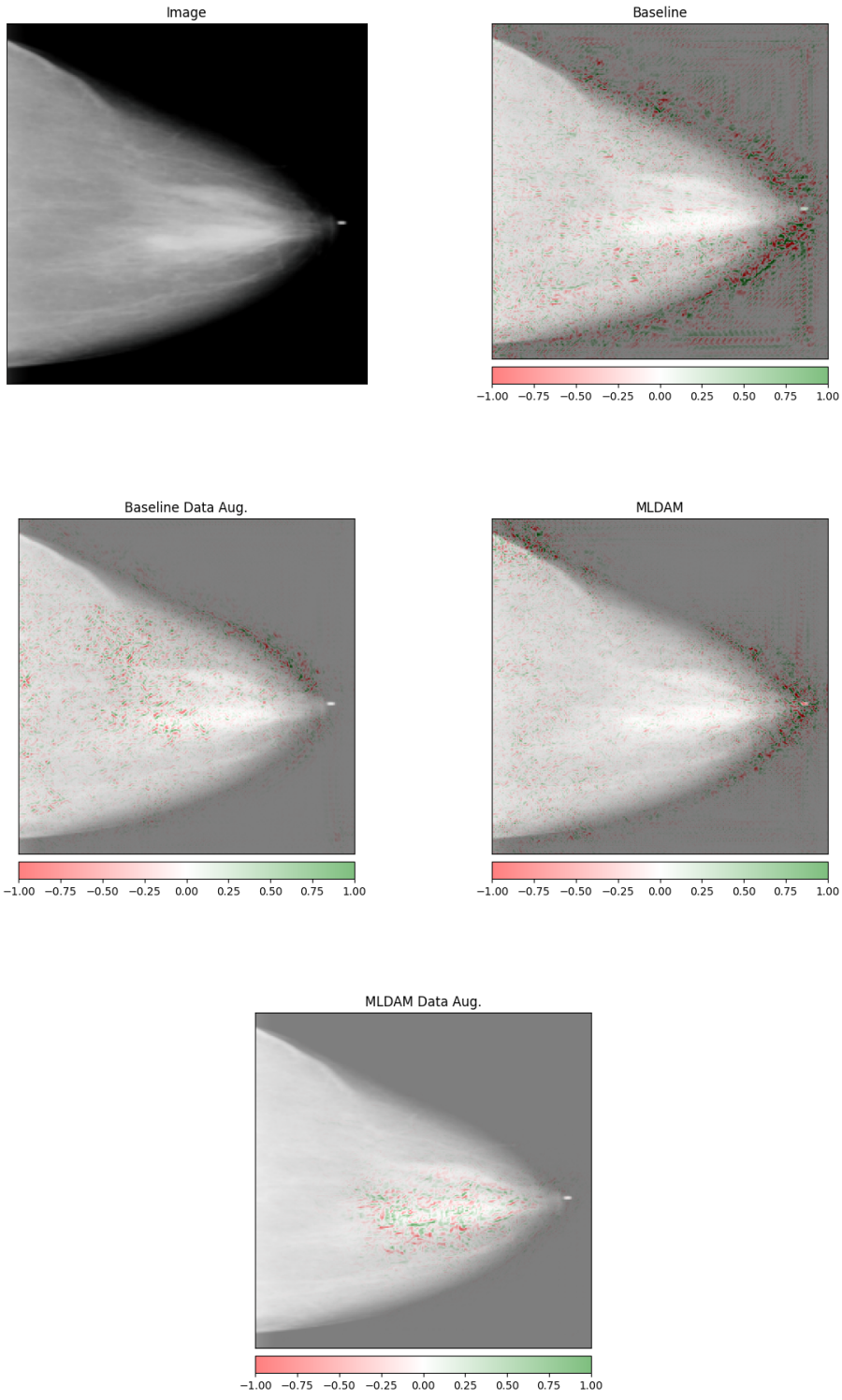

Figure 6: Examples of saliency maps obtained for an image with label "1" (*i.e.*, malignant) of the CBIS-DDSM data set, using the DenseNet-121 backbone model.

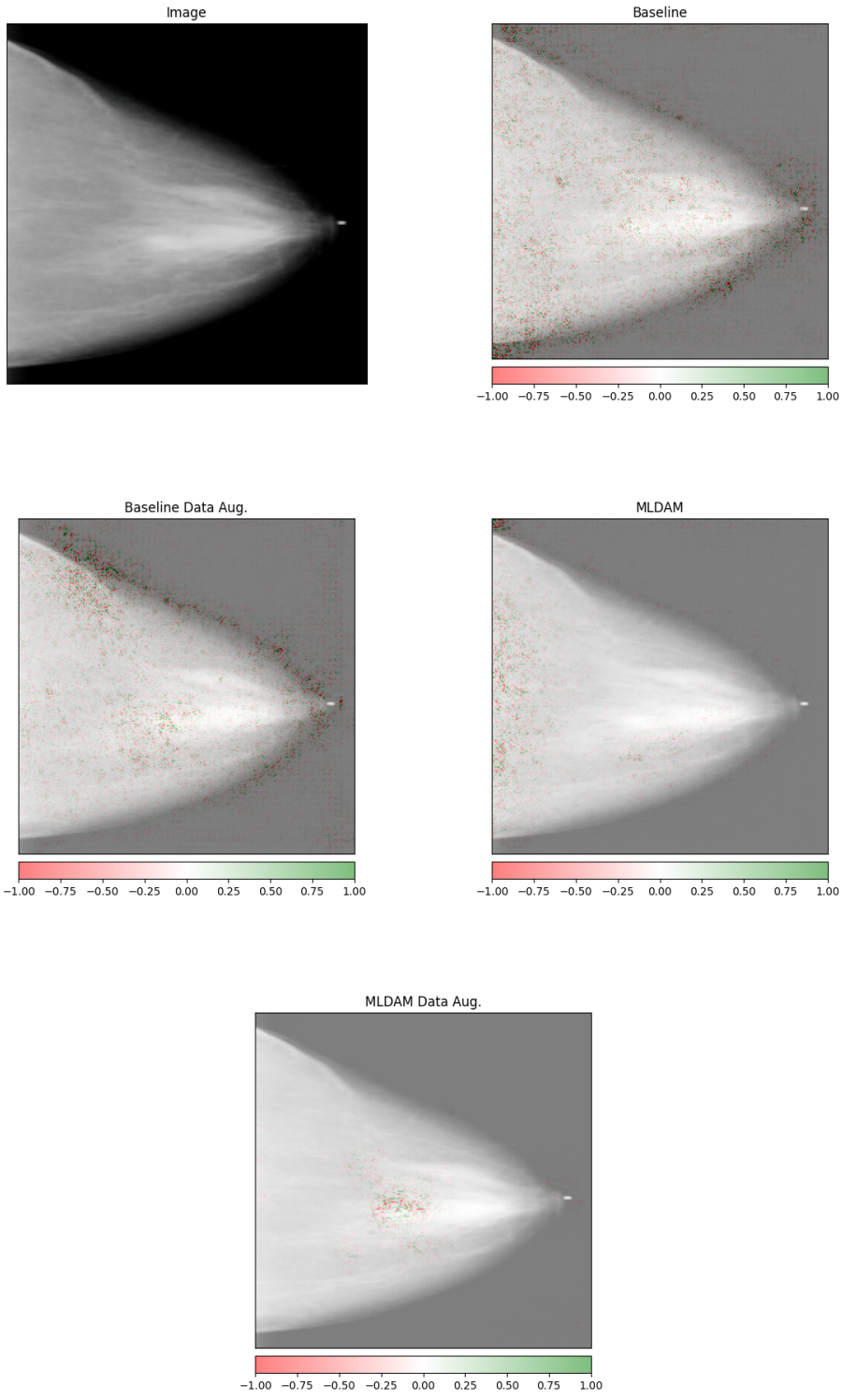

Figure 7: Examples of saliency maps obtained for an image with label "1" (*i.e.*, malignant) of the CBIS-DDSM data set, using the ResNet-50 backbone model.

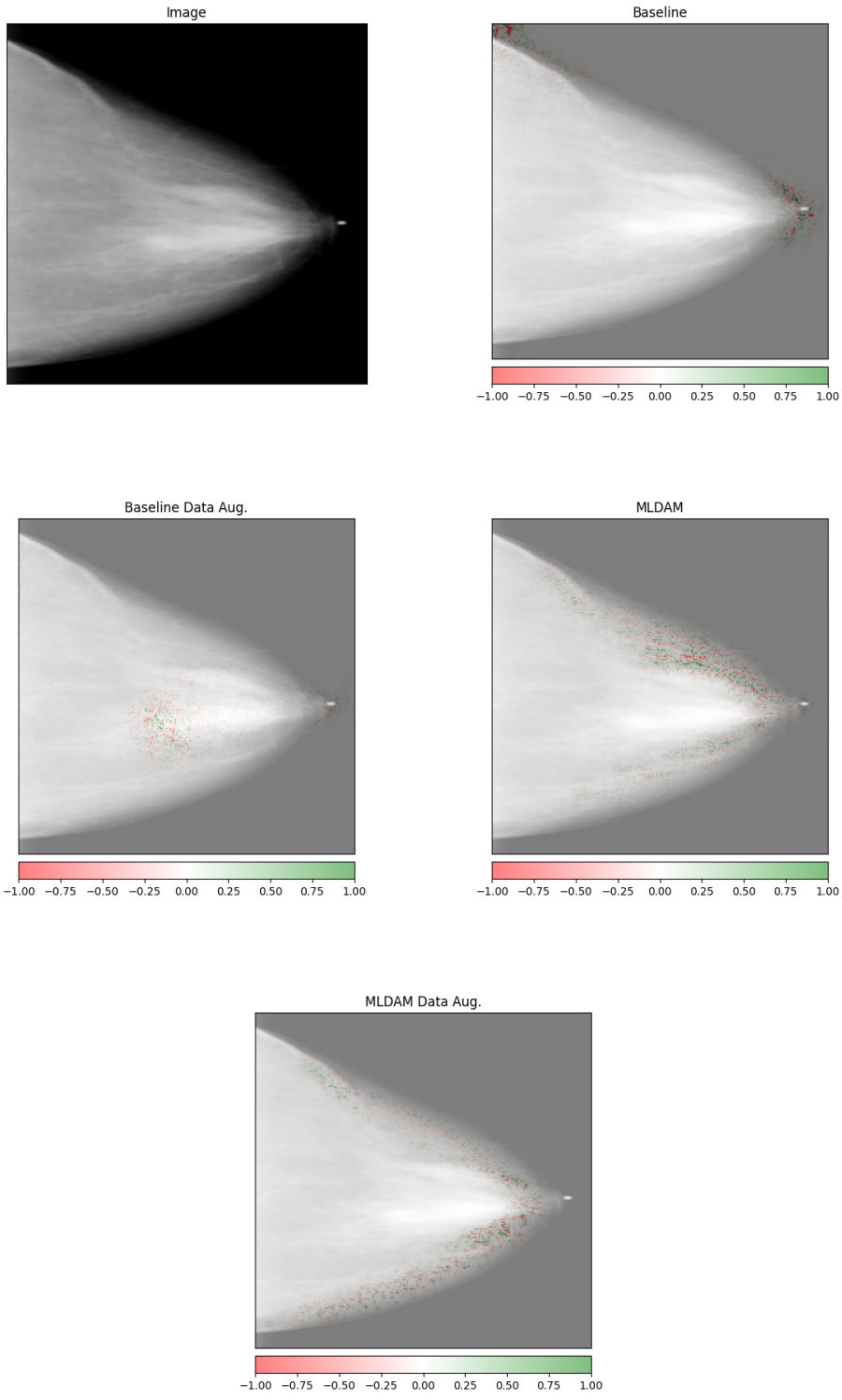

Figure 8: Examples of saliency maps obtained for an image with label "1" (*i.e.*, malignant) of the CBIS-DDSM data set, using the VGG-16 backbone model.

Table 2: Precision results obtained for the test set of the CBIS-DDSM data set.

Model	Weights	Baseline	Baseline with Data Augmentation	MLDAM	MLDAM with Data Augmentation
DenseNet-121	Training	0.6018	0.5313	0.5446	0.5000
	Validation	0.5235	0.4626	0.6165	0.5373
ResNet-50	Training	0.5922	0.5749	-	0.5125
	Validation	0.5079	0.5302	0.5400	0.4984
VGG-16	Training	0.6055	0.5285	0.5081	0.4979
	Validation	0.5440	0.6105	0.5909	0.5316

Table 3: Recall results obtained for the test set of the CBIS-DDSM data set.

Model	Weights	Baseline	Baseline with Data Augmentation	MLDAM	MLDAM with Data Augmentation
DenseNet-121	Training	0.5574	0.5574	0.5000	0.4467
	Validation	0.6393	0.4303	0.3360	0.4426
ResNet-50	Training	0.5000	0.4877	-	0.5041
	Validation	0.6557	0.6475	0.1107	0.6230
VGG-16	Training	0.5410	0.4549	0.5163	0.4959
	Validation	0.5574	0.4303	0.2664	0.3443

Depending on the backbone and the use case, these results suggest fairly different behaviours of the models. For instance, in the DenseNet-121 (Figure 3 and Figure 6), we can observe that the distribution of the saliency maps in the image gradually varies from a fuzzy distribution (*i.e.*, baseline) to a more focused distribution (*i.e.*, MLDAM with data augmentation). Similar behaviour can be observed in the ResNet-50 (Figure 4 and Figure 7), however, it is surprising to see that the presence or absence of data augmentation in the MLDAM changes the location where the model tends to focus. On the other hand, the VGG-16 (Figure 5 and Figure 8) shows an almost inverse behaviour, *i.e.*, the distribution of the saliency maps varies from a focused distribution (*i.e.*, in the baseline) to a more fuzzy distribution (*i.e.*, MLDAM with data augmentation). While in the DenseNet-121 and ResNet-50 the results are aligned with the intuition reported for the benefits of the use of attention mechanisms in DL algorithms, in the VGG-16 the results seem to contradict this assumption.

4.2 MIMIC-CXR

Table 5, Table 6, Table 7 and Table 8 present the accuracy, precision, recall and F1-score results obtained for the test set using the best model parameters for both training and validation sets applied to the use cases studied in this work. Results show that all the models perform very well, independently from the backbones, different training use cases and different best model parameters. Once again, it is not clear which is the strategy that will yield the best results.

Figure 9, Figure 10 and Figure 11 present the saliency maps obtained for an image with label "0" for all the use cases of the DenseNet-121, ResNet-50 and VGG-16 backbones, respectively.

Figure 12, Figure 13 and Figure 14 present the saliency maps obtained for an image with label "1" for all the use cases of the DenseNet-121, ResNet-50 and VGG-16 backbones, respectively.

Once again, depending on the backbone and the use case, these results suggest fairly different behaviours of the models. For instance, in the DenseNet-121, label "0", (Figure 3) we observe less dispersion of the saliency maps in the baseline use cases and a fuzzy distribution in the MLDAM use cases. On the other hand, in the DenseNet-121, label "1", (Figure 6) the saliency maps seem to be more focused in the MLDAM use cases (but we can observe similar behaviour in the baseline without data augmentation). In the ResNet-50, the model behaves similarly for both labels "0" and "1" (Figure 4 and Figure 7), where we can observe that the saliency maps are more or less focused in a specific location except for the baseline with data augmentation use case. In the VGG-16, label "0", (Figure 5) the saliency maps are present in the same location with more or less the same distribution. On the other hand, in the VGG-16, label "1", (Figure 8) we observe an irregular behaviour, with a fuzzy distribution in the baseline with data augmentation and the MLDAM use cases, and with focused distribution in the baseline and the MLDAM with data augmentation use cases.

5 Conclusions and Future Work

In this work we performed an exploratory study that aimed to study the impact of the use of attention mechanisms in the performance and the explainability of DL algorithms for medical image classification, using breast cancer detection

[7] `https://captum.ai/api/saliency.html`

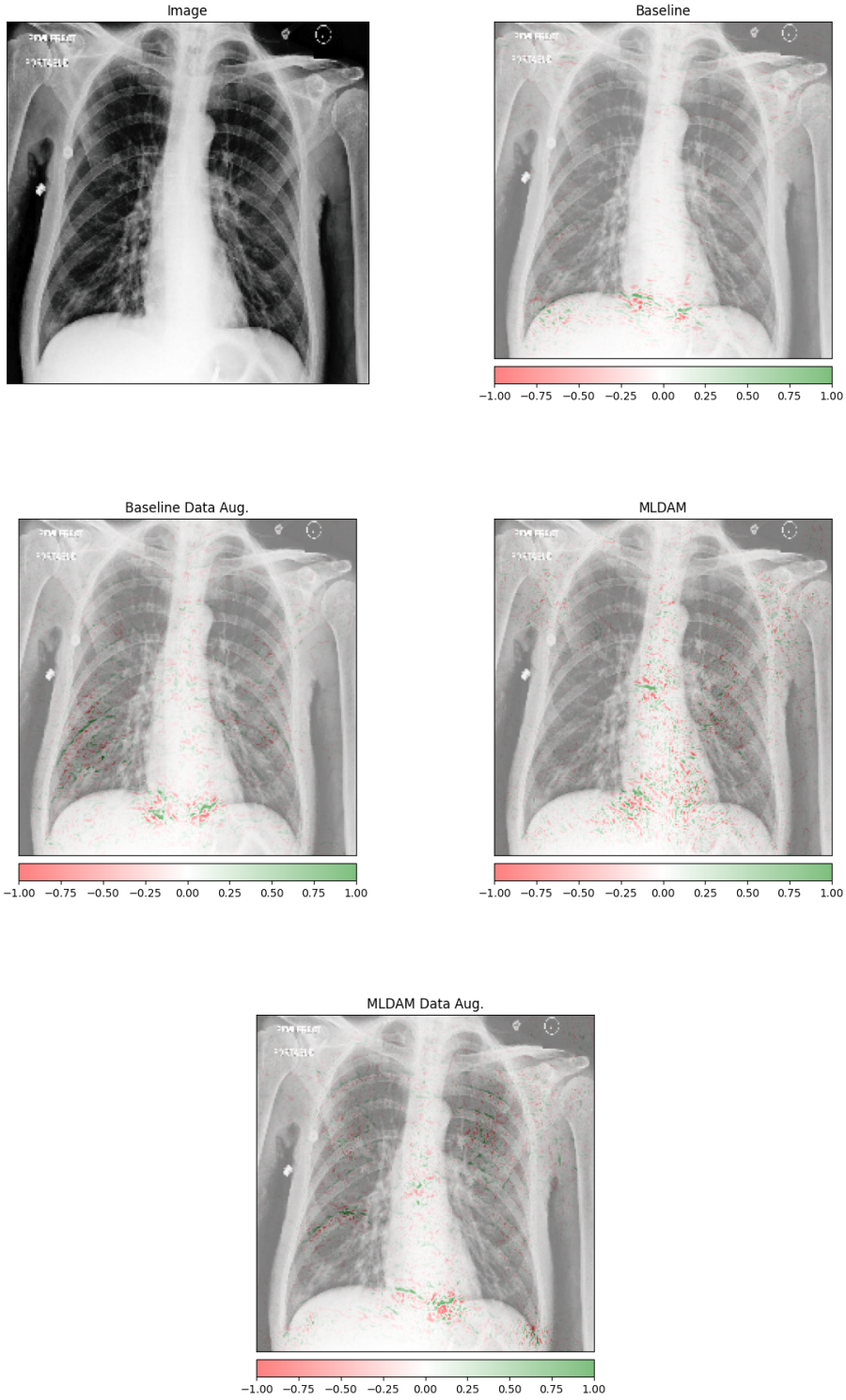

Figure 9: Examples of saliency maps obtained for an image with label "0" (*i.e.*, normal) of the MIMIC-CXR data set, using the DenseNet-121 backbone model.

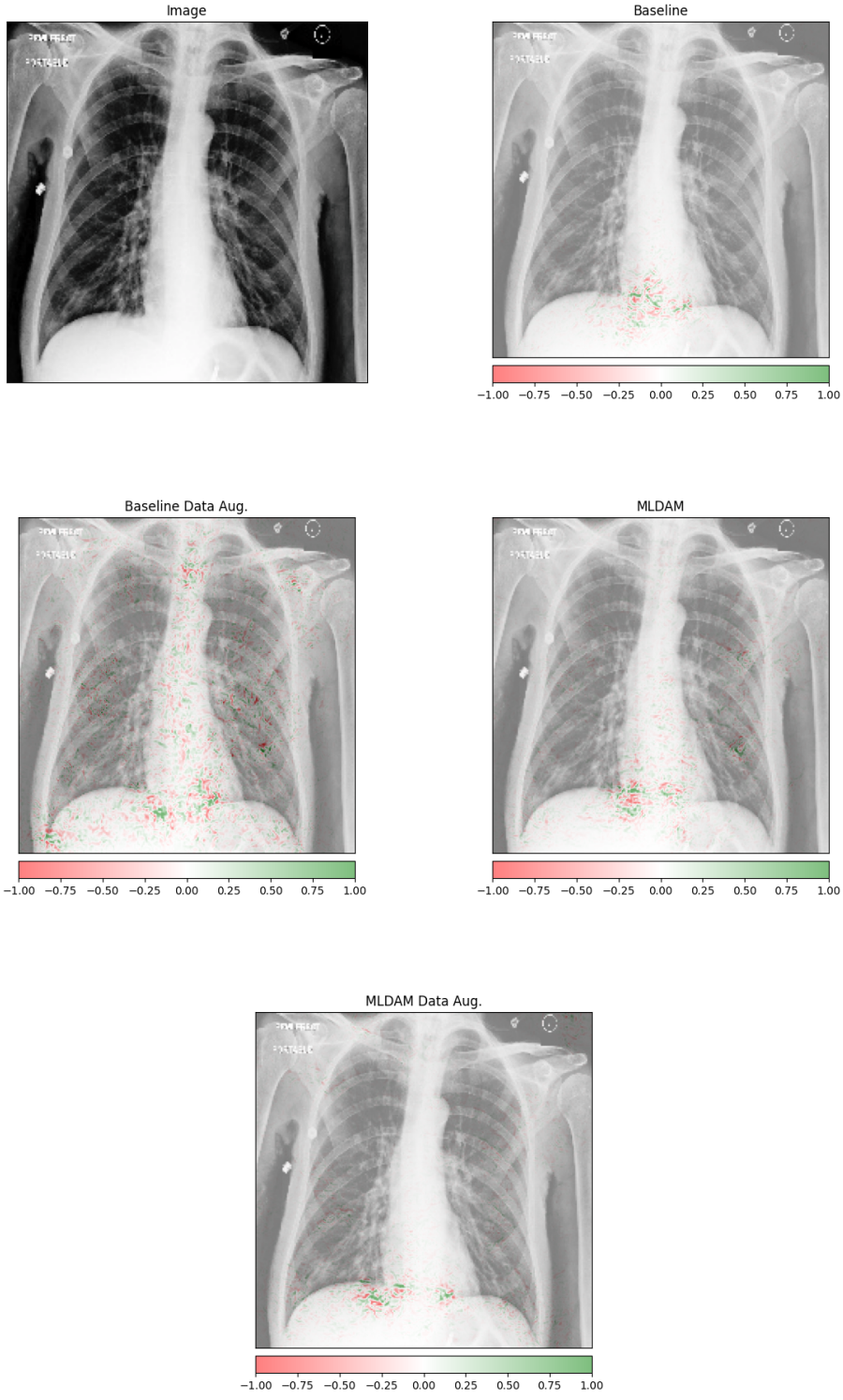

Figure 10: Examples of saliency maps obtained for an image with label "0" (*i.e.*, normal) of the MIMIC-CXR data set, using the ResNet-50 backbone model.

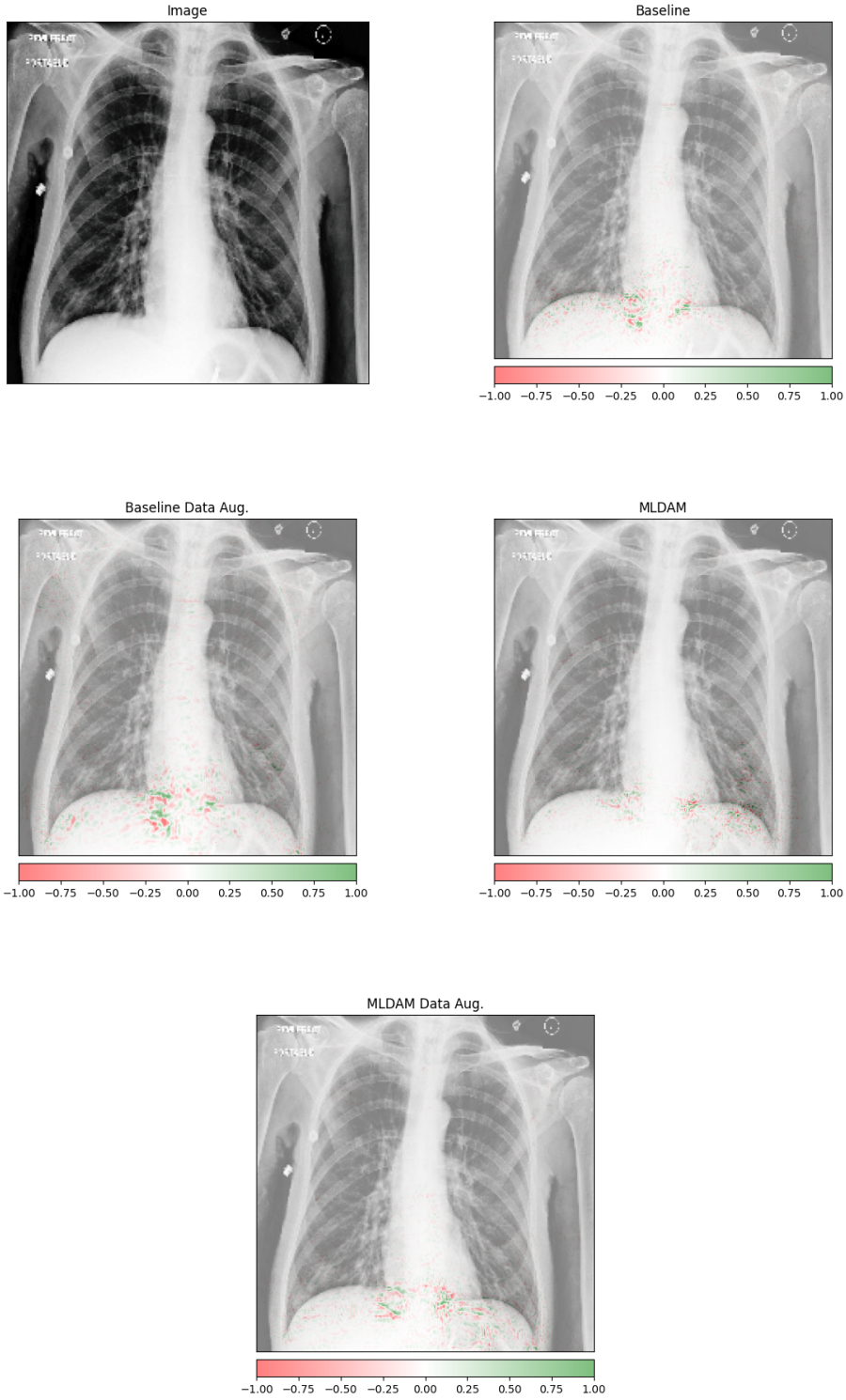

Figure 11: Examples of saliency maps obtained for an image with label "0" (*i.e.*, normal) of the MIMIC-CXR data set, using the VGG-16 backbone model.

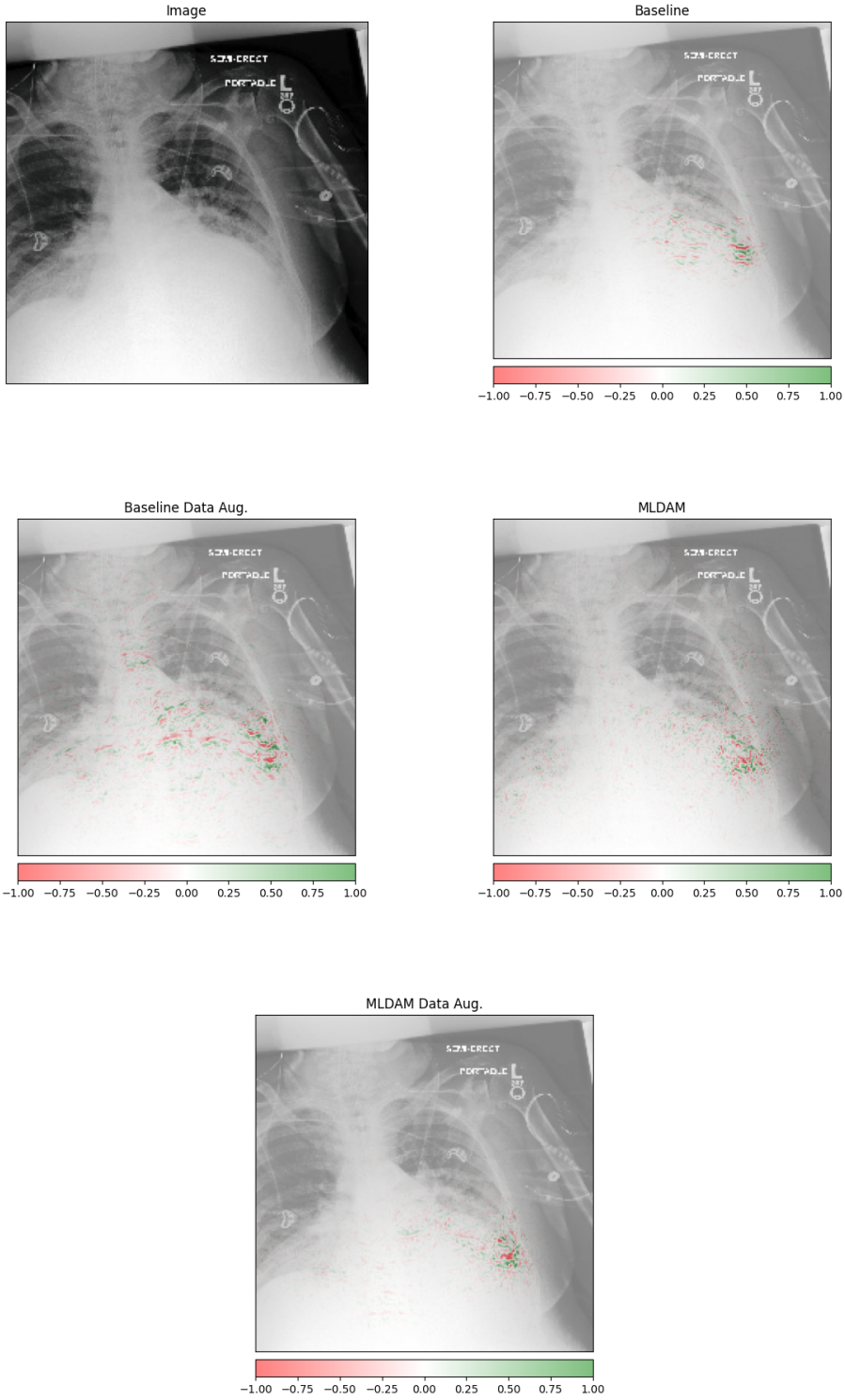

Figure 12: Examples of saliency maps obtained for an image with label "1" (*i.e.*, pleural effusion) of the MIMIC-CXR data set, using the DenseNet-121 backbone model.

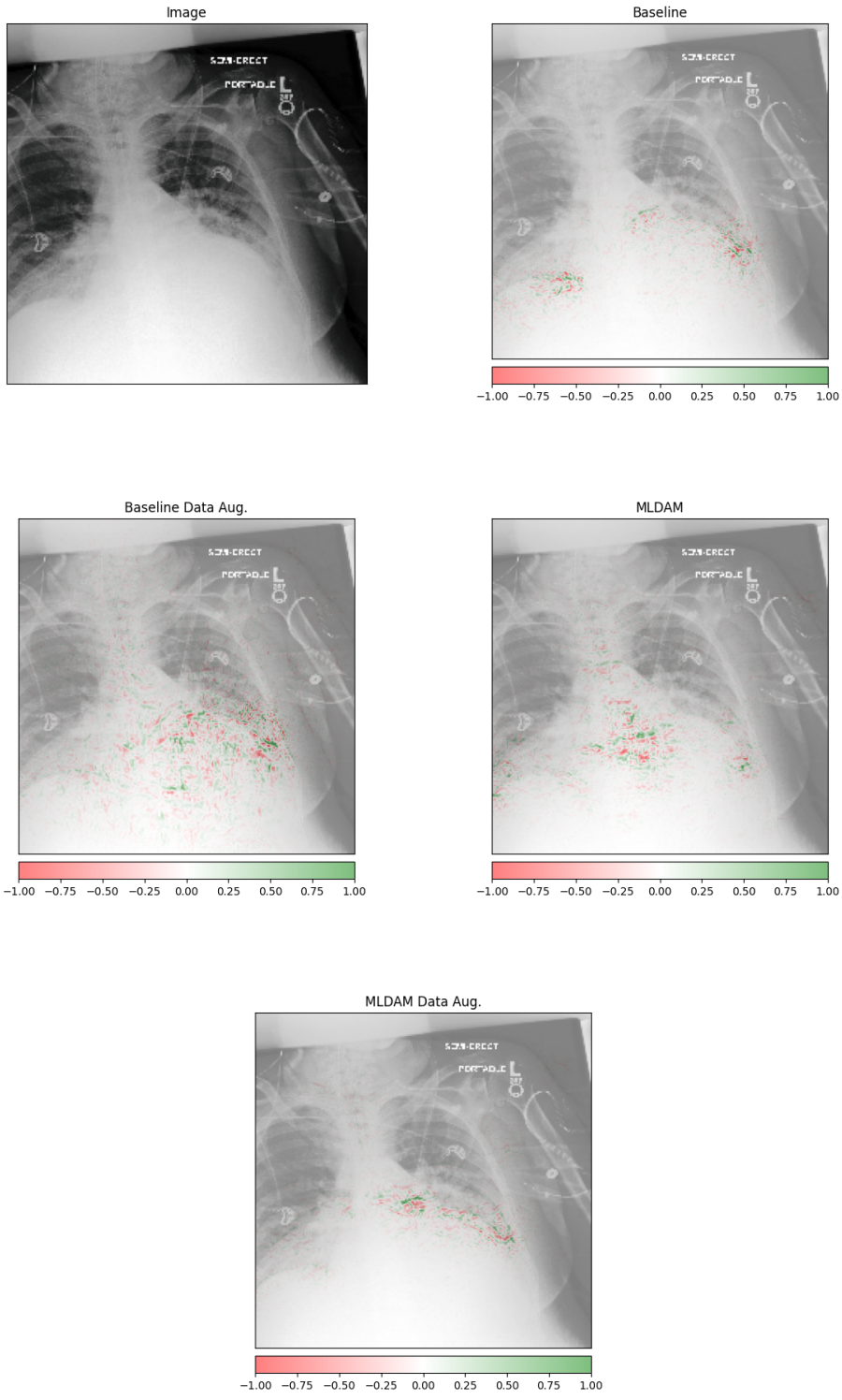

Figure 13: Examples of saliency maps obtained for an image with label "1" (*i.e.*, pleural effusion) of the MIMIC-CXR data set, using the ResNet-50 backbone model.

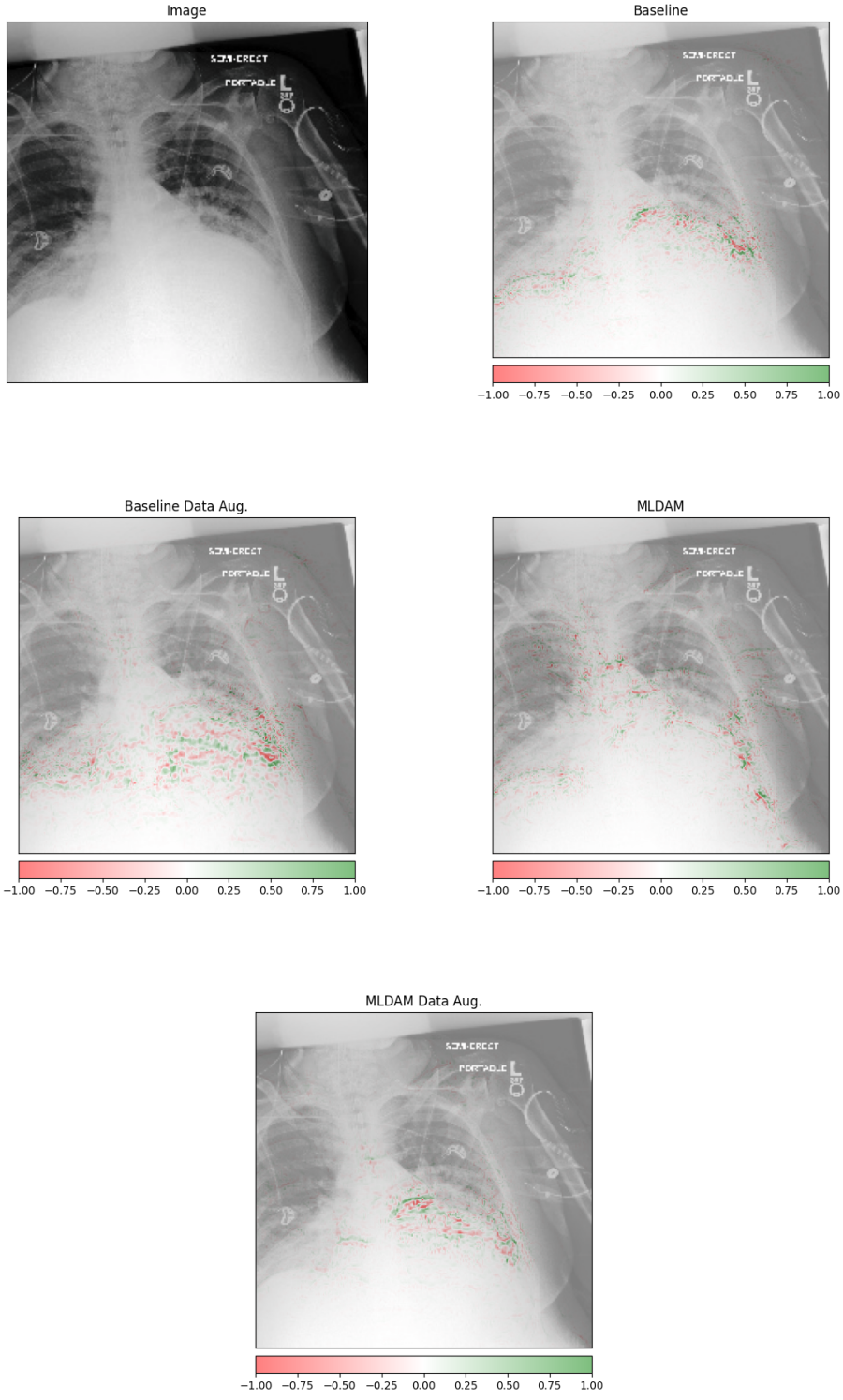

Figure 14: Examples of saliency maps obtained for an image with label "1" (*i.e.*, pleural effusion) of the MIMIC-CXR data set, using the VGG-16 backbone model.

Table 4: F1-score results obtained for the test set of the CBIS-DDSM data set.

Model	Weights	Baseline	Baseline with Data Augmentation	MLDAM	MLDAM with Data Augmentation
DenseNet-121	Training	0.5787	0.5440	0.5214	0.4719
	Validation	0.5756	0.4459	0.4350	0.4854
ResNet-50	Training	0.5422	0.5277	-	0.5083
	Validation	0.5725	0.5830	0.1837	0.5537
VGG-16	Training	0.5714	0.4890	0.5122	0.4969
	Validation	0.5506	0.5048	0.3672	0.4179

Table 5: Accuracy results obtained for the test set of the MIMIC-CXR data set.

Model	Weights	Baseline	Baseline with Data Augmentation	MLDAM	MLDAM with Data Augmentation
DenseNet-121	Training	0.8451	0.8312	0.8349	0.8386
	Validation	0.8629	0.8498	0.8535	0.8666
ResNet-50	Training	0.8340	0.8424	0.8470	0.8386
	Validation	0.8535	0.8694	0.8563	0.8414
VGG-16	Training	0.8507	0.8330	0.8293	0.8461
	Validation	0.8629	0.8731	0.8535	0.8647

in mammography images and pleural effusion detection in chest X-ray images as use cases. The best performance results are obtained in the task of pleural effusion detection in chest X-ray images, however, it is not clear that the use of an attention mechanism brought benefits. Besides, the visual analysis of the post-model saliency maps does not bring any solid conclusion towards the actual impact of the presence or absence of attention mechanism in the degree of explainability of the models. These conclusions also apply to the task of breast cancer detection in mammography images, although we acknowledge that performance metrics must be improved.

Further work should be devoted to: 1) the development of new experiences with different data processing and augmentation strategies, since it is not clear if these steps are harming the performance of the models, specially in the task of breast cancer detection in mammography images; 2) the design of different attention mechanisms that capture features from different scales or levels, since, to the authors knowledge, there isn't a clear pipeline on which are the best scales or levels that should be incorporated in a MLDAM module; 3) generate saliency maps with other methods (*e.g.*, Grad-CAM [31], Deep Taylor [37] or DeepLift [38]) to see if the results that we obtained are dependent of the post-model interpretability method or not; 4) experiment different state-of-the-art backbones (*e.g.*, MobileNets [39]) to see if their behaviour differs from the ones we used in this work; 5) try different data sets and different tasks to assess if results are data or task-dependent.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[2] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining Explanations: An Overview of Interpretability of Machine Learning," *arXiv:1806.00069 [cs, stat]*, Feb. 2019. arXiv: 1806.00069.

[3] C. Rudin, "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead," *arXiv:1811.10154 [cs, stat]*, Sept. 2019. arXiv: 1811.10154.

[4] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath, "An attentive survey of attention models," *arXiv preprint arXiv:1904.02874*, 2019.

[5] S. S. Mishra, B. Mandal, and N. B. Puhan, "Multi-level dual-attention based cnn for macular optical coherence tomography classification," *IEEE Signal Processing Letters*, vol. 26, no. 12, pp. 1793–1797, 2019.

[6] C. Chen and A. Ross, "An explainable attention-guided iris presentation attack detector," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 97–106.

[7] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[8] S. M. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafian, T. Back, M. Chesus, G. S. Corrado, A. Darzi, *et al.*, "International evaluation of an ai system for breast cancer screening," *Nature*, vol. 577, no. 7788, pp. 89–94, 2020.

[9] E. Çallı, E. Sogancioglu, B. van Ginneken, K. G. van Leeuwen, and K. Murphy, "Deep learning for chest x-ray analysis: A survey," *Medical Image Analysis*, p. 102125, 2021.

Table 6: Precision results obtained for the test set of the MIMIC-CXR data set.

Model	Weights	Baseline	Baseline with Data Augmentation	MLDAM	MLDAM with Data Augmentation
DenseNet-121	Training	0.9008	0.9000	0.9171	0.8822
	Validation	0.9122	0.8959	0.8868	0.9088
ResNet-50	Training	0.9090	0.9023	0.8983	0.9257
	Validation	0.8991	0.9083	0.8985	0.8860
VGG-16	Training	0.8916	0.8947	0.8924	0.9113
	Validation	0.8993	0.9230	0.8911	0.9013

Table 7: Recall results obtained for the test set of the MIMIC-CXR data set.

Model	Weights	Baseline	Baseline with Data Augmentation	MLDAM	MLDAM with Data Augmentation
DenseNet-121	Training	0.9071	0.8885	0.8734	0.9221
	Validation	0.9175	0.9199	0.9373	0.9268
ResNet-50	Training	0.8815	0.9013	0.9129	0.8688
	Validation	0.9210	0.9315	0.9257	0.9210
VGG-16	Training	0.9268	0.8978	0.8955	0.8955
	Validation	0.9338	0.9187	0.9315	0.9338

[10] C. Molnar, *Interpretable Machine Learning*. Christoph Molnar, 2020.

[11] Z. C. Lipton, "The Mythos of Model Interpretability," *arXiv:1606.03490 [cs, stat]*, Mar. 2017. arXiv: 1606.03490.

[12] F. Doshi-Velez and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning," *arXiv:1702.08608 [cs, stat]*, Mar. 2017. arXiv: 1702.08608.

[13] J. W. Tukey, *Exploratory data analysis*, vol. 2. Reading, Mass., 1977.

[14] K. R. Varshney, J. C. Rasmussen, A. Mojsilovic, M. Singh, and J. M. DiMicco, "Interactive visual salesforce analytics," in *ICIS*, 2012.

[15] R. L. Rivest, "Learning decision lists," *Machine Learning*, vol. 2, pp. 229–246, Nov. 1987.

[16] B. Kim, C. Rudin, and J. Shah, "The bayesian case model: A generative approach for case-based reasoning and prototype classification," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, (Cambridge, MA, USA), p. 1952–1960, MIT Press, 2014.

[17] M. Gupta, A. Cotter, J. Pfeifer, K. Voevodski, K. Canini, A. Mangylov, W. Moczydlowski, and A. van Esbroeck, "Monotonic calibrated interpolated look-up tables," *Journal Machine Learning Research (JMLR)*, 2016.

[18] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," *arXiv:1312.6034 [cs]*, Apr. 2014. arXiv: 1312.6034.

[19] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "SmoothGrad: removing noise by adding noise," *arXiv:1706.03825 [cs, stat]*, June 2017. arXiv: 1706.03825.

[20] K. Fernandes, J. S. Cardoso, and B. S. Astrup, "A deep learning approach for the forensic evaluation of sexual assault," *Pattern Analysis and Applications*, vol. 21, pp. 629–640, Aug. 2018.

[21] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," *arXiv:1311.2901 [cs]*, Nov. 2013. arXiv: 1311.2901.

[22] A. Dosovitskiy and T. Brox, "Inverting Visual Representations with Convolutional Networks," *arXiv:1506.02753 [cs]*, Apr. 2016. arXiv: 1506.02753.

[23] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network Dissection: Quantifying Interpretability of Deep Visual Representations," *arXiv:1704.05796 [cs]*, Apr. 2017. arXiv: 1704.05796.

[24] W. Silva, K. Fernandes, M. J. Cardoso, and J. S. Cardoso, "Towards Complementary Explanations Using Deep Neural Networks," in *Understanding and Interpreting Machine Learning in Medical Image Computing Applications* (D. Stoyanov, Z. Taylor, S. M. Kia, I. Oguz, M. Reyes, A. Martel, L. Maier-Hein, A. F. Marquand, E. Duchesnay, T. Löfstedt, B. Landman, M. J. Cardoso, C. A. Silva, S. Pereira, and R. Meier, eds.), vol. 11038, pp. 133–140, Cham: Springer International Publishing, 2018.

[25] W. Silva, K. Fernandes, and J. S. Cardoso, "How to produce complementary explanations using an Ensemble Model," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, July 2019. ISSN: 2161-4393.

[26] I. Y. Chen, S. Joshi, and M. Ghassemi, "Treating health disparities with artificial intelligence," *Nature Medicine*, vol. 26, pp. 16–17, Jan. 2020.

Table 8: F1-score results obtained for the test set of the MIMIC-CXR data set.

Model	Weights	Baseline	Baseline with Data Augmentation	MLDAM	MLDAM with Data Augmentation
DenseNet-121	Training	0.9039	0.8942	0.8947	0.9018
	Validation	0.9149	0.9077	0.9113	0.9178
ResNet-50	Training	0.8950	0.9018	0.9055	0.8963
	Validation	0.9099	0.9197	0.9119	0.9032
VGG-16	Training	0.9089	0.8962	0.8939	0.9033
	Validation	0.9162	0.9208	0.9108	0.9173

[27] F. Wang and D. M. Tax, "Survey on the attention based rnn model and its applications in computer vision," *arXiv preprint arXiv:1601.06823*, 2016.

[28] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, pp. 2048–2057, PMLR, 2015.

[29] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[31] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

[32] R. S. Lee, F. Gimenez, A. Hoogi, K. K. Miyake, M. Gorovoy, and D. L. Rubin, "A curated mammography data set for use in computer-aided detection and diagnosis research," *Scientific data*, vol. 4, no. 1, pp. 1–9, 2017.

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

[34] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[35] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[37] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[38] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*, pp. 3145–3153, PMLR, 2017.

[39] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.