Evaluating Grad-CAM heatmaps of X-ray images

Vaggelis Lamprou

On the current study we present a post-testing technique that evaluates the level a learned CNN-based classifier has understood a given dataset. We consider images from the "COVID-19 Radiology Database" and a transfer learning setup to train and test one VGG16 and one DenseNet210 based classifier. Afterwards, we evaluate test image heatmaps, generated by the Grad-CAM algorithm, through the area over the MoRF perturbation curve (AOPC) as an additional metric for our models. In the end, we reach the conclusion that the best generalizing model (with respect to recall and f1 scores) is coupled with better AOPC score as well; resulting in an additional indication for using it against the other model over this dataset.

GitHub Reposiroty : link

1 Introduction

Discovering patterns and structures in large collections of data in an automated manner is a core component of data science, and currently introduces applications in diverse areas such as medicine, biology, law and finance. However, such a highly positive impact comes with significant challenges: how do we understand the decisions suggested by these systems in order that we can trust them?

As already known the inner workings of a neural network model consist of many layers and nodes connected via non-linear relations and even if one attempts to inspect all these it is unfeasible to fully comprehend how the model makes its decisions. Thus, due to their high complexity, neural networks are often referred as "black boxes".

However, despite their promising performance in numerous cases, their structure raises various concerns because they may be biased in some way and such bias might go unnoticed. Especially in medical applications, this can have far-reaching or even fatal consequences, as they might fail to detect an existing disease in time while it is still treatable.

All above leads to the need of developing techniques that can shed light into the aspects leading models to predict as they do. In that way, the field experts can give their experienced opinion and help data scientists to correct flaws, improve the model's understanding of the data and enhance its credibility.

In this article we explore medical X-ray images of people who might suffer from Covid, Lung Opacity or Viral Pneumonia. In the first place we develop two feature pre-trained CNN classifiers to detect the disease (if any) and then use the Grad-CAM algorithm, as introduced in [1], to produce heatmaps that explain the classifiers' behaviour. In the end, we compute the area over the MoRF perturbation curves, as presented in [2], in an attempt to evaluate the heatmaps and conclude which model "understands" better the dataset apart from common metrics, such as recall and f1-score.

Before we move into details about the methods used in this study, we also refer the interested reader to relevant scientific publications in order to enrich their knowledge base. In the first place, in [3] the writers use the famous LIME and SHAP (as per [4]) explanation techniques on three Machine Learning classifiers that predict Brain Tumour Survival. On the other hand, in [5], a Mammographic Image Classification task is ad-

dressed via CNN structures and the Grad-CAM and GradCAM++ (as per [6]) algorithms are considered for the explainability part. Finally, in [7] one can find a summary of XAI methods in deep learning-based medical image analysis. A large variety of algorithms is categorized into one of the following categories: Visual, Textual or Example-based explanation algorithms; and then further sub-categorized into model-based, post-hoc, model-specific or model-agnostic and into local and global methods.

2 Methods

2.1 Data

For the purposes of this study we use the "COVID-19 Radiograhy Database" (Kaggle link). The current version consists of image X-ray data divided into four classes: Covid disease (3616 samples), Lung Opacity disease (6012 samples), Normal x-rays (healthy people) (10192 samples) and Viral Pneumonia disease (1345 samples).

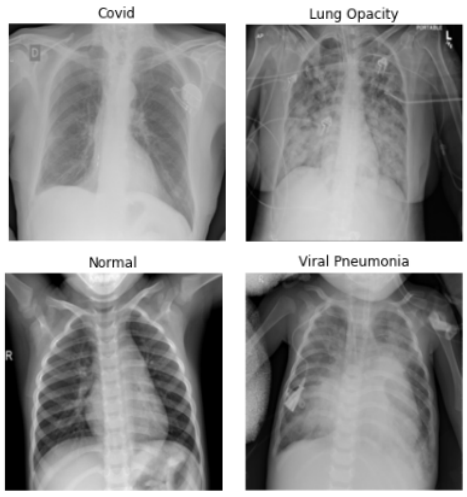


Figure 1: X-ray examples

However, due to the large size of the dataset and the limited (Colab) hardware resources we perform *down-sampling* of the larger classes to the size of the small one, resulting in a dataset of 5380 samples (1345 instances per class).

Afterwards, each class is split into data portions of 80%, 10% and 10% for the construction of the training, validation and test datasets respectively. This procedure gives class-balanced training dataset of 4304 samples, validation dataset of 536 samples and test dataset of 540 samples.

2.2 The classifier models

The CNN classifiers are two VGG16 (link) and DenseNet201 (link) feature-based models followed by a sequence of MaxPooling ($(4,4)$ pooling size), Flatten and Dense (64 nodes, ReLU) layers before the final Dense (4 nodes, Softmax) layer.

For both models the pre-trained layers were kept frozen during training, resulting in the following structures:

- VGG16-based model : Total params 14,846,084 - Trainable params 131,396
- DenseNet201-based model : Total params 18,813,828 - Trainable params 491,844

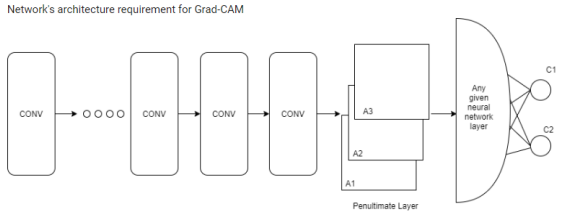
We also mention that both models were compiled with Categorical Crossentropy loss function and optimized with Adam. During training we monitored the validation recall per class and eventually kept the model instances which maximized the three diseases validation recall scores.

The following two subsections are more theory-based and contain the basic mathematical framework to understand the Grad-CAM and MoRF techniques which are the main points of this article.

2.3 The Grad-CAM algorithm

The Grad Cam algorithm is a post-hoc (i.e. applies to learned model) model-specific technique that applies to CNN-based models and produces class dependent heatmaps (visual explanations) with the image regions of high importance (local explanations). The heatmaps are generated by considering a weighted positive sum of the feature maps (activation maps) $A^1, .., A^K$ of the last convolutional layer, where the weights are class dependent averaged derivatives of the class output (before softmax) with respect to the respective feature map coordinates (pixels).

As we see below (in step 1) the only requirement to compute the heatmaps is to have differentiable layers after the convolutional part of the network.



We denote by A^k the k-th feature map of the last convolutional layer, where $k = 1, 2, ..K$. In order to get the heatmap for class c, the algorithm steps can be summarized as follows :

- Step 1: We first compute the gradient of the score for class c , y^c (before the softmax), with respect to feature map activations A^k_{ij}, where i, j run over all pixels locations.

$$\frac{\partial y^c}{\partial A^k_{ij}}$$

  The computation of the derivatives is achieved by back-propagation, indicating the need for differentiable layer structure after the maps A^k.

- Step 2: We use global average pooling over step 1 derivatives to define the quantity

$$a^c_k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}$$

  This is an overall "importance" score of A^k for predicting class c.

- Step 3: Perform a weighted linear combination of a^c_k and A^k and apply ReLU activation to get the class heatmap L^c

$$L^c_{Grad-CAM} = ReLU(\sum_k a^c_k A^k) \quad (1)$$

  We note that the choice of ReLU ensures that only pixels of positive influence for the class of interest are kept. In other words, pixels whose intensity should be increased in order to increase y^c.

2.4 MoRF perturbation curves

In this section we present the theoretical background for developing a heatmap evaluation process called "MoRF" (Most Relevant First), as introduced in [2].

The technique looks at the heatmaps as a decreasing sequence of importance regions

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

and performs an iterative procedure that measures how much the class encoded in the image (e.g. as measured by the learned model f) disappears when we progressively apply perturbations to the most relevant regions by the order given in $\mathcal{O}$.

From a mathematical point of view, for each test image x, it produces a *sequence of images* $x_{MoRF} = \{x^{(0)}_{MoRF}, x^{(1)}_{MoRF}, .., x^{(L)}_{MoRF}\}$, as

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

where g is a function that "removes" information from the image $x^{(k-1)}_{MoRF}$ at region r_k, *and* the *"MoRF graph"* as defined by the points

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

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

For a given test point x the area over the perturbation curve can be controlled (approximated) by the quantity

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

while for the entire dataset by

$$AOPC = \frac{1}{L+1} < \sum_{k=1}^{L} [f(x^{(0)}_{MoRF}) - f(x^{(k)}_{MoRF})] > \quad (2)$$

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

3 Experimental results

3.1 Model Training & Validation

The models of subsection 2.2 were trained for 20 epochs and monitored with the ModelCheckpoint callback function. Below we see the recall history per class and model architecture (Figures 2 and 3) and the final scores of the model instance we chose.

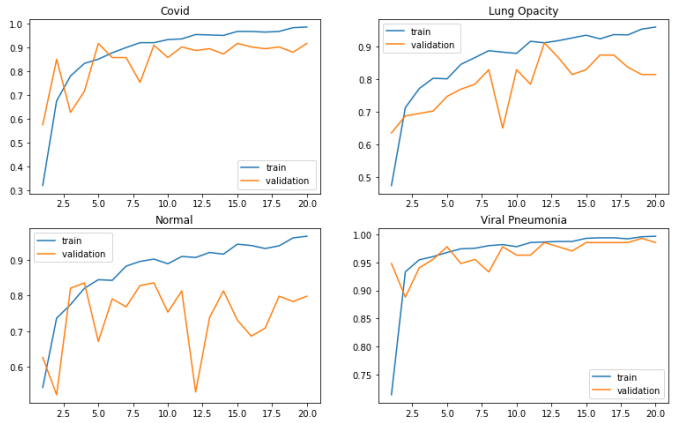


Figure 2: VGG16 based model - Recall history

As per Figure 2 graphs and keeping in mind that we want to maximize validation recall scores of the diseases, we eventually kept the instance of epoch 16 for the VGG16 based architecture. Its validation recall scores are : COVID class 0.9, Lung Opacity class 0.87, Normal class 0.69, Viral Pneumonia class 0.99 and Overall 0.86.

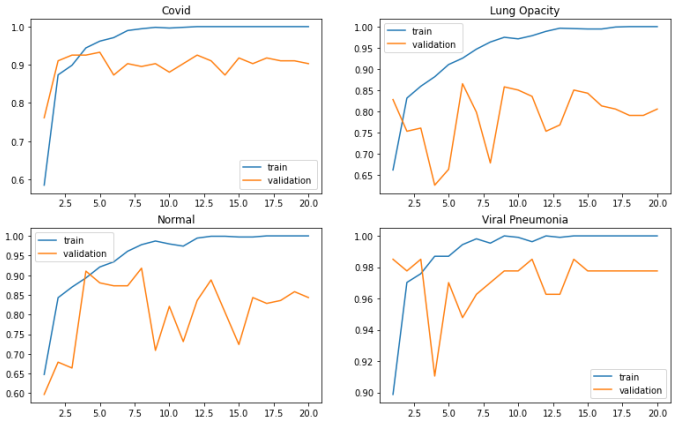


Figure 3: DenseNet201 based model - Recall history

On the other hand, regarding the DenseNet201 based model of Figure 3, we kept the instance of epoch 15. Its validation recall scores are : COVID class 0.92, Lung Opacity class 0.84, Normal class 0.72, Viral Pneumonia class 0.98 and Overall 0.86.

We remark here that the Viral Pneumonia seems to be the most easily identified class by both models. In addition, the normal class seems to have the least correctly predicted points but we may put less emphasis on this at the moment as it is more crucial to identify unhealthy cases.

3.2 Model Testing

In this section we briefly present the classification report results on the test set. Again we focus more on the recall and the f1 scores.

	precision	recall	f1-score	support
Covid	0.88	0.84	0.86	135
Lung Opacity	0.82	0.82	0.82	135
Normal	0.79	0.82	0.80	135
Viral Pneum	0.96	0.96	0.96	135
accuracy	-	-	0.86	540
macro avg	0.86	0.86	0.86	540
weighted avg	0.86	0.86	0.86	540

Figure 4: VGG16 based model - Classification report

	precision	recall	f1-score	support
Covid	0.91	0.87	0.89	135
Lung Opacity	0.79	0.81	0.80	135
Normal	0.83	0.86	0.85	135
Viral Pneum	0.99	0.98	0.98	135
accuracy	-	-	0.88	540
macro avg	0.88	0.88	0.88	540
weighted avg	0.88	0.88	0.88	540

Figure 5: DenseNet201 based model - Classification report

The reports of Figures 4 and 5 highlight that with respect to the overall recall and f1 scores, the VGG16 based model achieves 86% while the DenseNet201 outperforms it with 88%.

At the same time, as expected, this aligns well with the class scores, where the DenseNet201

based model achieves better scores in all categories except for the lung opacity disease. However, even in this case the VGG16 model manages better class scores only by 2% and 1% in the recall and f1 metrics respectively.

Metric-wise speaking, the focus now turns into whether this slight superiority of the DenseNet201 model is also preserved in the AOPC scores. This is discussed in subsection 3.4.

3.3 Grad-CAM Heatmaps

Recalling the theory of subsection 2.3 and the resulting equation (1), in this section, we explore the produced Grad-CAM heatmaps via a test image example which is correctly classified by both models as "Covid".

Note that the produced heatmaps are the ones of the top predicted class (i.e. Covid).

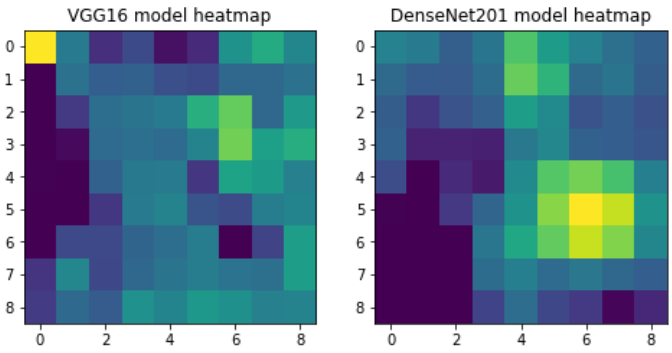


Figure 6: Grad-CAM heatmaps

We observe that the Grad-CAM divides the image into a 9*9 square block regions where the lighter the colour the most important the region is for the class.

At this point, we might also encounter a misconception of the VGG16 based model as it focuses very much on the top left region of the X-ray where it is unlikely to detect the disease. On the other hand, the DenseNet201 based model correctly focuses on the lungs area where the disease could be located. It behaves more reasonably.

Finally, in order to get a better understanding of the relation between the original X-ray and the heatmap, in Figure 7 we see multiple superimposed images at different α-levels.

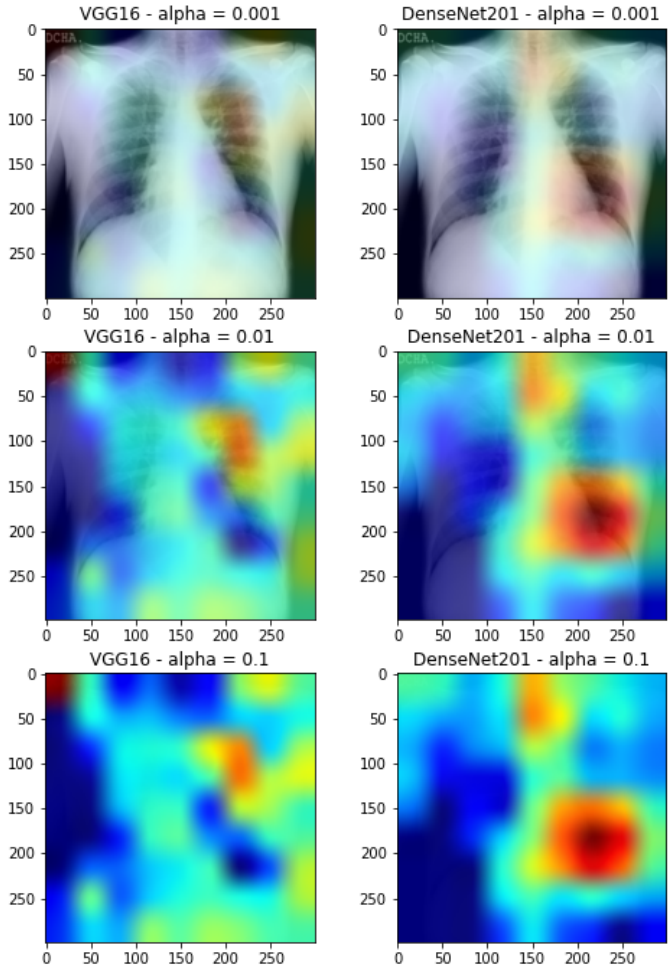


Figure 7: Superimposed images

3.4 Area Over Perturbation Curves (AOPC)

In this final section we evaluate the quality of Grad-CAM heatmaps via the AOPC score given in equation (2). As already explained the higher the value the better the corresponding model heatmaps.

In our computations we consider only the correctly predicted test images per model and progressively perform perturbations in the original X-ray regions ($(34, 34)$ pixel regions) by adding normal noise points of 0 mean and 0.1 standard deviation. The findings are presented in the next graph, which displays the AOPC values as functions of the perturbation steps.

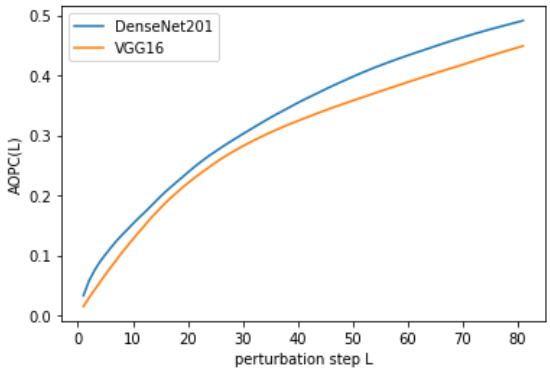


Figure 8: AOPC values as function of perturbation steps

We observe that the DenseNet201 based model values are higher during all perturbation steps. In the end, the DenseNet201 based model achieves 0.49 score while the VGG16 based model 0.45.

We can look at this graph as another way of evaluating what the model has actually learned from the training data. It suggests that the DenseNet201 model can focus batter on the important regions of the X-ray and combining it with the classification report recall and f1 scores we get an additional indication that the DenseNet201 based model could be preferred over the VGG16 based model when it comes to X-ray predictions for this kind of dataset.

4 Conclusion

4.1 Summary

In this study we saw an application of the post-hoc explainability algorithm named Grad-CAM on images of the "COVID-19 Radiology Database" via CNN-based classifiers. Grad-CAM is a model dependent technique that generates class-dependent heatmaps over (test) images and highlights the image regions where the model "sees" the class inside the image.

Having build more than one classifier the challenge of comparing their heatmaps quality came up and led us to consider the AOPC score. As explained in subsection 2.4 this metric emerges from the area above the image perturbation curve which is constructed by the MoRF iterative procedure.

Eventually, for our models we saw that the test and AOPC scores align well, in the sense that the best test performing model has a better AOPC score as well.

More specifically, the final scores are :

Base model	Test Recall	Test F1	AOPC
VGG16	86%	86%	0.45
DenseNet201	88%	88%	0.49

4.2 Future improvements

At this point, it is essential to highlight that, despite the final presented results, there are many points in this study that allow further investigation and the present could be considered only a starting point for more experiments that could lead to more concrete results. Unfortunately, in the interest of time, experiment aspects such as tuning the hyperparameters of the classifier part put on top of the pretrained layers or developing more hardware demanding sampling techniques, other than the simple down-sampling method used here, in order to deal with the original class imbalance problem, were not dealt as desired at first. Furthermore, we note that one might try to make use of the image masks that come with the dataset and which were ignored for this study. Since the mask marks the chest part of the image and has black colour for the non-information part, an approach would be to use the masks as an input to a segmentation model which would then output a segmentation map of the image. This segmentation map could then be used as an input to the image classification model hopefully helping the model to focus on the relevant parts of the image when making predictions. Finally, one might consider many different variations of the noise function g in the MoRF procedure. It is worth exploring how this aspect could affect the final AOPC values and if the DenseNet201 based model can still outperform the VGG16 after more intensive noise perturbations.

5 References

1. *"Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Local-*

ization" by Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra

2. *"Evaluating the visualization of what a Deep Neural Network has learned"* by Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Bach, Klaus-Robert Müller

3. *"Interpretable Machine Learning Classifiers for Brain Tumour Survival Prediction"* by Colleen E. Charlton, Michael Tin Chung Poonb, Paul M. Brennanb, Jacques D. Fleuriot

4. *"A Unified Approach to Interpreting Model Predictions "* by Scott M. Lundberg and Su-In Lee

5. *"Interpretable Mammographic Image Classification using Case-Based Reasoning and Deep Learning"* by Alina Jade Barnett, Fides Regina Schwartz , Chaofan Tao, Chaofan Chen, YinhaoRen , Joseph Y. Lo and Cynthia Rudin

6. *"Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks"* by Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader and Vineeth N Balasubramanian

7. *"Explainable artificial intelligence (XAI) in deep learning-based medical image analysis"* by Bas H.M. van der Velden, Hugo J. Kuijf, Kenneth G.A. Gilhuijs, Max A. Viergever