electronics

Article

CXAI: Explaining Convolutional Neural Networks for Medical Imaging Diagnostic

Zakaria Rguibi [1,*,†], Abdelmajid Hajami [1,†], Dya Zitouni [1,†], Amine Elqaraoui [2,†] and Anas Bedraoui [1]

[1] Research Laboratory Watch Laboratory for Emerging Technologies (LAVETE), Hassan First University of Settat, Settat 21000, Morocco; abdelmajid.hajami@uhp.ac.ma (A.H.); zitouni.dya@uhp.ac.ma (D.Z.); bedraoui.a.fst@uhp.ac.ma (A.B.)
[2] National School of Computer Science and Systems Analysis (ENSIAS), Mohammed V University, Rabat 10000, Morocco; amine_elqaraoui@um5.ac.ma
* Correspondence: rguibi.fst@uhp.ac.ma; Tel.: +212-666406750
† These authors contributed equally to this work.

Abstract: Deep learning models have been increasingly applied to medical images for tasks such as lesion detection, segmentation, and diagnosis. However, the field suffers from the lack of concrete definitions for usable explanations in different settings. To identify specific aspects of explainability that may catalyse building trust in deep learning models, we will use some techniques to demonstrate many aspects of explaining convolutional neural networks in a medical imaging context. One important factor influencing clinician's trust is how well a model can justify its predictions or outcomes. Clinicians need understandable explanations about why a machine-learned prediction was made so they can assess whether it is accurate and clinically useful. The provision of appropriate explanations has been generally understood to be critical for establishing trust in deep learning models. However, there lacks a clear understanding on what constitutes an explanation that is both understandable and useful across different domains such as medical image analysis, which hampers efforts towards developing explanatory tool sets specifically tailored towards these tasks. In this paper, we investigated two major directions for explaining convolutional neural networks: feature-based post hoc explanatory methods that try to explain already trained and fixed target models and preliminary analysis and choice of the model architecture with an accuracy of 98% ± 0.156% from 36 CNN architectures with different configurations.

Keywords: explainability; convolutional neural networks; medical imaging

Citation: Rguibi, Z.; Hajami, A.; Zitouni, D.; Elqaraoui, A.; Bedraoui, A. CXAI: Explaining Convolutional Neural Networks for Medical Imaging Diagnostic. *Electronics* **2022**, *11*, 1775. https://doi.org/10.3390/electronics11111775

Academic Editors: Radu Ciorap, Jiri Hozman and Jan Vrba

Received: 26 April 2022
Accepted: 30 May 2022
Published: 2 June 2022

1. Introduction

Artificial intelligence in medical imaging is a recent development that has the potential to revolutionize the field. The ability of AI to learn and make predictions can help doctors diagnose diseases earlier and more accurately. For example, doctors are using deep learning algorithms to diagnose diseases from medical images such as X-rays and CT scans faster and more accurately than humans can [1]. In addition, scientists are working on creating robots that will be able to assist nurses in caring for patients. These robots will be able to do things such as measure patients' vital signs and report them back to nurses. This will allow nurses to spend more time with patients who need their attention the most [2].

AI can help reduce the number of false positives and negatives in medical images, which can lead to more accurate diagnoses and treatment plans, while there are some concerns about how much data AI requires in order for it to be effective, these concerns are outweighed by the benefits that this technology brings to the medical community. By working closely with physicians and other clinical professionals, we can create systems that not only produce reliable results but also provide explanations for why those results were produced. This will help build trust between doctors and AI models, leading to greater adoption rates and satisfaction levels across the board.

Since its inception, AI has grown increasingly complex, with deep learning algorithms and deep learning algorithms becoming ubiquitous in everything from user interfaces to business intelligence. However, this complexity has also made it difficult for humans to understand how these algorithms work or even why they produce certain results. This lack of explainability has led some experts to express concern that these advanced AI systems may contain unforeseen vulnerabilities or biases that could have harmful consequences [3,4].

Fortunately, there are efforts underway to make AI more understandable and accountable. For example, researchers at MIT have developed an algorithm called EXSUM: from local explanations to model understanding a mathematical framework for quantifying model understanding, and propose metrics for its quality assessment [5]. Furthermore, businesses are beginning to realize the importance of explaining their decisions not just to customers but also regulators and other stakeholders who need assurance that AI is being used responsibly [6].

Our paper aims to do two stages of explanation: the first is finding the right model while respecting a few major factors; then, a visual explanation or medical specialist can fully understand the decision made by our model. This also allows for detection of drawbacks where necessary. The rest of the paper is organized as follows: the background and the context of explainable AI in the medical setting is discussed in Section 2. Then, Section 3 covers Materials and Methods by introducing the CNN's basic and different techniques used. In Section 4, we discuss the results and the analysis of the experimental results and discussion and conclusion of this work is presented in Sections 5 and 6.

The term explaining convolutional neural networks describes the explanatory process. We will often refer to CXAI for explaining convolutional neural networks and explainable AI as XAI.

2. Background Study

There are many reasons why a doctor or other medical professional might need to understand what a CNN model is telling them. For example, if the model predicts that a patient has cancer, the doctor will want to know what specifically led the machine to that conclusion. This may be important for two reasons: first, so that the doctor can verify that the prediction is correct; and second, so that the doctor can understand which aspects of the image were most important in making the diagnosis [7].

CNN interpretability is particularly important in certain medical imaging tasks. One such task is detecting lesions in images of human skin. In this application, it's often very difficult for a human expert to determine whether or not an observed lesion represents cancer. However, if we have an interpretable CNN model trained on these images, we can ask it to explain its predictions in terms of specific image features (e.g., colouration patterns or textures) [8,9]. This information could then be used by doctors as part of their decision-making process when diagnosing patients with skin lesions [10,11].

The black-box nature of deep learning models has been a concern for many researchers and practitioners in various industries, but it is particularly worrisome in medical applications where lives may be at stake. Indeed, if something goes wrong with a decision made by an AI algorithm, it may be difficult or even impossible to determine what led to that decision and how to fix it. This lack of interpretability can also make explaining the results of ML algorithms to clinicians difficult, which can hinder adoption [12].

There are two main causes for why explainability and interpretability are barriers for AI's practical implementation in medicine: the gap between research communities and real-world medical applications, and the lack of interpretability in deep learning models.

First, there is often a disconnection between breakthrough research findings and their practical application-this is especially true in the field of AI where new techniques are rapidly evolving. For example, while some cutting-edge ML algorithms have been shown to outperform humans on specific tasks (such as object recognition), they have not yet been widely adopted by medical professionals due to concerns about reliability and safety.

Second, one major challenge facing AI development today is that most deep learning models are "black boxes"meaning that it is often very difficult (or even impossible) to understand why they produce certain results or make specific decisions. This presents a serious obstacle when trying to deploy these models into clinical settings where mistakes could potentially lead to life-threatening consequences [13].

When it comes to the application of artificial intelligence (AI) and deep learning (DL), there are two main axes: performance and understanding. The first axis is performance, which is mainly concerned with how well AI and DL can do a certain task. The second axis is that of knowledge. AI has helped research across the world with the task of inferring relations that were far beyond the human cognitive reach [14].

There are many important factors to consider when choosing a research model. In our article, we discuss two stages of explanation: the first is finding the right model while respecting a few major factors; then, a visual explanation or medical specialist can fully understand the decision made by our model. This also allows for detection of drawbacks where necessary.

3. Materials and Methods

3.1. Convolutional Neural Networks

The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. In essence, convolution allows the network to learn features in the input data by overlapping and combining small regions of it. This makes it possible for the network to identify patterns even when they are not explicitly stated in the training dataset. As such, convolutional nets often achieve better performance than traditional feedforward nets on tasks such as object recognition and classification

This sections focuses and illustrates basic technical knowledge regarding deep learning with CNNs. The CNN architecture includes several building blocks, such as convolution layers, pooling layers, and fully connected layers. A typical design comprises of numerous convolution layers and a pooling layer repeated several times, followed by one or more fully linked layers. Forward propagation refers to the process of transforming input data into output data via these levels (Figure 1).

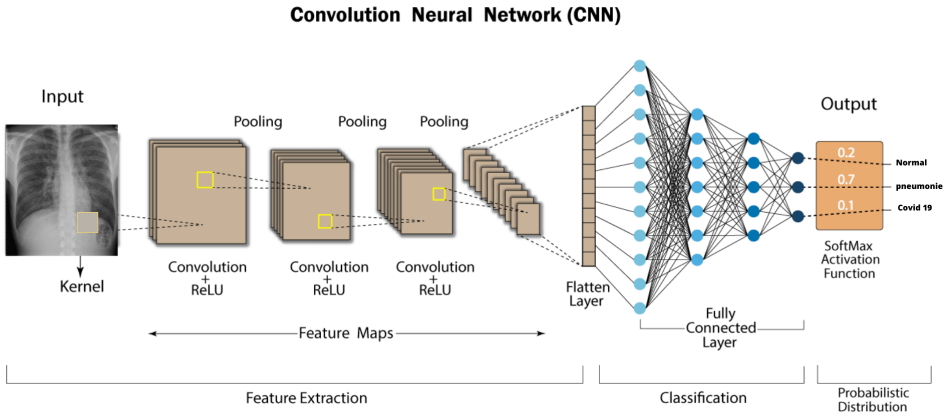

Figure 1. Convolutional networks.

3.1.1. Feature Maps Phase

One of the most important stages in CNN is the feature extraction stage (as you can see in Figure 2). This stage is responsible for identifying and extracting key features from an image. This is a critical step, as the features extracted will be used to determine if there are any anomalies in the image. A convolution layer is a core component of the CNN architecture that conducts feature extraction. It generally comprises of a combination of linear and nonlinear processes, such as convolution and activation functions.

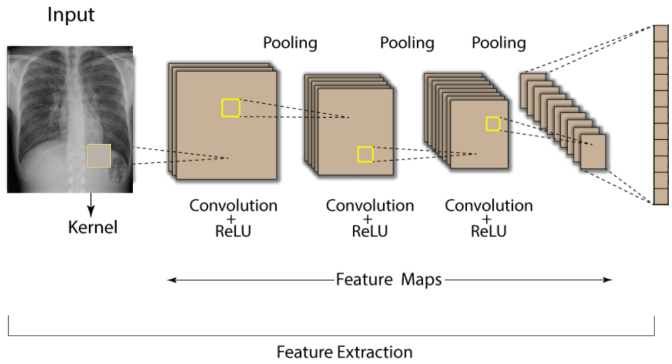

Figure 2. Feature maps phase.

3.1.2. Fully Connected Layer

The most significant step in the CNN model is feature extraction. This step is based on the mathematical property of the convolution operation, which allows us to extract features. These features are then pooled together and downsampled before being mapped into the final outputs of the network. However, it is important to note that each full connected layer in a CNN typically holds the same number of outcome nodes as the number of clusters (as you can see in Figure 3). This ensures that all information related to a certain task (such as clustering) is contained within one layer, and can be easily interpreted by a human observer. Finally, after all layers have been fully connected, a non-linear function is applied in order to produce results.

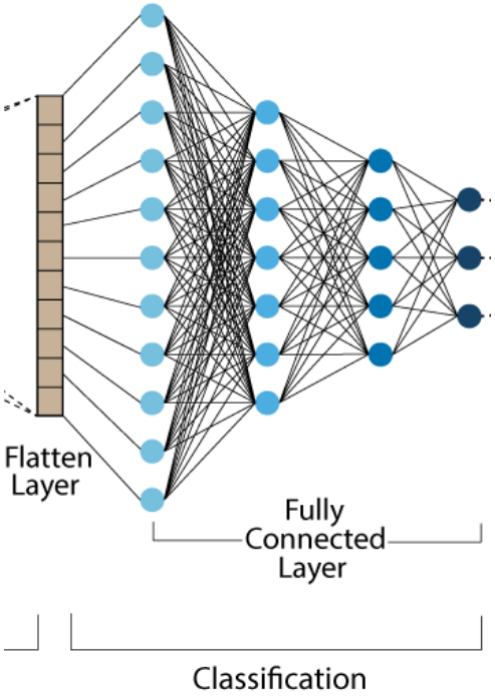

Figure 3. Fully connected layer.

3.1.3. Probabilistic Distribution

There are a variety of activation functions that can be applied to the last fully connected layer of a neural network. The most common is the linear function, but there are others that may be more appropriate for specific tasks. For example (See Figure 4), when performing multiclass classification, it is often useful to use a softmax function to normalize output values and produce target class probabilities. where each value ranges between 0 and 1 and all values sum to 1. Typical choices of the last layer are a linear or sigmoid activation function.

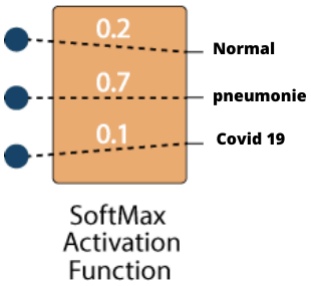

Figure 4. Probabilistic distribution.

3.2. Global Versus Local Explanation

In this section, we will address the problem of explanations specific to convolutional neural networks, but before that, we will introduce some basics of XAI. We will suppose that the CNN model is pre-trained and that the input-output relationship it implements is abstracted by a function. In the context of medical imaging, this function may take as input an image, and the output of the function may be proof of a certain medical condition or help clinicians or radiologists decide on the presence or absence of anomalies.

Many approaches have been put forth to explain deep learning predictions. We can divide them into two general categories: global and local explanations. Global explanations provide a high-level understanding of the inner workings of the entire target model. Local explanations aim to provide an explanation for the prediction of the target model on any individual instance [15].

A global explainer is a deep learning model that is designed to explain the behaviour of the entire target model, usually via distilling the target model into an interpretable one. Global explainers can be helpful in understanding how a complex deep learning model works, and can also provide instance-wise explanations as to why certain outputs were produced. However, it should be noted that global explainers are not perfect and may not always produce accurate results. Additionally, most of the current research in this area focuses on designing local explainers rather than global ones [16].

Global explanations are beneficial because they give us a broad understanding of how the target model works as a whole. However, they often lack detail and can be difficult to interpret. Local explanations are more detailed, but may not be representative of how the target model behaves on other instances [16].

$$x^* = \arg\max_x f(x) \tag{1}$$

In principle, verifying that a function has a high value only for the valid cases is an important task. However, it is difficult, if not impossible, for an interpretable model to accurately capture all the irregularities learned by a highly non-linear model. Hence, local explanations derived from global explainers might not always be accurate. The majority of the current works in the literature focus on designing local explainers.

In our medical application context the global explanation may be good as a first step to understand the model's behaviour. Global approaches focus on the interior of a model by leveraging general information about the model, training, and associated data. It attempts to describe the model's behaviour in general. Feature importance is a good example of this method, which tries to figure out the features which are in general responsible for better performance of the model among all different features. Global explainers are particularly useful when the modeller wants to understand the general mechanisms in the medical data or debug a model.

Local explanations or local interpretable methods are applicable to a single outcome of the model. This can be completed by designing methods that can explain the reason for a particular prediction or outcome. For example, it is interested in specific features and their characteristics. Specifically, we would like to know for that very example what input features contribute positively or negatively to the given prediction. These local analyses

of the decision function have received growing attention, and many approaches have been proposed. For simple models with limited non linearity, the decision function can be approximated locally as the linear function (Equation (2)) [17].

$$f(x) \approx \sum_{i=0}^{d} \underbrace{[\nabla f(\check{x})]_i \bullet (x_i - \check{x}_i)}_{R_i} \tag{2}$$

where $\check{x}$ is some nearby root point (see Figure 5).

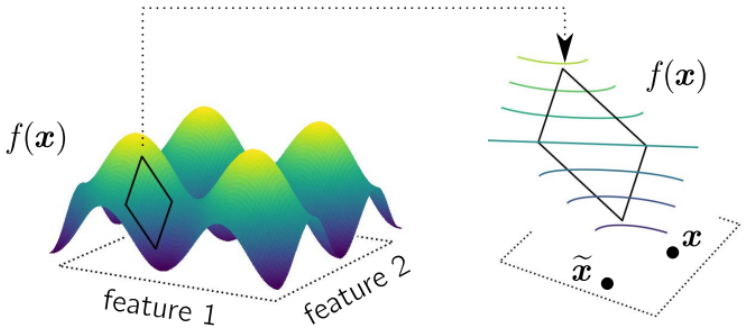

Figure 5. Nonlinear function of the input features, which produces some prediction. The function can be approximated locally as a linear model.

Many deep learning models are designed to produce a single output (such as predicting whether an email is spam or not), which can often be explained using local interpretable methods. By breaking down the model into its individual parts (e.g., looking at which input features contribute positively/negatively to predicting an outcome), we can better understand how it works and why it produces certain results.

Ultimately, which approach we use depends on our goals and what we want to learn from deep learning predictions. If we want a broad understanding of how the target model works, then global explanations are ideal. If we need more specific information about why a particular prediction was made, then local explanations are better suited for that task.

3.3. Post-Hoc versus Self-Explanatory

One of the most prominent distinctions among current explanatory methods is to divide them into two types post hoc and self explanatory methods.

Post-hoc explanatory methods are stand-alone methods that aim to explain already trained and fixed target models. For example, LIME is a post hoc explanatory method that explains a prediction of a target model by learning an interpretable model, such as a linear regression, on a neighbourhood around the prediction of the model. So, post hoc explainability can be cast as a type of "explanation-by-justification", an after-the -prediction explanation step where some evidence/information/visualisation is given to elucidate the predictions made by the AI system [18].

Another popular post hoc explanatory method is Shapley value attribution [19], which aims to identify which input features are most important for predicting the target label. This information can then be used to improve or fine tune the target model. Finally, Bayesian local interpretable model-agnostic explanations [20] combines multiple interpretable models into one overall interpretation of predictions from complex deep learning models. This approach has been found to be more accurate than single interpretable models in some cases.

Overall, post hoc explanatory methods provide valuable insights into why AI systems make certain decisions or predictions. These insights can help us improve our systems and better understand how they work.

Self-explanatory models are a relatively new development in the field of deep learning, but they have already shown great promise. At a high level, self-explanatory models have

two interdependent modules: (i) a predictor module, which is responsible for making predictions about some task or outcome, and (ii) an explanation generator module, which is responsible for providing explanations for the predictions made by the predictor. This separation of responsibilities allows self-explanatory models to be more accurate and efficient than traditional deep learning models [21].

One example of a self-explanatory model is Lei et al.'s [22] neural network model. In this model, the explanation generator selects a subset of input features that are then passed on to the predictor module. This allows the predictor to make its prediction based solely on those selected features, without being influenced by any extraneous information. The advantage of this approach is that it eliminates noise from the data and results in more accurate predictions. Additionally, because only relevant information is passed on to the predictor module, self-explanatory models are also more efficient than traditional deep learning models in terms of computational resources required

Self-explanatory models do not necessarily need to have supervision on the explanations. For example, the models introduced by Lei et al. [22,23] do not have supervision on the explanations but only at the final prediction. On the other hand, the models introduced by [23,24] require explanation-level supervision in order to generate accurate predictions.

In general, for self-explanatory models, the predictor and explanation generator are trained jointly, hence the presence of the explanation generator is influencing the training of the predictor. This is not always true for post hoc explanatory methods which do not influence at all predictions made by already trained target models. In cases where adding an explanation generator to a neural network results in significantly lower task performance than that of a neural network trained only to perform the task, it may be preferable to use latter model with a post hoc explanatory method. On the other hand it can be case that enhancing neural network with an explainer and jointly training them results in better performance on task at hand. This could potentially due to additional guidance in architecture model or extra supervision on explanations if available [25]. For example, sentiment analysis is the process of identifying and quantifying the attitude of a writer or speaker with respect to a particular topic. This task can be difficult, as the sentiment an author expresses may not be straightforward. In a study by Lei et al. (2016), it was found that adding an intermediate explanation generator module to a sentiment analysis model did not hurt performance. This suggests that giving explanations for the results of deep learning models can help improve understanding among users.

However, in [26] the authors found that using self-explaining models outperformed neural networks trained only to perform the task of sentiment analysis on common sense question answering tasks. This suggests that explanatory methods have their advantages and disadvantages depending on what task is being performed.

4. Results

4.1. The Preliminary Analysis Part

Machine learning algorithms are used to teach computers how to learn from data. Deep learning algorithms are used to create networks that can learn how to recognize patterns on their own [27]. Both machine learning and deep learning algorithms require hyperparameters tuning in order to work properly. Hyperparameters are the variables that control the behaviour of a machine-learning algorithm. They include things such as the number of layers in a neural network, the size of training datasets, and the type of optimization algorithm used. Tuning these parameters is essential for obtaining good results from an AI system [28,29].

There are many factors that go into the success or failure of a CNN model. Choosing appropriate parameters and factors is critical to good performance. In this part, we will discuss some of the most important considerations for parameter selection.

To notice the impact of each hyperparameter, we decided to train a CNN using the same data on a different set of hyperparameters, and observe the underlying change in performance and accuracy of the model. Although there are lots of hyperparameters one

can tweak, we decided to focus mainly on the number of convolution blocks, batch size and the learning rate. As you can see from these three figures (Figures 6–8), we have 36 CNN models with different configurations and measured their accuracy. The CNN model with each set of hyperparameters was trained using backpropagation with Adam optimizer. The data used to train/test this model consists of MRI Brain Scans labelled as tumourous or non-tumourous, split into train and test datasets. All models were trained using the same train dataset and tested on the same test dataset. Accuracy for such task is then calculated as such

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Accuracy is then calculated for each epoch and plotted on a graph for each set of hyperparameters to clearly differentiate the changes in performance and accuracy.

When it comes to selecting the number of convolution blocks to use in training a CNN model, there is no one-size-fits-all answer. The features or attributes of data set determine how well a CNN model will perform on new data. If the wrong features are chosen, it can lead to poor performance or even inaccurate predictions. The number of convolution blocks is directly responsible for how deep and complex the CNN model is, so the choice of the number of convolution blocks in the CNN need to reflect the complexity of the problem at hand.

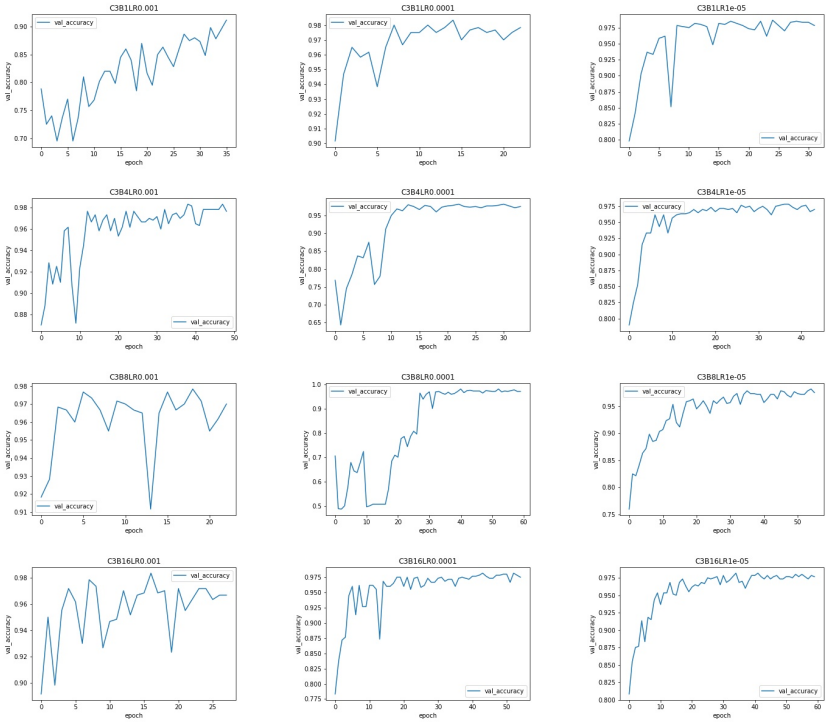

Figure 6. Three convolution block with different batch number and learning rate.

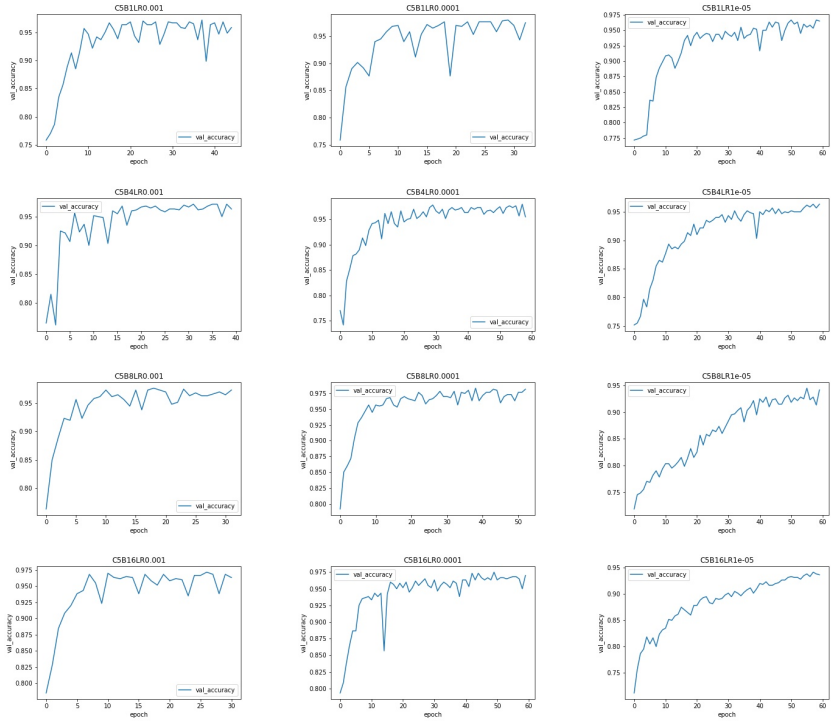

Figure 7. Five convolution block with different batch number and learning rate.

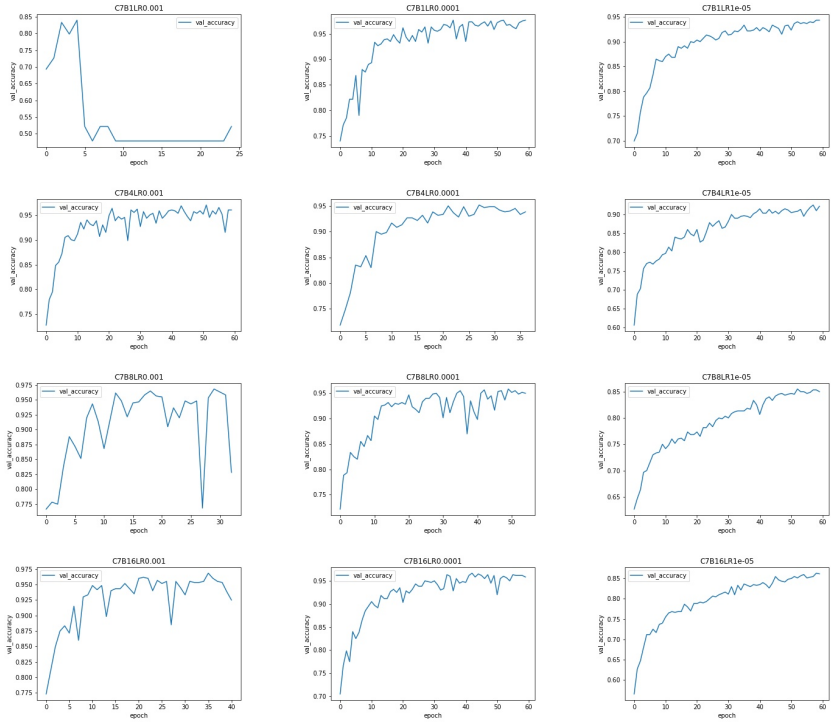

Figure 8. Seven convolution block with different batch number and learning rate.

Another critical factor is the size of the batch and the learning rate parameter. A large size of the batch set will not provide enough information for a CNN model to learn effectively, leading to poorer predictions on new data sets. However, practitioners often want to use a bigger batch size to coach their model because it allows computational speedups from the parallelism of GPUs. However, it is well-known that too large of a batch

size will result in poor generalization. This is reflected on the figures above; the validation accuracy decreases once we use a batch size too big.

Moreover, choosing the training rate is admittedly curial for the model's convergence still because the speed at which the model does so, as we see from all the figures, choosing a learning rate too big can result in divergence of the model, but on the other hand choosing a learning rate which is too small can decrease the convergence speed dramatically. The perfect learning rate for the given problem lies within the middle.

Additionally, if training datasets are not representative of actual use cases, models may be inaccurate when applied outside of the context for which they were trained. Choosing an inappropriate feature engineering strategy can also have a negative impact on model accuracy.

In this paper, we are investigate the impact of three key parameters on model performance: the initial knowledge of the AI engineer or researcher, the number of architectures used in training, and the number of settings used in each architecture.

We first trained 36 CNN models with different configurations and measured their accuracy. We then looked at how changing each parameter affected accuracy. The results were clear: the initial knowledge of the AI engineer or researcher had a big impact on explaining model behaviour; more architectures led to better accuracy and more settings led to worse accuracy.

These findings have important implications for anyone designing or using an AI model. First, it is crucial that engineers have a good understanding not just of how their models work but also why they work well (or poorly). Second, if you want your CNN model to be as accurate as possible, you need to use as many architectures as possible in your training set. Finally, it's important to carefully select only those settings that are likely to improve performance—otherwise you may end up making things worse!

4.2. TheFeature-Based Post Hoc Explanatory Methods Part

In the early days of AI, computer scientists relied on rules written by humans to program machines. However, this method was not scalable and resulted in brittle systems that could only do what they were programmed to do. In the past few years, there has been a resurgence of interest in artificial intelligence (AI) due to advances in deep learning (ML) algorithms and large amounts of data available for training models. These advances have led to some impressive achievements such as AlphaGo's victory over a world champion Go player and self-driving cars becoming more common on our streets [30].

Despite these successes, we are still far from having general AI systems that can match or exceed human intelligence across all tasks. One reason for this is that we don't really understand how these ML algorithms work or why they produce the results they do. This lack of understanding is partly due to our inability to interactively probe deep neural networks (DNNs)—the current state-of-the-art in deep learning—and see what is happening inside them at runtime. This limitation has led some researchers such as Geoff Hinton to call for a new era of AI research called "XAI" which stands for explainable artificial intelligence [31].

Medical imaging is a critical part of diagnosing and treating many medical conditions. However, the use of medical imaging can be difficult due to the material limitation and the difficulty with the use of CNN models. In our implementation, we will use the post hoc explanation to explain CNN models due to these difficulties [32].

The first challenge in using medical images for diagnosis is that there are often many different types of images that can be used for a particular condition. For example, an MRI image may show different information than an X-ray image or a CT scan image. This makes it difficult to create a model that can accurately diagnose a condition from any type of medical image.

Another challenge in using medical images for diagnosis is that current CNN models are not able to effectively learn from large amounts of data. Medical imaging often involves large files with lots of data points. Current CNN models cannot effectively learn from all

this data without becoming very complex and slow to run. This makes it difficult to create accurate diagnostic models using current technology.

In this section, we will pass to the second stage of our method- the visual explanation to help medical staff to understand the logic behind AI algorithms. After choosing the best model based in test datasets, now we will use it in this stage by using interactive visualization tools. These tools allow users to explore data sets visually while also seeing what impact changes made to individual features have on predicted classifications. To achieve the goal of our paper, we will as a mathematical base the following techniques including Integrated Gradients (IG) [33]:

- Vanilla gradients [34];
- XRAI [35];
- Blur IG [36];
- Guided IG [37].

To implement this technique we will use two medical data set for sample task classification of the presence or absence of disease. The first data set is about brain tumor classification and the second one is about classification of pneumonia diseases.

4.3. XRAI: Better Attributions through Regions

Saliency maps are an important part of computer vision and deep learning. They are used to determine which parts of an image or video are most important, and can be used for tasks such as object detection or tracking. XRAI is a new algorithm that uses region information to improve upon integrated gradients (IG), the most popular saliency map algorithm.

Let us now explain a bit about the algorithm behind the XRAI technique, as seen in Algorithm 1.

Algorithm 1 XRAI

$Given\_image I, model f and\_attributuion\_method\_g$
$Over - segement Ito\_segmentss \in S$
$Get\_attribution\_map\_A = g(f, I)$
$Let\_saliency\_maskM = 0, trajectoryT = []$
while $S \neq \emptyset$ *and* $area(M) < area(i)$ **do**
  for $s \in S$ **do**
    $Computegain^2 : g_{(s)} = \sum_{i \in s/M} \frac{A_i}{area(s/M)}$
  end for
  $s = argmax_s g_s$
  $S = S/s$
  $M = M \cup s$
  $Add\_M\_to\_list\_T$
end while
  return T

XRAI [35] makes three sets of contributions. Firstly, it presents a new region-based saliency approach based on the commonly utilized integrated gradients (IG). Importantly, XRAI may be utilized with any DNN-based model as long as the input features can be clustered into segments using any similarity metric. Second, it adds to the growing body of sanity checks for attribution methods by introducing a perturbation-based sanity check that can be used to test the reliability of an attribution method [38]. Third, it provides empirical evidence that XRAI outperforms IG in terms of both accuracy and robustness across different datasets and models

In recent years there has been a lot of research into techniques for improving image saliency detection—the process of identifying which parts of an image are most important to focus on. These techniques can be used for a variety of purposes, from medical diagnosis to object recognition in pictures posted online [39].

Several different methods have been proposed for measuring image saliency, but so far there has been no consensus on which metric is best. In our work, we propose two new metrics: accuracy information curves (AICs) and softmax information curves (SICs). We believe these metrics provide a more accurate measure of image saliency than existing methods.

As you can see in Figures 9 and 10 the XRAI heatmap explanation for brain tumour detection and classification with the top 8% of the salient of the image.

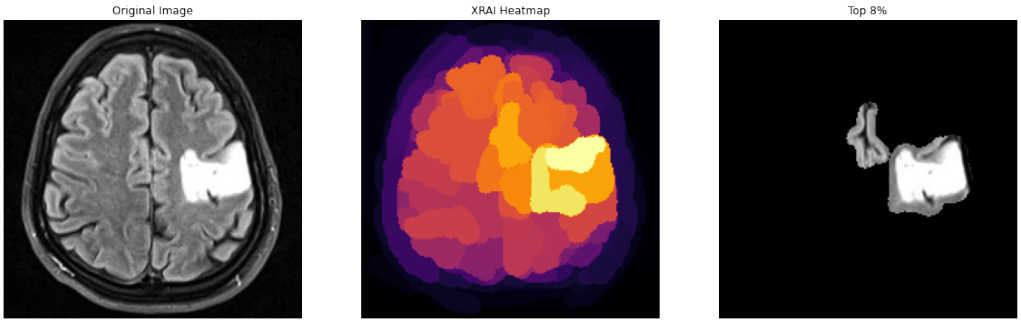

Figure 9. XRAI heatmap explanation for brain tumour detection and classification with the top 8% of the salient of the image.

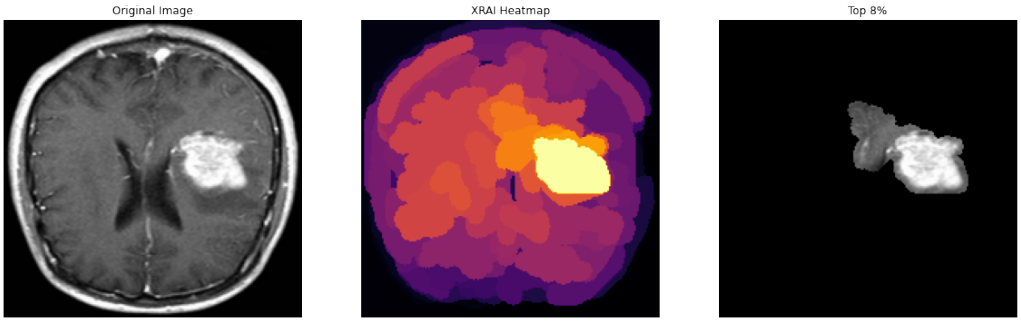

Figure 10. XRAI heatmap explanation for brain tumour detection and classification with the top 8% of the salient of the image.

4.4. Vanilla Integrated Gradients

Now we will show the next two technique that allow us to demonstrate the CXAI the vanilla integrated gradients and SmoothGrad integrated gradients.

The original integrated gradients technique has demonstrated its usefulness in debugging networks, extracting rules from a network, and enabling users to engage with models better. The two fundamental axioms—sensitivity and implementation invariance—that attribution methods ought to satisfy have been found to hold for integrated gradients [26].

The vanilla integrated gradients technique builds on the original by adding the ability to debug networks more effectively. In particular, it can identify which neurons are activated or suppressed when a given input is applied. This information can help improve the design of neural networks by revealing which neurons are important for achieving a desired outcome.

The SmoothGrad integrated gradients technique further enhances the vanilla approach by incorporating an algorithm that smooths gradients as they propagate through the network. This helps avoid any sudden changes in activation or suppression that could occur with traditional gradient descent techniques. As such, it leads to more stable and accurate results when training neural networks. The results from Figure 11 show that the classifier is able to correctly identify the part of the pixel used to make its decision. This information can help us to improve our classifier's accuracy and ensure that it is making accurate decisions. Additionally, this data can be used to develop new methods for improving our classifier's performance.

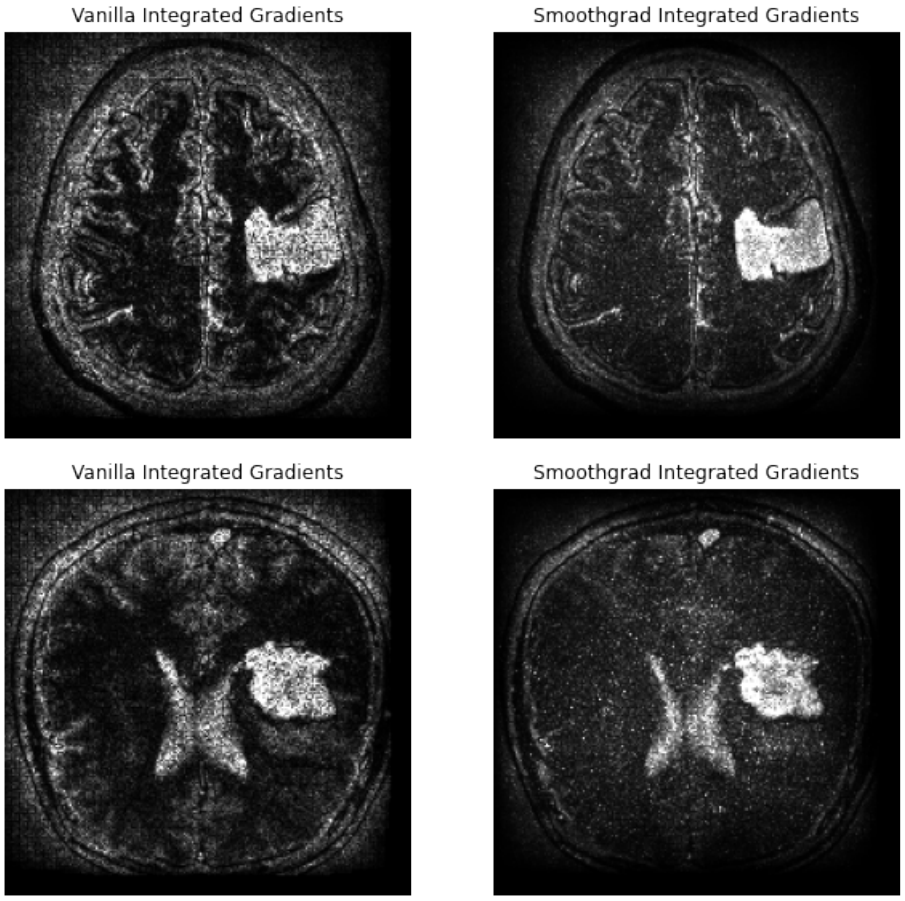

Figure 11. Vanilla IG results.

Vanilla gradient is an approach to pixel assignment that was pioneered by Simonyan et al. It is based on backpropagation, which gives us a map of the size of the input features with negative to positive values. This makes it easy to determine how much change is needed in each pixel in order to achieve the desired outcome. Why is this important? Because it makes creating accurate images much easier than ever before! With vanilla gradient, you can fine-tune your images until they look perfect—no more guesswork involved. Plus, because the algorithm is so efficient, you can apply it even when working with large images files.

The following is the key to this method:

1. Make a forward pass of the problematic picture.
2. Determine the gradient of the class of interest's score relative to the input pixels:

$$E_{grad}(I_0) = \frac{\delta S_c}{\delta I}|_{I=I_0} \tag{4}$$

3. Picture the gradients. You have the option of displaying absolute figures or highlighting negative and positive contributions individually.

In more technical terms, we have an image I, and the convolutional neural network assigns it a score $S\_c(I)$ for class c. The rating of our image is a very non-linear function. The concept behind using the gradient is that we can use a first-order Taylor expansion to approximate this score.

$$S_c(I) \approx w^T I + b \tag{5}$$

where w is the derivate of our score:

$$w = \frac{\delta S_C}{\delta I}|_{I_0} \tag{6}$$

4.5. Guided Integrated Gradients

Guided integrated gradients models can be used to explain the observed data and help to improve the quality of images generated by a rendering system. However, when these models are used in a path-based global illumination (GI) system, they can often produce noisy results due to accumulation of errors along the GI path. In this paper, we will try to implement the guided IG technique in order to adapt the attribution path itself—by conditioning the path not only on the image but also on the explained model—in order to mitigate the effect of noise accumulation along the GI path. Empirical evidence suggests that guided GI produces saliency maps that are more closely aligned with the model prediction and input image being explained.

To reduce the influence of attribution accumulation in high gradient directions unrelated to the input, we are willing to build a path that avoids (input) locations that generate anomalies in the model's behaviour. This is called $(\ell noise)$, and one method for reducing it is as follows:

$$\gamma^{F*} = arg \min_{\gamma^F \in T} \ell_{noise} \tag{7}$$

$$\ell_{noise} = \sum_{i=1}^{N} \int_{\alpha=0}^{1} \left| \frac{\partial F(\gamma^F(\alpha))}{\partial \gamma_i^F(\alpha)} \frac{\partial \gamma_i^F(\alpha)}{\partial \alpha} \right| d\alpha \tag{8}$$

We can calculate a $(\ell noise)$ value for each pixel in an image by taking the gradient magnitude at each pixel, and dividing it by the standard deviation of all gradient magnitudes within a neighbourhood around that pixel. The smaller this value is, the less likely it is that any given direction will contribute significantly to model error. We can then use this value to weight our input images when training our models, so that those areas with higher $(\ell noise)$ values contribute less towards model error than those with lower values.

In order to reduce the amount of noise in our data, we need to first understand what contributes to the noise. By minimizing ℓ_{noise} at every feature (pixels in an image, for example), we can hopefully avoid high gradient directions. However, before we can define $\gamma^F(\alpha)$ precisely, optimizing the above objective requires knowing the prediction surface of the neural network F at every point in the input space. This is infeasible and therefore a greedy approximation method called guided integrated gradients is proposed instead.

This approach approximates the prediction surface using a low-dimensional subspace that is learned during training. The gradient along each dimension of this subspace is computed using a simple linear regression model. These gradients are then combined into a single gradient vector and used as input to optimize $\gamma^F(\alpha)$. Experiments show (as you can see in Figure 12) that this approach achieves good performance with minimal parameter tuning.

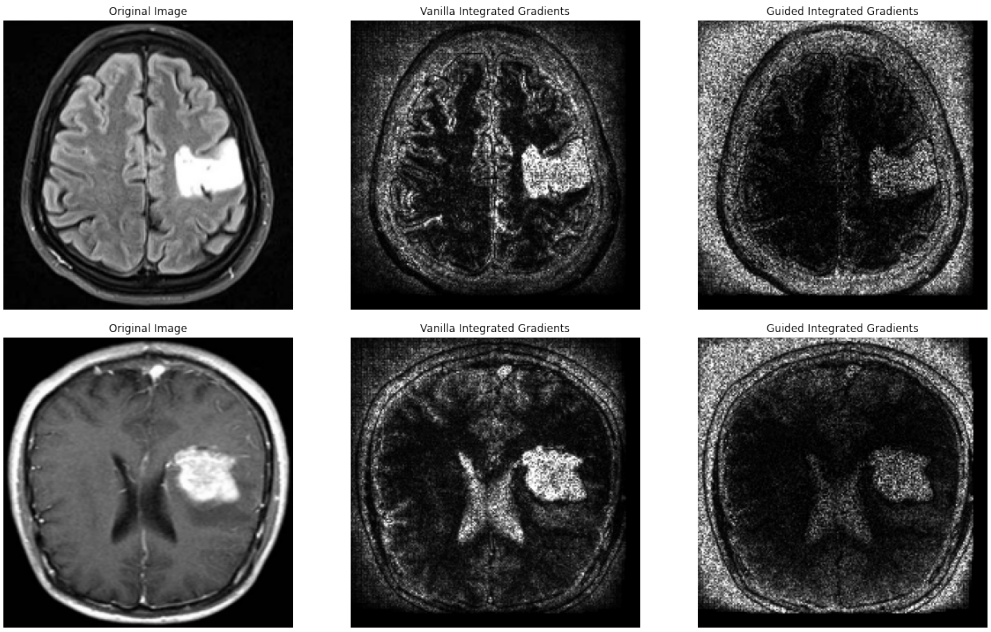

Figure 12. Guided integrated gradients results.

4.6. Blur Integrated Gradients

The attributions in computer vision identify the value of each pixel to the forecast. Perception tasks vary from other tasks in that the fundamental characteristics, such as pixels or time points, are never important on their own; instead, information is usually always stored in higher level features such as textures, edges, or frequencies. The classifier decision in Figure 13 is based on the pixel at the bottom-left corner. This part of the pixel has good results for distinguishing between classes, which can help us to well identify it.

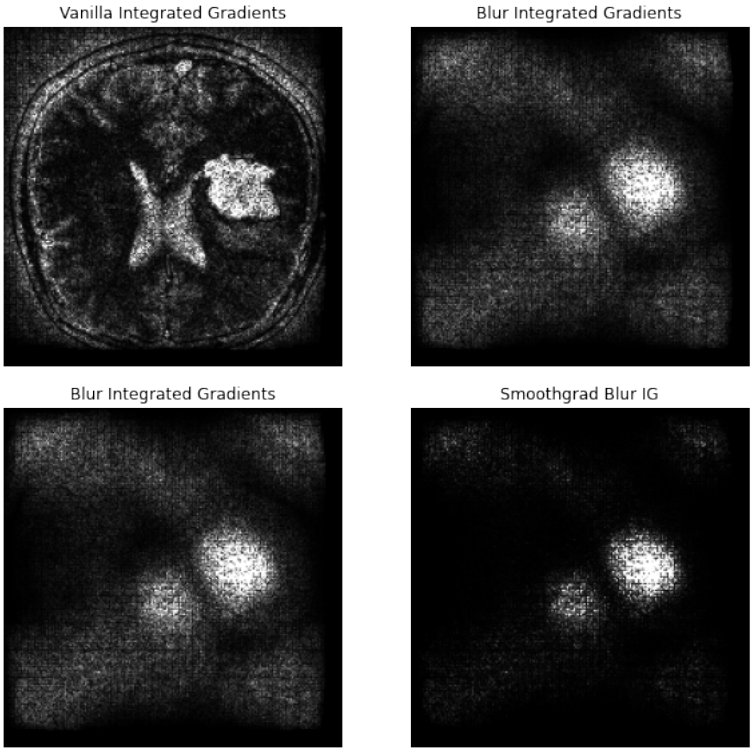

Figure 13. Blur integrated gradients and SmoothGrad IG results.

BlurIG [36] extends the integrated gradients technique. Formally, suppose we have a function $F : R_{m*n} \rightarrow [0,1]$ that represents a deep network.

Specifically, let $z(x,y) \in R^{m*n}$ be the 2D input at hand, and $z'(x,y) \in R^{m*n}$ be the 2D baseline input, meant to represent an informationless input.

We consider the straightline path from the baseline z′ to the input z

$$\gamma(x,y,\alpha) = z'(x,y) + \alpha * (z(x,y) - z'(x,y)) \tag{9}$$

Integrated gradients are obtained by accumulating these gradients.

$$IG(x,y) = (z(x,y) - z'(x,y)) * \int_{\alpha=0}^{1} \frac{\partial F(\gamma(x,y,\alpha))}{\partial \gamma(x,y,\alpha)}\, d\alpha \tag{10}$$

Let

$$L(x,y,\alpha) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \frac{1}{\pi * \alpha} e^{\frac{x^2+y^2}{\alpha}} z(x-m, y-n) \tag{11}$$

be the discrete convolution of the input signal with 2D Gaussian kernel with variance α (also known as the scale parameter)

BlurIG is obtained by accumulating the gradients along the path defined by varying the α parameter:

$$BlurIG(x,y) = \int_{\alpha=\infty}^{0} \frac{\partial F(L(x,y,\alpha))}{\partial L(x,y,\alpha)} \frac{\partial L(x,y,\alpha)}{\partial \alpha}\, d\alpha \tag{12}$$

Implementation wise, the integral can be approximated using a Riemann sum:

$$BlurIG(x,y) = \sum_{i=1}^{s} \frac{\partial F(L(x,y,\alpha_i))}{\partial L(x,y,\alpha_i)} \frac{\partial L(x,y,\alpha_i)}{\partial \alpha_i} \frac{\alpha_{max}}{s} \tag{13}$$

where $\alpha_i = i\frac{\alpha_{max}}{s}$ ans s is the number of steps in the Riemann approximation

Attributions and explanations are based on perturbations. BlurIG prescribes a specific set of perturbations to the input (gradient computation). If the perturbations destroy 'information', then the resultant change in prediction can be interpreted as feature importance; this is the desired interpretation. However, if the perturbation creates information, then the resultant change in score is not due to a feature present in the input, and the result will be a misleading, uninterpretable explanation.

CNN explanation is a powerful tool that can be used to improve the performance of deep neural networks. By identifying the importance of each pixel to the prediction, CNN explanation can help us understand how these networks work and why they produce certain results. This information is essential for debugging and improving the performance of these networks.

5. Discussion

There is a lot of talk about the role of artificial intelligence (AI) in radiology and how it will shape the future of the field, while there is no doubt that AI can play an important role in helping to improve radiologic diagnosis, it is important to understand that this is not a simple task. Proper integration of AI solutions into the next future radiology workflow requires a deep understanding of both the medical and scientific background behind disease detection [40].

One thing that often gets lost in all the discussion around AI is just how complex image classification and detection really are. There are many factors that need to be considered when making a diagnosis, including patient history, physical examination findings, and laboratory results as well as images. It is not simply enough for an algorithm to be able to correctly identify lesions on images; it must also take into account all other relevant information in order to make an accurate diagnosis.

This complexity means that those working with AI solutions need to have a strong understanding not only of computer science but also of medicine and pathology. Only by having this comprehensive knowledge can they properly integrate these solutions into radiologic workflows and help ensure accuracy in diagnoses.

There are many reasons why in our work we believe that it is very important to have two-stage of explanation. The first one is how we can choose the best architecture means the best accuracy than in the second a visual explanation to understand their decision but also their drawbacks.

Choosing a CNN architecture for our project is not an easy task. It is like making a choice between several different paths, all of them with advantages and disadvantages. We need to be sure that whatever decision we make will bring us closer to our goal and will not cause any problems along the way. That is why having more than one stage of explanation is so important—it allows us to make informed decisions based on accurate information. The first stage of explanation is analytical—it helps us understand what each option offers and how well it meets our needs. This part is crucial, because if we do not know what each path entails, we cannot possibly choose the right one for us. It is like trying to find your way in the dark—you might stumble upon something interesting, but you are also likely to end up lost or injured . The second stage of explanation provides a visual representation of each option. This helps us see not only how well they meet our needs, but also their potential drawbacks. For example, if two options seem equally good on paper, this stage can help us decide which one would be better suited for our project based on its specific requirements . Having two-stage of explanations allow us not only choose the right architecture but also to understand their effectiveness and why they were decided up on interest place.

In this paper, we investigated two major directions for explaining convolutional neural networks: feature-based post hoc explanatory methods and preliminary analysis for choice of the model architecture. For both directions, we investigated the question of verifying a good architecture and why it is good. Post hoc explanations describe the decision-making processes of the models that they aim to explain, while preliminary analysis focuses on choosing an appropriate model architecture. We found that both approaches are necessary for understanding how convolutional neural networks work; post hoc explanations help us understand why a particular network performs well on a task, while preliminary analysis can help us choose an appropriate network structure from among many possible alternatives.

Our future work will focus on integration into a radiology unit in order to study their impact on the radiology workflow in a real case. This will allow us to gain an understanding of how our work can be integrated into the medical field as a whole and help us to continue making progress in this area.

6. Conclusions

The objective of our work is to put at the disposal of any beginner researchers, students and teachers a tool of explanation that allows us to understand the bases of this field of research and to show the importance of this kind of work. In our work, we obtained good results that help us to understand how the CNNs work. The convolutional neural network (CNN) is a model that has been found to be very effective for image recognition with an accuracy of 98% ± 0.156%. Through our experiments, we have found that there are a number of factors that control the behaviour and effectiveness of this model, from 36 CNN architectures with different configurations, we choose the best one for the next level of explanation based on the performance metric in the first step. The first stage is important because it determines which research model is most appropriate for the data at hand. The main considerations are number of convolution blocks, batch size, and the learning rate. After selecting an appropriate model, we can move on to stage two—explaining the results in detail. Medical specialists often need visuals to help them understand complex models and results. Our article includes several figures that illustrate how our approach works

in practice. We believe that these visuals will be helpful for both researchers and medical specialists alike.

Author Contributions: Conceptualization, Z.R.; methodology, Z.R.; software, Z.R. and A.E.; validation, Z.R., A.H. and D.Z.; formal analysis, Z.R.; investigation, Z.R.; resources, Z.R.; data curation, Z.R.; writing—original draft preparation, Z.R.; writing—review and editing, Z.R. and A.E.; visualization, Z.R.; supervision, A.H. and D.Z.; project administration, Z.R. and A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to licensing agreements.

Acknowledgments: We appreciate LAVETE Team and faculty staff.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
DNN	Deep neural networks
IG	Integrated Gradients
XRAI	Better Attributions Through Regions

References

1. Zhang, F. Application of machine learning in CT images and X-rays of COVID-19 pneumonia. *Medicine* **2021**, *100*, e26855. [CrossRef] [PubMed]
2. Christoforou, E.G.; Avgousti, S.; Ramdani, N.; Novales, C.; Panayides, A.S. The Upcoming Role for Nursing and Assistive Robotics: Opportunities and Challenges Ahead. *Front. Digit. Health* **2020**, *2*, 585656. [CrossRef] [PubMed]
3. Yampolskiy, R. Unexplainability and Incomprehensibility of AI In the domain of AI safety, the more accurate the explanation is, the less comprehensible it is. *Artif. Intell.* **2020**, *7*, 277–291.
4. Crockett, L. *The Turing Test and the Frame Problem: AI's Mistaken Understanding of Intelligence*; Intellect Books: Bristol, UK, 1994.
5. Zhou, Y.; Ribeiro, M.T.; Shah, J. ExSum: From Local Explanations to Model Understanding. *arXiv* **2022**, arXiv:2205.00130.
6. Oxborough, C.; Cameron, E.; Rao, A.; Birchall, A.; Townsend, A.; Westermann, C. Explainable AI: Driving Business Value through Greater Understanding. Retrieved from PWC. 2018. Available online: https://www.pwc.co.uk/audit-assurance/assets/explainable-ai.pdf (accessed on 25 April 2022).
7. Samek, W.; Montavon, G.; Lapuschkin, S.; Anders, C.J.; Müller, K.R. Explaining deep neural networks and beyond: A review of methods and applications. *Proc. IEEE* **2021**, *109*, 247–278. [CrossRef]
8. Iqbal, M.S.; Ahmad, I.; Bin, L.; Khan, S.; Rodrigues, J.J. Deep learning recognition of diseased and normal cell representation. *Trans. Emerg. Telecommun. Technol.* **2020**, *32*, E4017. [CrossRef]
9. Iqbal, M.S.; El-Ashram, S.; Hussain, S.; Khan, T.; Huang, S.; Mehmood, R.; Luo, B. Efficient cell classification of mitochondrial images by using deep learning. *J. Opt.* **2019**, *48*, 113–122. [CrossRef]
10. Shao, Y.; Cheng, Y.; Shah, R.U.; Weir, C.R.; Bray, B.E.; Zeng-Treitler, Q. Shedding light on the black box: Explaining deep neural network prediction of clinical outcomes. *J. Med. Syst.* **2021**, *45*, 1–9. [CrossRef]
11. Strzelecki, M.H.; Strąkowska, M.; Kozłowski, M.; Urbańczyk, T.; Wielowieyska-Szybińska, D.; Kociołek, M. Skin Lesion Detection Algorithms in Whole Body Images. *Sensors* **2021**, *21*, 6639. [CrossRef]
12. Molnar, C. Interpretable Deep Learning: A Guide for Making Black Box Models Explainable, 2nd ed. 2022. Available online: https://christophm.github.io/interpretable-ml-book (accessed on 25 April 2022).
13. Arbelaez Ossa, L.; Starke, G.; Lorenzini, G.; Vogt, J.E.; Shaw, D.M.; Elger, B.S. Re-focusing explainability in medicine. *Digit. Health* **2022**, *8*, 20552076221074488. [CrossRef]
14. Gilpin, L.H.; Bau, D.; Yuan, B.Z.; Bajwa, A.; Specter, M.; Kagal, L. Explaining explanations: An overview of interpretability of machine learning. In Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), Turin, Italy, 1–3 October 2018.

15. Rguibi, Z.; Hajami, A.; Dya, Z. Explaining Deep Neural Networks in medical imaging context. In Proceedings of the 2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA), Tangier, Morocco, 30 November–3 December 2021.
16. Huber, T.; Weitz, K.; André, E.; Amir, O. Local and global explanations of agent behavior: Integrating strategy summaries with saliency maps. *Artif. Intell.* **2021**, *301*, 103571. [CrossRef]
17. Plumb, G.; Molitor, D.; Talwalkar, A.S. Model agnostic supervised local explanations. *Adv. Neural Inf. Process. Syst.* **2018**, *31*.
18. Vale, D.; El-Sharif, A.; Ali, M. Explainable artificial intelligence (XAI) post-hoc explainability methods: Risks and limitations in non-discrimination law. *AI Ethics* **2022**, 1–12. [CrossRef]
19. Papapetrou, P.; Gionis, A.; Mannila, H. A Shapley value approach for influence attribution. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 549–564.
20. Zhao, X.; Huang, W.; Huang, X.; Robu, V.; Flynn, D. Baylime: Bayesian local interpretable model-agnostic explanations. In *Uncertainty in Artificial Intelligence*; PMLR: Sydney, Australia, 2021; pp. 887–896.
21. Alvarez Melis, D.; Jaakkola, T. Towards robust interpretability with self-explaining neural networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*.
22. Sun, Z.; Fan, C.; Han, Q.; Sun, X.; Meng, Y.; Wu, F.; Li, J. Self-explaining structures improve nlp models. *arXiv* **2020**, arXiv:2012.01786.
23. Lei, T.; Barzilay, R.; Jaakkola, T. Rationalizing neural predictions. *arXiv* **2016**, arXiv:1606.04155.
24. Elton, D.C. Self-explaining AI as an alternative to interpretable AI. In *International Conference on Artificial General Intelligence*; Springer: Cham, Switzerland, 2020; pp. 95–106.
25. Camburu, O.M. Explaining Deep Neural Networks. Ph.D Thesis, University of Oxford, Oxford, UK, 2020
26. Zheng, H.; Fernandes, E.; Prakash, A. Analyzing the interpretability robustness of self-explaining models. *arXiv* **2019**, arXiv:1905.12429.
27. Sarker, I.H. Machine learning: Algorithms, real-world applications and research directions. *SN Comput. Sci.* **2021**, *2*, 1–21. [CrossRef]
28. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316. [CrossRef]
29. Du, X.; Xu, H.; Zhu, F. Understanding the effect of hyperparameter optimization on machine learning models for structure design problems. *Comput.-Aided Des.* **2021**, *135*, 103013. [CrossRef]
30. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hassabis, D. Mastering the game of Go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef] [PubMed]
31. Kanehira, A.; Harada, T. Learning to explain with complemental examples. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–19 June 2019.
32. Singh, A.; Sengupta, S.; Lakshminarayanan, V. Explainable deep learning models in medical image analysis. *J. Imaging* **2020**, *6*, 52. [CrossRef] [PubMed]
33. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Deep Learning*; PMLR: Sydney, Australia, 2017.
34. Erhan, D.; Bengio, Y.; Courville, A.; Vincent, P. Visualizing higher-layer features of a deep network. *Univ. Montr.* **2009**, *1341*, 1.
35. Kapishnikov, A.; Bolukbasi, T.; Viégas, F.; Terry, M. Xrai: Better attributions through regions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 4948–4957.
36. Xu, S.; Venugopalan, S.; Sundararajan, M. Attribution in scale and space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9680–9689.
37. Kapishnikov, A.; Venugopalan, S.; Avci, B.; Wedin, B.; Terry, M.; Bolukbasi, T. Guided integrated gradients: An adaptive path method for removing noise. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 5050–5058.
38. Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; Kim, B. Sanity Checks for Saliency Maps. *arXiv* **2018**, arXiv:1810.03292.
39. Gupta, A.K.; Seal, A.; Prasad, M.; Khanna, P. Salient object detection techniques in computer vision—A survey. *Entropy* **2020**, *22*, 1174. [CrossRef] [PubMed]
40. Hosny, A.; Parmar, C.; Quackenbush, J.; Schwartz, L.H.; Aerts, H.J. Artificial intelligence in radiology. *Nat. Rev. Cancer* **2018**, *18*, 500–510. [CrossRef] [PubMed]