Pneumonia Detection with Explainable Predictions

Ajeet Yadav IIIT Delhi
Yash Amin IIIT Hyderabad
Arth Raj IIIT Hyderabad
Piyush Bhopalka MarkovML

March 10, 2023

Abstract

Neural networks are known for their good predictive performance, but their lack of interpretability makes them difficult to use in certain applications. In this paper, we present an implementation of Layer-wise Relevance Propagation (LRP), a technique for explaining the predictions of a neural network. We apply LRP to a neural network that detects pneumonia from chest X-ray images and demonstrate its effectiveness in providing transparent explanations for the network's decisions.

The LRP technique decomposes the network's decision into its constituent parts and attribute relevance scores to each input feature. These scores can then be used to explain why the network arrived at a particular decision. By analyzing the relevance scores assigned to each input feature, we can gain insights into the model's decision-making process and identify regions of the X-ray images that contribute most to the prediction. Our results show that the LRP method can be a powerful tool for understanding the decision-making process of neural networks and has the potential to improve their interpretability and trustworthiness in a wide range of applications.

Our work contributes to the growing body of research on explainable artificial intelligence (XAI), which aims to make machine learning models more transparent and trustworthy. The LRP techique can be applied to other neural network models for different tasks, allowing for greater insight into the decision-making process of these models. This has implications for a wide range of applications, including medicine, finance, and autonomous driving, where interpretability and transparency are essential.

Keywords: Interpretability, Deep Neural Networks, Layer-wise Relevance propagation

Chapter 1

Introduction

Artificial intelligence has greatly benefited from the usage of neural networks, which are extensively employed in a wide range of tasks, including speech recognition, image recognition, and natural language processing. Despite their adaptability, Deep Neural Networks (DNNs) are frequently criticised for their interpretability issues, which make it difficult to comprehend how they make their predictions. Instead of offering explanations for their decisions, such as relevance distributions over the input features, traditional DNNs are primarily producee probability distributions over classes. There is a lot of interest in creating Explainable AI (XAI) techniques since the lack of transparency raises questions about the accountability and trustworthiness of the decisions made by these models.

In this work we have implemented Layer-wise Relevance Propagation Technique (LRP) to explain the predictions of a neural network which acceptsa an input chest-xray image and determines whether the person have any type of Pnuemonia or not. LRP works by propagating the relevance of the output prediction back through the network, assigning a relevance score to each neuron in the network based on its contribution to the final prediction.

Here's a high-level overview of how LRP works:

1. Select an output neuron: LRP starts by selecting the output neuron of the network whose prediction we want to explain. The relevance of this neuron's prediction will be propagated back through the network.

2. Compute the relevance score: The relevance score of the output neuron is calculated based on the prediction made by the neuron and the desired output. This relevance score represents the importance of the output neuron's prediction in terms of the overall goal of the network.

3. Propagate the relevance score: The relevance score is then propagated back through the network, starting from the output neuron and working backwards towards the input layer. As it propagates, the relevance score is distributed to the neurons in the previous layer based on their contribution to the prediction made by the output neuron.

4. Assign a relevance score to each neuron: Once the relevance score has been propagated back through the network, each neuron in the network will have a relevance score assigned to it. This relevance score represents the importance of that neuron's output in terms of the overall prediction made by the network.

Overall, LRP is a useful tool for interpreting and explaining the predictions made by deep neural networks and can help to shed light on the internal workings of these complex models.

Chapter 2

Theory

Layer-wise Relevance Propagation (LRP) is a technique for explaining the predictions made by a neural network. LRP was introduced in the paper [1] since then, it has been applied to a wide range of NLP tasks, including text classification, sentiment analysis, and question answering. LRP has been shown to be effective at providing insights into the decision-making process of neural networks and has been used to improve the performance and interpretability of NLP models. Lapuschkin et al. [3] have used the LRP technique to investigate networks predicting gender and age from image data. Thomas et al. [6] applied the technique to a large corpus of fMRI neuroimaging data, explaining brain states from 3D data. Srinivasan et al. [5] use LRP to find out exactly what parts of a video are used by a classifier for human action recognition. Montavon et al.

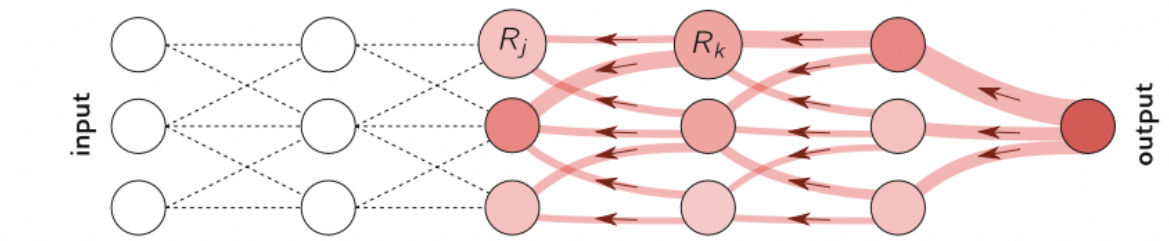

Figure 2.1: Illustration of the LRP procedure. Each neuron redistributes to the lower layer as much as it has received from the higher layer, (G. Montavon et al.)

Above figure is an illustration how LRP works, LRP works by decomposing the prediction made by a neural network into contributions from each input feature. It does this by propagating output score backwards through the layers of the network, starting from the output layer and working backwards to the input layer. The relevance scores reflect the contribution of each input feature to the overall prediction made by the network.

Considering the above figure, Let j and k be neurons at two consecutive layers of the neural network. Propagating relevance scores Rk at a given layer onto neurons of the lower layer is achieved by applying the rule:

$$R_j = \sum_k \frac{a_j w_{j,k}}{\sum_{j=0}^{j} a_j w_{j,k}} R_k \tag{2.1}$$

.

In the figure above, we have a simple binary classification neural network that accepts an input image and determines whether the image is of a cat or not. It outputs the probability of the image being a cat or not a cat. Now, let's say we're interested in knowing why the neural

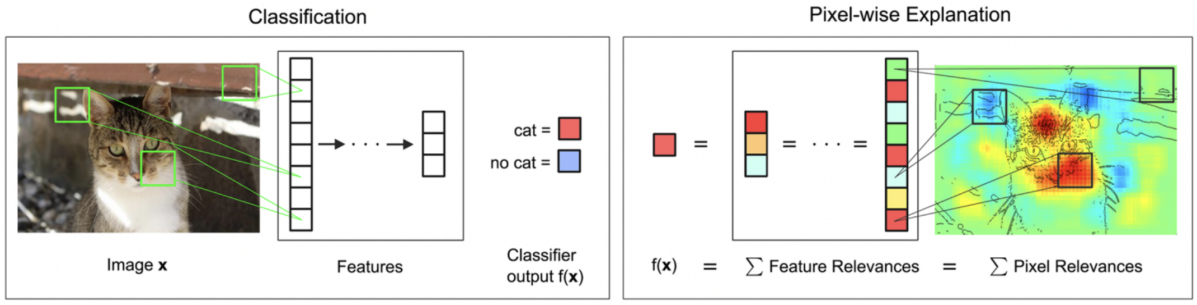

Figure 2.2: Visualization of LRP explanations and the pixel-wise decomposing process, (G. Montavon et al.)

network has predicted the image as a cat image. This is where LRP comes into the picture. LRP takes the cat output probability value and redistributes it backward in the network layer-by-layer. Eventually, we will reach the input layer where each node in the input layer will have its relevance score. This relevance score of each node represents the extent of its contribution to the final cat prediction. After re-constructing the image using the each pixels relevance score, we can see the regions in the image that are more highlighted, indicating that these regions are responsible for the cat prediction.

One of the key advantages of LRP is that it allows researchers to understand how the network is making its predictions, which can be useful for improving the performance of the network and for understanding the limitations of the model. It can also be used to identify potential biases or errors in the model, as well as to understand the generalization properties of the network.

There have been several papers published on LRP, including an overview paper that provides a comprehensive summary of the method and its applications. Some of the key findings from these papers include:

LRP has been shown to be effective for interpreting the predictions made by various types of neural networks, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer-based models.

LRP has been used to identify the specific input features that are most influential in the predictions made by the network, which can be useful for identifying potential errors or biases in the model.

LRP has been used to understand the generalization properties of neural networks, including how well the network is able to generalize to new data.

Overall, LRP is a useful technique for interpreting and understanding the predictions made by neural networks, and has been applied to a wide range of tasks and applications.

Chapter 3

Model Building

3.1 Dataset Details

We have used *Chest X-Ray Images (Pneumonia)* dataset from Kaggle, which contains a total of 5,856 chest X-ray images, with 4,556 images showing evidence of pneumonia and 1,300 images serving as negative controls. The dataset is split into three subsets: a training set with 4,387 images, a validation set with 16% of the images, and a test set with 20% of the images. There are three classes of images in the dataset namely 'Normal', 'Viral Pneumonia', 'Bacterial Pneumonia'.

The images were collected from two different sources: the first source consists of pediatric patients from Guangzhou Women and Children's Medical Center in Guangzhou, China, and the second source consists of adult patients from the National Institutes of Health Clinical Center in Bethesda, Maryland, USA.

The images are provided in PNG format and have a variable image size, with a minimum size of 262 x 205 pixels and a maximum size of 3,328 x 2,432 pixels. The dataset is publicly available on Kaggle and can be used for various machine learning applications, including image classification and pneumonia detection.

In the preprocessing step, we resized each image to equal size of 225*225 and did some data-augmentations, 'RandomHorizontalFlip', 'RandomRotation', 'RandomAffine', 'ColorJitter' to increase the no. of samples.

3.2 Pneumonia Detection Model

For building the Pneumonis Detection model which can accept an input as a person's chest-Xray image and determine whether the person have any type of Pneumonia or not, we have finetuned VGG-16 pretrained architecture.

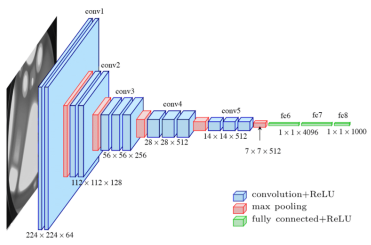

Figure 3.1: The standard VGG-16 network architecture, (Max Ferguson et. al)

VGG16 is a deep convolutional neural network architecture developed by the Visual Geometry Group (VGG) at the University of Oxford. It consists of 16 convolutional and fully connected layers and was introduced as part of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014.

The VGG16 architecture has several unique characteristics, including:

1. All convolutional layers have a 3x3 filter size and a stride of 1.
2. Max pooling layers are used after every two or three convolutional layers.
3. The number of filters in each convolutional layer is doubled after every max pooling layer.
4. The number of filters in each convolutional layer is doubled after every max pooling layer.
5. The network ends with three fully connected layers, followed by a softmax activation function for classification.

Due to its simplicity and effectiveness, VGG16 has become a popular choice for various computer vision tasks, including image classification, object detection, and segmentation.

Chapter 4

MarkovML

MLOps (Machine Learning Operations) is a set of practices and tools that aims to streamline the process of building, deploying, and managing machine learning models in production environments. MLOps is a combination of DevOps (Development Operations) and data science best practices, which helps organizations to automate the machine learning lifecycle, including data preparation, model training, model evaluation, deployment, and monitoring.
Following are the features that the MarkovML provide to automate the tasks involved in MLOPs.

1. **Dataset Analysis:** When we register a dataset on MarkovML it analyze our data and help us to understand key characteristics such as distributions, column correlations, empty value frequency, and more.

2. **Experiments:** MarkovML Experiments feature let us record data about our model training sessions, and get insights about how a customizable set of metrics changes with respect to model training time.

3. **Evaluations:** Once we have trained our model, Evaluations let us record additional model runs on new data and objectively evaluate model performance. MarkovML provides valuable insights into our models' strength and weakness, and lets us easily compare the performance of multiple models so we can select the best model for our use case with confidence.

4. **Models:** Models in MarkovML represent the model binaries created during our ML workflow, which are used to predict an outcome (the target value) based on some input data (the features).

5. **Projects:** Projects let us organize our team's model-related resources, i.e. Models, Experiments, and Model Evaluations.

We have used some of the above features of MarkovML which helped us to automate the various tasks involved in MLOPs and helped us to get better insights from our project.

4.1 Experiment Recording

Experiment recording in machine learning training refers to the practice of keeping track of the details and results of machine learning experiments. This can involve recording information such as the dataset used, the hyperparameters tuned, the model architecture, the training duration, training and validation accuracies, losses, etc.
Experiment recording can also help to identify potential issues, such as overfitting or underfitting,

and provide insights into how to improve the model's performance. Additionally, experiment recording can be useful for sharing findings with others and for building upon previous work, as it provides a clear record of the development process and results. We have recorded several experiments on MarkovML by changing various hyper-parameters in order to improve our model's learning.
Following figures shows accuracy & loss track of the experiment while training the Pneumonia detection model.

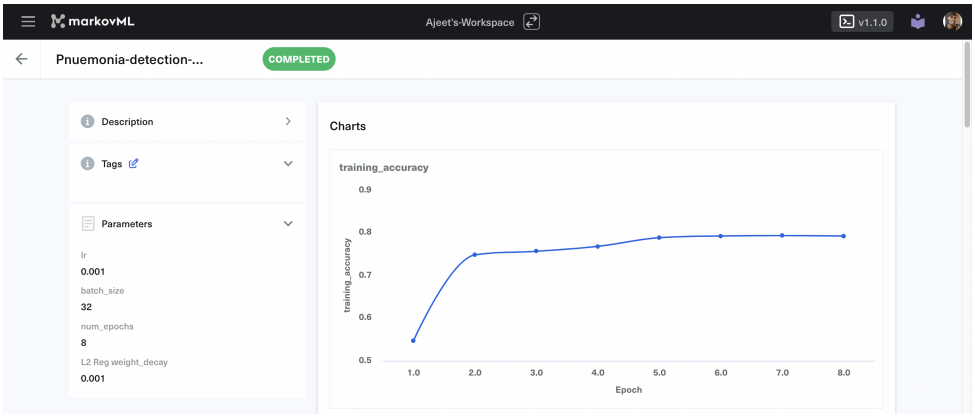

Figure 4.1: Accuracy track of the Pneumonia detection model at each epoch

Figure 4.2: Loss track of the Pneumonia detection model at each epoch

Chapter 5

Evaluations

After the experiment recording phase in MLOps, the next step is typically model evaluation. Model evaluation involves assessing the performance of the trained machine learning model on a set of data that was not used during the training process. This allows us to determine how well the model is likely to perform when deployed in the real world.

We have used MarkovML's evaluations feature to evaluate the performance of our model on unseen data. There are several metrics that MarkovML-Evaluations feature uses to evaluate the performance of machine learning models, including Accuracy, Precision, Recall, F1 score, model disagreement.

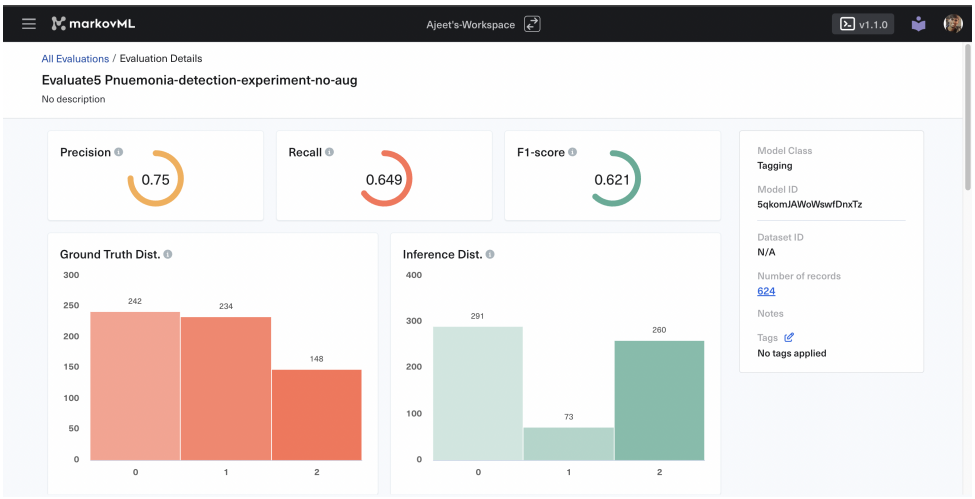

Figure 5.1: Evaluation metrics of Pneumonia Detection model on unseen data

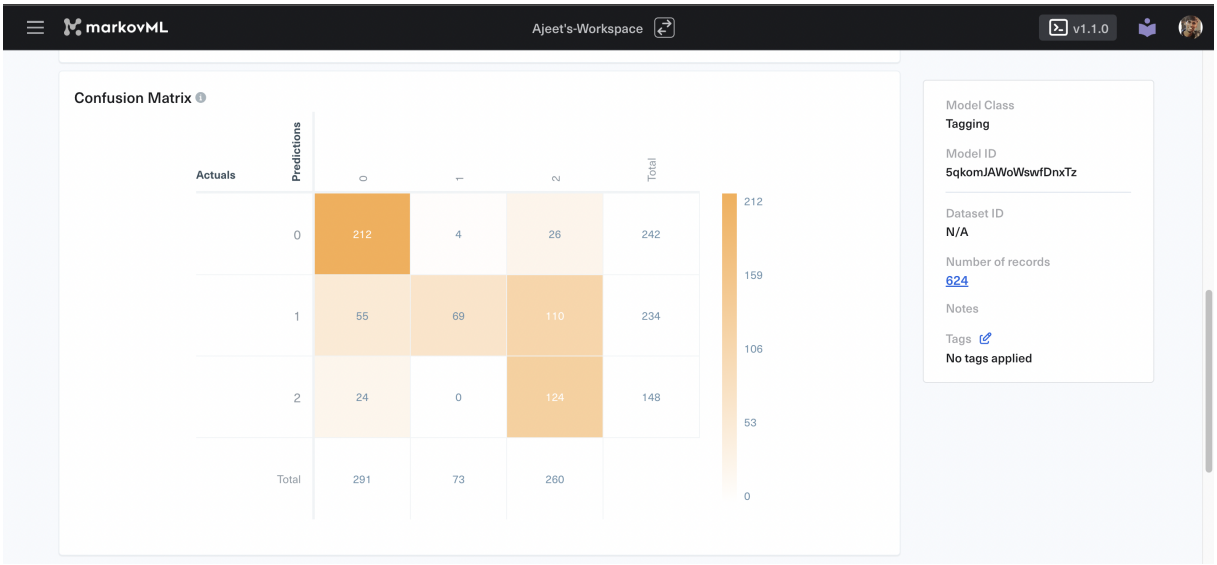

Figure 5.2: Confusion Matrix of Pneumonia Detection model on unseen data

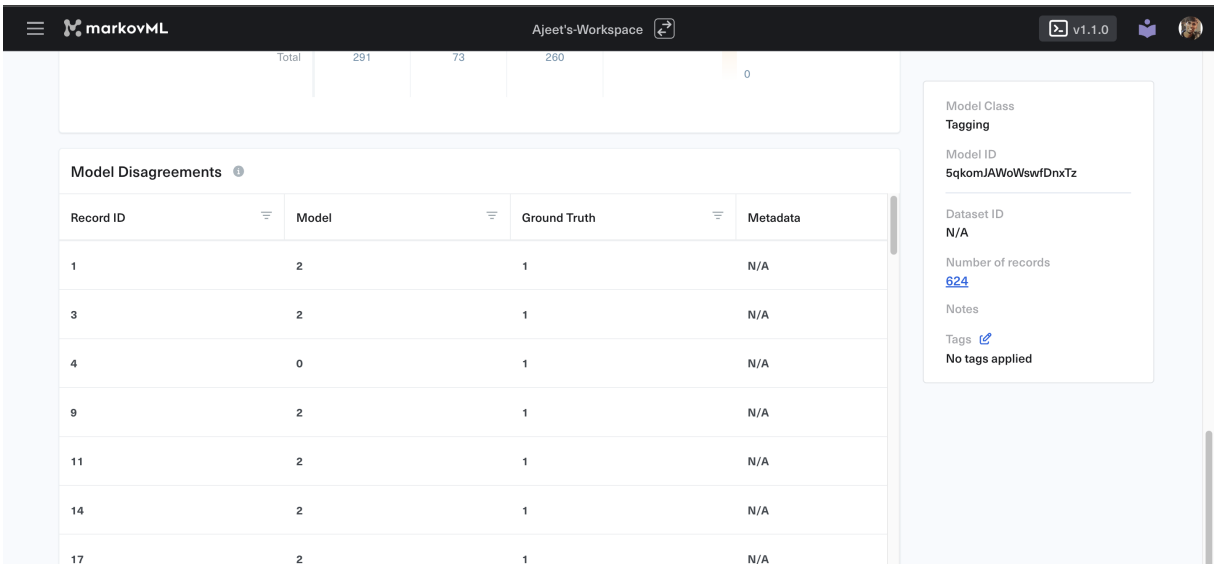

Figure 5.3: Model Disagreement on unseen data

Chapter 6

LRP Implementation

So far we have built a Pneumonia Detection model which can take an chest-xray input image of a person and determine whether the person have any type of Pneumonia or not. Now next step, we want to explain our Pneumonia detection model's predictions and for that we have implemented the LRP technique that we have discussed earlier.

The structure of LRP rules presented in, allows for an easy and efficient implementation and it can be decomposed in four steps.

$$\begin{aligned}
\forall_k : \; z_k &= \epsilon + \textstyle\sum_{0,j} a_j \cdot \rho(w_{jk}) && \text{(forward pass)} \\
\forall_k : \; s_k &= R_k / z_k && \text{(element-wise division)} \\
\forall_j : \; c_j &= \textstyle\sum_k \rho(w_{jk}) \cdot s_k && \text{(backward pass)} \\
\forall_j : \; R_j &= a_j \, c_j && \text{(element-wise product)}
\end{aligned}$$

Figure 6.1: LRP implementation steps, (G. Montavon et al.)

The first step determines the sum of influences for each neuron in the higher layer and is analogous to a modified forward pass.
The second and fourth steps are simple element-wise operations.
And in the third step, this c_j can be seen as how much relevance trickles down to neuron j from the succeeding layer.
Intuitively, a neuron is relevant if
1) it has a high activation, and
2) it contributes a lot to relevant neurons of the higher layer.

Chapter 7

LRP Results

To visualize the LRP explanation on the input image we have used heatmap using the resulting relevance scores to assign a color to each pixel in the input image.

The heatmap is visualized on top of the input image, with higher relevance scores corresponding to brighter colors. This allows the user to identify the regions of the image that were most important for the model's decision, and can help provide insights into how the model is making its predictions. This can be useful for comparing the relative importance of different regions of the image, and for identifying specific features that are important for the model's decision.

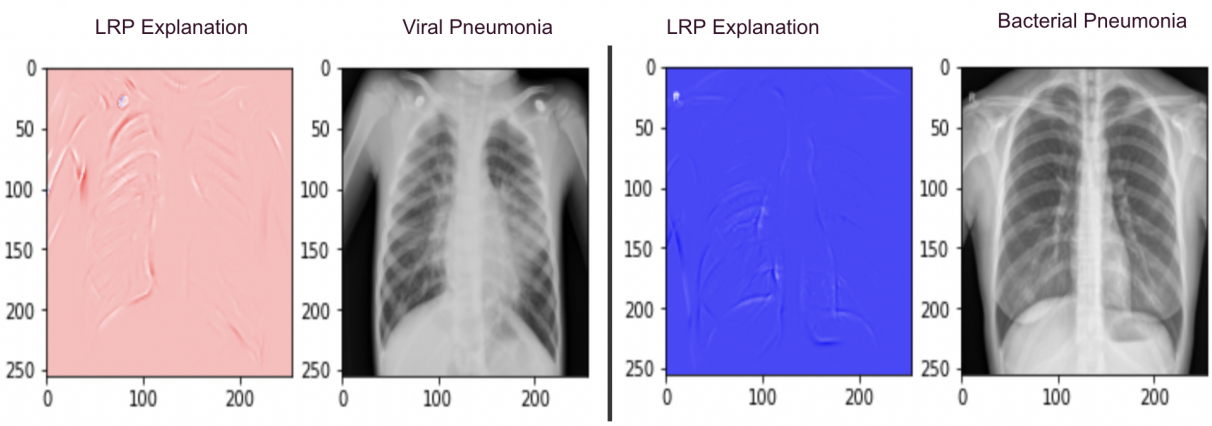

Figure 7.1: Visualization of LRP explanation on the input image.

Here we can see we have two input images and our Pneumonia Detection model predicted one as 'Viral Pneumonia' and other as 'Bacterial Pneumonia' and our LRP implementation has highlighted some regions in the input images saying these are the regions based on which the model has made its predictions.

One thing we realised after the LRP implementation that we do not have any metrics to evaluate how good these explanations are, this can only be evaluated by domain-experts particularly Radiologists.

Chapter 8

Conclusion & Future scope

We were successfully able to train our Pneumonia detection model and also LRP implementation for the model's prediction. But in the experiment recording step, we have realised that our pneumonia detection model was overfitting the training data, which resulted in lower performance on the validation set and potentially affected the reliability of the LRP explanations. It's important to address overfitting in order to build a more reliable and robust model, we tried several methods like using dropout layers, and regularization techniques which improved our model's performance but still the model there was overfitting issue. In terms of conclusions, it's important to acknowledge that the model's performance on the validation set is a key metric for evaluating its effectiveness, and overfitting can be a major issue that affects the model's performance and generalizability. The implementation of LRP is also influenced by the model's performance, and it's important to ensure that the model is performing well on both the training and validation sets in order to have more confidence in the LRP explanations.

Moving forward, there are several potential future directions for the project. One option is to explore more other ways to address the overfitting issue, such as using techniques like early stopping during training and data-augmentation to increase no. of training samples.
Overall, LRP is a useful tool for interpreting and explaining the predictions made by deep neural networks and can help to shed light on the internal workings of these complex models and can be applied to a wide range of tasks and applications.

Bibliography

[1] Montavon, G., Binder, A., Lapuschkin, S., Samek, W., Müller, K. R. (2019). Layer-wise relevance propagation: An overview. In W. Samek, G. Montavon, A. Vedaldi, L. Hansen, K. R. Müller (Eds.), Explainable AI: Interpreting, explaining and visualizing deep learning (Vol. 11700, pp. 142-158). Springer, Cham. https://doi.org/10.1007/978-3-030-28954-6_10

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

[3] Lapuschkin, S., Binder, A., Müller, K.-R., Samek, W. (2019). Understanding and comparing deep neural networks for age and gender classification. In Proceedings of the IEEE International Conference on Computer Vision (pp. 454-463).

[4] Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Müller, K.-R. (2020). Unmasking Clever Hans predictors and assessing what machines really learn. Nature Communications, 11(1), 1-15.

[5] Srinivasan, V., Lapuschkin, S., Hellge, C., Müller, K.-R., Samek, W. (2019). Interpretable human action recognition in compressed domain. arXiv preprint arXiv:1904.07826.

[6] Thomas, A. W., Heekeren, H. R., Müller, K.-R., Samek, W. (2019). Analyzing neuroimaging data through recurrent deep learning models. Frontiers in Neuroscience, 13, 133.