Machine Learning for Healthcare Report
Project 2: Interpretability and Explainability

Afonso Ferreira da Silva Domingues
24-936-114
aferreira@student.ethz.ch

Thösam Norlha-Tsang
20-828-075
tnorlha@student.ethz.ch

Rahul Kaundal
19-931-468
rkaundal@student.ethz.ch

Abstract

This report investigates interpretable and explainable machine learning methods for medical classification tasks using both tabular and imaging data. In Part 1, we focus on the heart disease prediction task, applying shallow models like logistic regression with L1 regularization and deep models such as MLPs and NAMs, using SHAP for post-hoc explanation. In Part 2, we analyze chest X-ray images for pneumonia detection with CNNs, employing saliency-based methods including Integrated Gradients and Grad-CAM to highlight relevant image regions. Our findings offer insights into selecting suitable, trustworthy, and interpretable models for practical medical applications.

Part 1: Heart Disease Prediction Dataset

By exploring the dataset, we first notice that some columns consist of numerical values (e.g., the *RestingBP* column) and some columns consist of categorical values (e.g. the *ChestPainType*). Some of the categorical values are strings, which the machine learning model cannot work with. That means, later, we have to transform the categorical values such that we are able to feed them into a model. The sample size of training data is 734 and that of test data is 184 each with 12 columns. Checking out the training data further, we notice that it doesn't contain any missing values.

Looking at the distribution of each column where the data is separated by male and female patients, we see that the vast majority of the patients are male. From the distribution of *Cholesterol* we can clearly see that there are a lot of values that are zero. These values are outliers because cholesterol levels can never be zero, so we can later remove them safely. By also looking at the data using boxplots, we can see that there are some data points that lie outside of the whiskers of the boxplots.

The training data set is balanced according to Google[1]. If the minority class has 20-40% of the dataset then the imbalance is mild. If it has 1-20% then the imbalance is moderate. If the minority class has <1% of the dataset, then it is extremely imbalanced. In our case, the ratio of heart disease patients to no heart disease patients (*HeartDisease* is 0.54 and 0.45, respectively). So we can conclude that our data set is balanced, and we don't have to balance it. (**Q1.1**)

Next, we remove the outliers from our data. There are many ways to remove them, but here we consider two methods, namely, the interquartile range (IQR) method and removing outliers with the Z-score method, because the numerical columns follow a normal distribution. The next step is to impute missing values, but there are none, so we can skip this step. Finally, an important step is to encode the categorical string values in order to feed them into the model. Here, we use one-hot encoding. For the last step, we scale the training and test dataset with *MinMaxScaler*. (**Q1.2 & Q1.3**)

We then fit a logistic regression model with an L1 norm on the preprocessed training data set. (**Q2.1**). The most crucial preprocessing steps are encoding categorical features and standardization of the input features. As mentioned before, we have to encode the categorical data to be able to feed it into the model, which we do with one-hot encoding. Additionally, standardization of the input feature is crucial as well. Lasso applies a penalty on the absolute value of of feature coefficients. If the features are not in the same scale, it will unfairly penalize features with larger scales more heavily. (**Q2.2**)

Evaluating the logistic lasso regression model on test data, we can see from Table 1 that it performs quite well with an F1-score of 0.8811 and with a balanced accuracy of 0.8397. (**Q2.3**)

The largest contributing features are *Cholesterol, ChestPainType_ATA, Oldpeak and ChestPainType_NAP*, because they have the largest coefficients. In contrast, there is almost no contribution from the features *RestingECG_ST, RestingECG_Normal and RestingBP*. Features with positive coefficients are associated with heart disease. For instance, *Oldpeak* has the largest positive value, and hence it is associated with a higher likelihood of heart disease (see Figure 3). (**Q2.4**)

Good idea: Feature selection with Logistic Lasso Regression can simplify the model by removing non-contributing features, which can make the prediction values of the final model easier to explain.

[1] https://developers.google.com/machine-learning/crash-course/overfitting/imbalanced-datasets

Bad idea: There might be features that the researcher discards due to their low contribution, but due to uncertainty of feature importance, the features might still have an impact on the outcome. (**Q2.5**)

	Logistic Lasso Regression	MLP	NAM
F1-Score	0.8811	0.8750	0.8621
Balanced Accuracy	0.8397	0.8373	0.8059

Table 1: Performance metrics for Logistic Lasso Regression, MLP and NAM models

Additionally, we train a simple MLP model with two fully connected hidden layers, where we used the sigmoid function for output. We trained and tested the model with the same data set as the previously mentioned regression model. The performance of the MLP was comparable to the one of the regression model, with an F1-score of 0.8750 and with a balanced accuracy of 0.8373 (see Table 1). (**Q3.1**)

Taking a look at the plots (**Q3.2**) for the negative sample, we see that the plots have a similar shape, but the order of the features in the plot is not the same. Also, for the two positive samples, we also have a similar shape, except for the *Age* feature. However, also in this case we have a different feature ordering. Furthermore, taking a look at the overall feature importance, we see that it also has a different order. So none of the plots are consistent with each other (see Figure 5). (**Q3.3**)

Thirdly, we trained and tested a NAM model on the same training and test data as the previous two models. The NAM model achieves a similar performance to the logistic lasso regression and MLP models, with an F1-score of 0.8621, and a balanced accuracy of 0.8059 (see Table 1). (**Q4.1 & Q4.2**)

If we take a look at the feature importance of the NAM model, we can see that the top three important features are *ST_Slop_Up, ChestPainType_ATA, ExerciseAngina_Y* (see Figure 4). (**Q4.2**)

Comparing the three different approaches, we can conclude that logistic regression is a model where it is simple to extract the importance of features, but it is less powerful compared to neural networks (such as MLPs). On the other hand, neural networks are very flexible, but since you often have fully connected layers, it makes it hard to extract information about the feature importance. NAMs combine the interpretability of logistic regression and the flexibility of neural networks. This is done by having for each feature a small neural network. (**Q4.3**)

We thus find that NAMs are more interpretable than standard MLPs because each input feature is handled by its own small neural network, and the output is the sum of these independent functions. In contrast, MLPs mix all features together in complex ways, making it difficult to isolate the influence of individual inputs. Thus, NAMs preserve interpretability while still modeling non-linear relationships through feature-specific subnetworks. (**Q4.4**)

Part 2: Pneumonia Prediction Dataset

By exploring the data, we can see that the dataset contains 5863 images in total, divided into training (5216 samples), validation (16 samples), and test (624 samples) sets. The images are labeled either as "normal" or "pneumonia", and in the case of pneumonia, the title of the image indicates if it was caused by bacterial or viral infection.

Plotting some images, we observe considerable visual differences between healthy and disease samples. The normal chest X-rays show healthy, clear lungs with no unusual cloudy areas. Bacterial pneumonia cases present local white patches in specific parts of the lung, whereas viral pneumonia looks more spread out, with cloudy patterns across both lungs **(Q1.2)**. However, these cloudy regions are difficult to see in some images, as in some samples the overall image is too bright or lacks contrast. For this reason, we preprocess the images by applying adaptive histogram equalization (CLAHE), which improves the contrast and makes these regions more distinguishable. In addition to low contrast, the images also vary in size, and therefore we resize them to 300×300 pixels for consistency **(Q1.4)**. Quality-wise, the images are adequate, although some samples exhibit slight noise, and minor markings, typically on the left side **(Q1.1)**.

In regard to the labels of the images, these are distributed in the following manner:

As we can see from Table 2, pneumonia cases outnumber normal cases in the training set by nearly a factor of three, which can introduce biases into our model **(Q1.3)**. To address this imbalance, we perform data augmentation by generating 2534 additional images from the normal samples, ensuring a more uniform distribution of labels **(Q1.4)**. To this end, we apply a series of stochastic image transformations such as rotation (between -15 and 15 degrees), horizontal flip, zoom, translation, shearing, random brightness and contrast, and elastic deformation to generate new training samples. Gaussian noise addition was also tested but resulted in poorer performance.

Next, we trained a CNN classifier with a simple two-layer architecture. It consists of two convolutional blocks, each followed by activation functions, batch normalization, max pooling, and spatial dropout to extract and refine features while preventing overfitting. After feature extraction, the model flattens the output and passes it through a dense layer with dropout for further

	Normal	Pneumonia	Total
Training set	1341	3875	5216
Validation set	8	8	16
Test set	234	390	624
Total	1583	4273	5856

Table 2: Label distribution of images across training, validation, and test sets.

regularization, ending with a single output unit. Furthermore, the model was trained using early stopping and learning rate reduction to avoid overfitting. **(Q2.1)**

During our initial training, we noticed that the model's attributions were largely focused on the "R" mark in the X-ray images, suggesting it was relying on a spurious correlation to make predictions (see Figure 7). To address this, we cropped 25 pixels from the top and left sides of each image, as these markings usually lie on the top-left region. This removed the "R" mark from most images, encouraging the model to learn relevant features for proper classification rather than taking shortcuts. **(Q1.4)**

We experimented with deeper architectures, varying kernel and filter sizes, as well as different batch sizes and activation functions. Our best results were achieved with a compact two-layer model using 3×3 kernels and filter sizes of 32 and 64. A batch size of 8 yielded optimal performance during training. Finally, using the LeakyReLU activation function led to better results compared to standard ReLU. The model achieves **0.85 accuracy** on the test set. The table below summarizes the model's performance in more detail **(Q2.2)**:

Class	Precision	Recall	F1-score	Support
Normal	0.89	0.68	0.77	234
Pneumonia	0.83	0.95	0.89	390
Macro avg	0.86	0.81	0.83	624
Weighted avg	0.85	0.85	0.84	624

Figure 1: Detailed classification report on the test set.

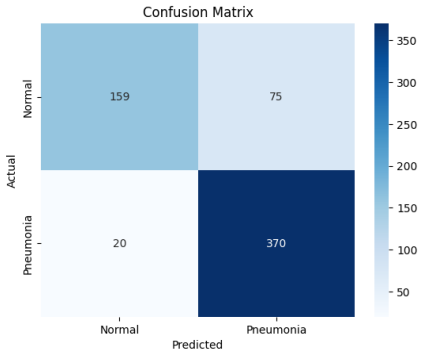


Figure 2: Confusion Matrix

To evaluate the model's predictions, we first use the Integrated Gradients method to generate attribution masks that give us insights into what the model is doing. By visualizing these masks, we observe that the model mostly focuses on pixels within the lung fields and, in some cases, on areas outside the lung region. While the model occasionally highlights irrelevant regions, the regions within the lung fields are critical for distinguishing between normal and disease samples, as pneumonia is typically identified in X-rays by the presence of opaque areas in the lungs. **(Q3.2)** Attributions are not fully consistent across samples, as some attribution maps have more subtle or no highlights, especially in the pneumonia samples. In addition, as previously mentioned, the model sometimes focuses on image details that are not related to pneumonia, such as some lines present on some samples, indicating some learned spurious correlation. However, the model mostly focuses on the right areas of the chest to make predictions. **(Q3.3)**

As baselines, we tested solid black, solid white, and mean image, finding that the solid white baseline produced the most informative visualizations. The attribution maps were heavily affected by the choice of baseline, as demonstrated in Figure 6 **(Q3.4)**. While the solid white baseline provides clear insights into the model's functionality, highlighting the clear and uninfected areas of the lungs, the solid black and mean image baselines highlight pixels around the ribs and the image's contours.

In contrast, the heatmaps generated using the Grad-CAM method effectively highlight opaque and lighter regions in the X-ray, such as the ribs, spine, heart, and other bright areas (see Figure 8a)). These areas are clinically relevant, as opaque regions in the lungs may indicate pneumonia if in unusual positions. Nonetheless, these maps also show that the model uses the patient's bone structure to make predictions, which is not sensible information for detecting pneumonia. Additionally, in some samples, the model uses irrelevant elements outside the chest region to make the prediction, revealing some learned spurious correlations (see Figure 8b)). **(Q4.2)** The outputted attributions for this method are more consistent than in the previous method, despite some variation in intensity across images, but overall, the highlighted regions remain relevant and stable across samples. **(Q4.3)**

Comparing the two explainability methods, we can conclude that they capture complementary features: Grad-CAM identifies regions likely associated with disease, while Integrated Gradients emphasizes healthy regions. **(Q4.4)**

To test the trustworthiness of the previously calculated saliency maps, we perform the Data Randomization Test. After randomly permuting the labels of all samples, we notice that the attribution maps for Integrated Gradients do not change and still emphasize the clear parts of the lungs. On the contrary, the heatmaps from Grad-CAM change, now highlighting more the region of the ribcage. This indicates that out of the two methods, only the Grad-CAM method is trustworthy and effectively captures the relation between the labels and instances. On the other hand, the Integrated Gradients method purely acts as an edge detector and does not model any relationship between the outputs and inputs. **(Q5.3)**

Part 3: General Questions

Q1: How consistent were the different interpretable/explainable methods? Did they find similar patterns?

For Part 1, the different interpretable and explainable methods were not very consistent across models, because the features had a different ranking. However, some features, such as *ST_Slope_Up, Cholesterol, ChestPainType_ATA* appeared among the top predictors across methods, suggesting a pattern in the data. (**Q1.1**) With regard to Part 2, the patterns observed from the two methods were quite different yet consistent. While Integrated Gradients focuses more on the darker regions of the image, suggesting that the model uses the absence of opacity as a key indicator of normal lungs, Grad-CAM highlights the lighter, more opaque areas typically associated with infection. We can thus conclude that both methods distinguish between normal and pneumonia cases by evaluating the contrast between clear and opaque regions in the lungs. Integrated Gradients emphasizes areas that support the absence of disease, while Grad-CAM highlights regions that may include infection. **(Q1.2)**

Q2: Given the "interpretable" or "explainable" results of one of the models, how would you convince a doctor to trust them? Pick one example per part of the project.

Taking logistic lasso regression as an example, we would show the doctor not only the performance of the model but also the feature importance plot. We would explain to the doctor that the model shows each feature's importance in having an effect on heart disease risk. The doctor can verify with his medical knowledge if the plot makes sense with the prediction. (**Q1.2**) Using the Grad-CAM method, we would convince a doctor to trust the model by explaining that the heatmap generated by Grad-CAM highlights the specific regions of the medical image that the model focused on to make its prediction. Using this interpretable output, the doctor can verify if the model is using clinically relevant regions of the X-ray and thus understand if the prediction makes sense or not. These interpretable outputs therefore allow doctors to visualize the outputs of black-box models, which makes them more trustworthy to use, as doctors can get an intuitive sense of what the prediction is based on. In addition, showing the result of the Data Randomization Test could also be an interesting way of showing that the model learns the relationships between the label and the input image. **(Q2.2)**

Q3: Elaborate whether the feature importances from the interpretability/explainability methods intuitively make sense to find the respective disease.

The interpretability methods do identify features. If we take a look at the plots of the three models, we can see that, e.g., *ChestPainType_ATA and Cholesterol* have a relatively high feature importance, which means that those features are highly likely to contribute to heart disease. This lines up with the medical knowledge. (**Q3.1**) Regarding part 2, the feature importances from the methods intuitively make sense, as they use the contrast in the image to infer relevant regions representing the absence/presence of infection, although in some samples it is clear that the model exploits some elements outside the lung region to make the prediction despite the usage of regularization. **(Q3.2)**

Q4: If you had to deploy one of the methods in practice, which one would you choose and why?

For Part 1, if we were to deploy one method in practice, we would choose the NAM because it combines the flexibility of MLPs and the interpretability of logistic lasso regression. Even though all three models perform the same, it might be possible that we get more data in the future, and we would like to retrain it with the larger data set, and that is where neural networks are superior. (**Q4.1**) Regarding Part 2, we believe that deploying Grad-CAM in practice would be more beneficial, as it offers superior interpretability compared to the Integrated Gradients method. While Integrated Gradients highlight individual pixels in the image that contribute to the model's output, Grad-CAM produces a heatmap based on the convolutional feature maps that identifies the most relevant regions of the image. Grad-CAM is thus likely to be more intuitive and trusted by clinicians, thereby increasing its potential for practical adoption. In addition, Grad-CAM is an interpretability technique specifically designed for vision tasks, whereas Integrated Gradients is a general technique for Deep Neural Networks, which makes Grad-CAM particularly well-suited for medical imaging. Finally, after label permutation, the attribution heatmaps changed, which shows that Grad-CAM accurately captures the relationship between instances and their labels in contrast to Integrated Gradients. **(Q4.2)**

Appendix A

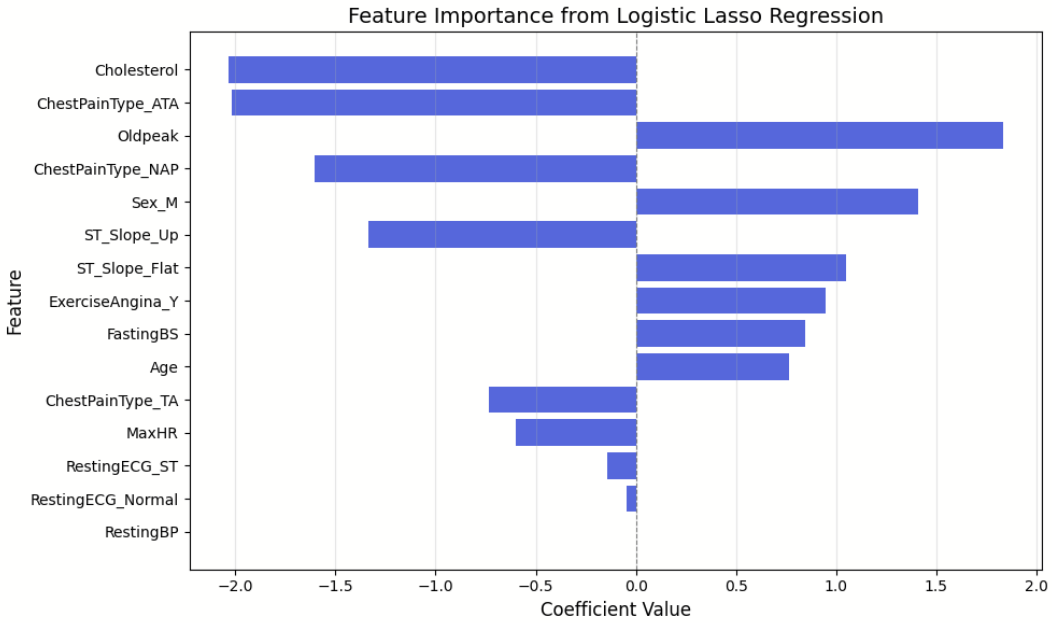


Figure 3: Feature Importance from Logistic Lasso Regression

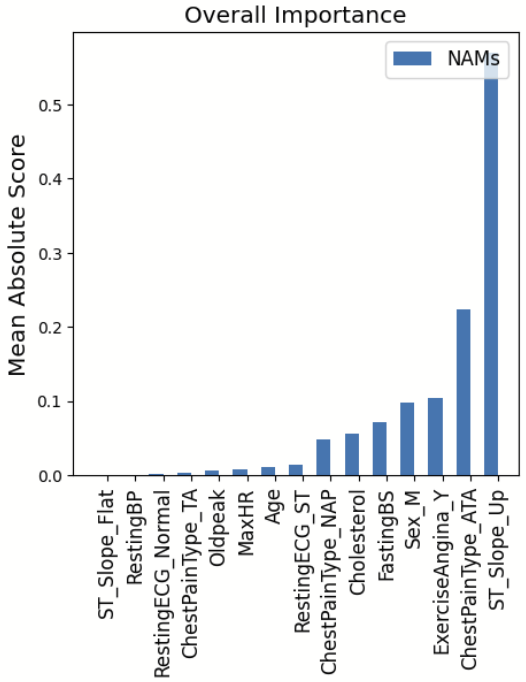
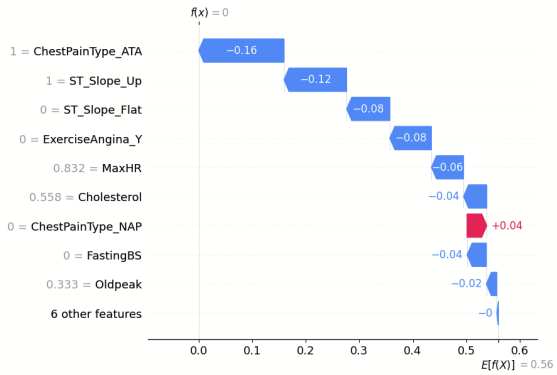
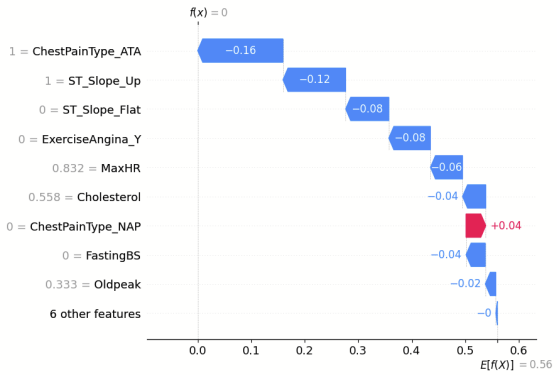


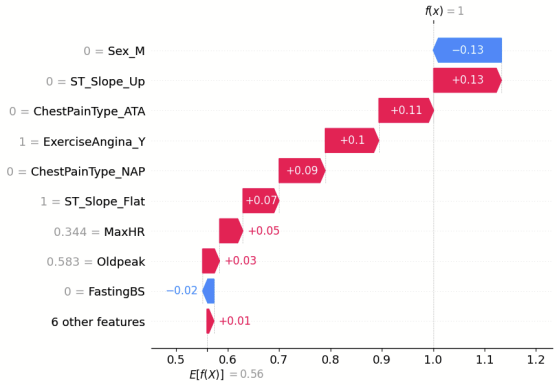
Figure 4: Feature Importance from NAM model



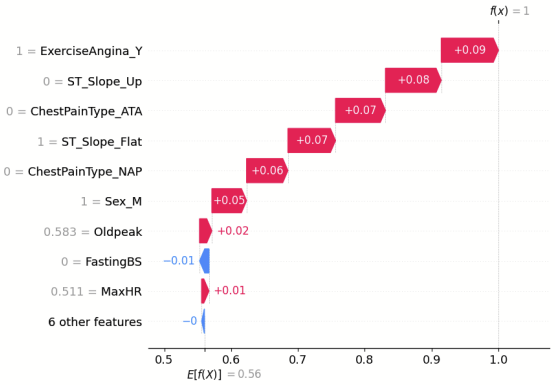
(a) Feature importance of negative sample 1



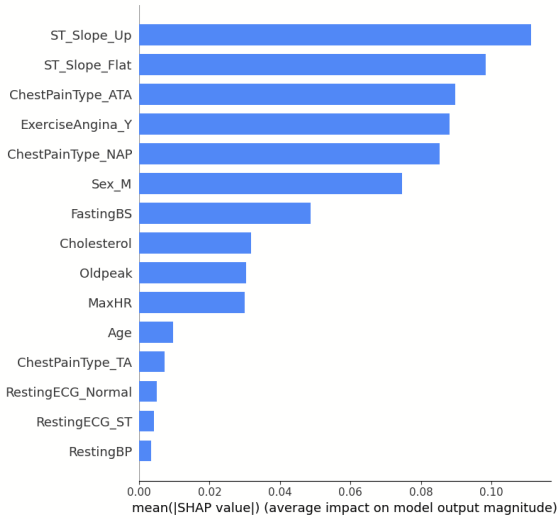
(b) Feature importance of negative sample 2



(c) Feature importance of positive sample 0

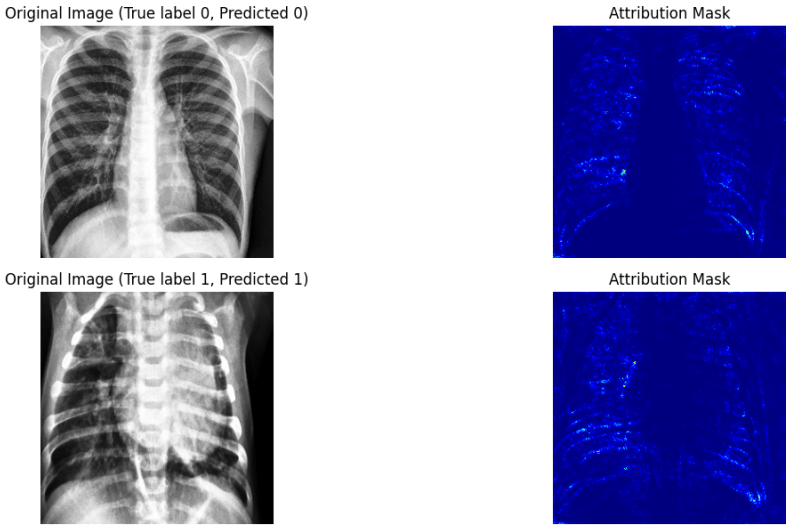


(d) Feature importance of positive sample 1

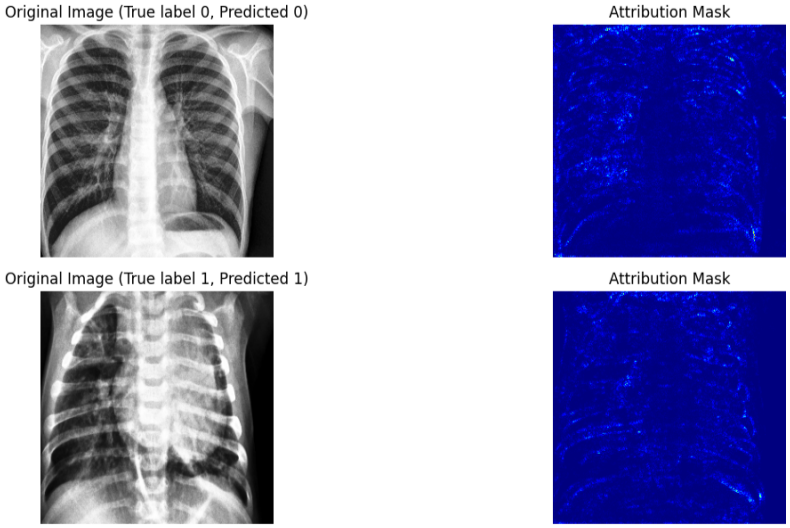


(e) Overall feature importance

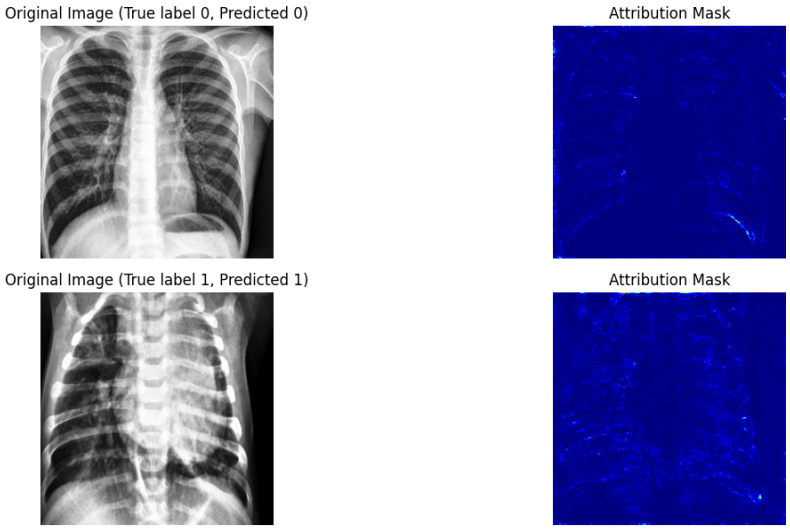
Figure 5: Feature importance of 2 negative and 2 positive samples with as well asoverall feature importance



(a) Solid white baseline



(b) Solid black baseline



(c) Mean image baseline

Figure 6: Visual comparison of attribution maps from Integrated Gradients using different baseline inputs

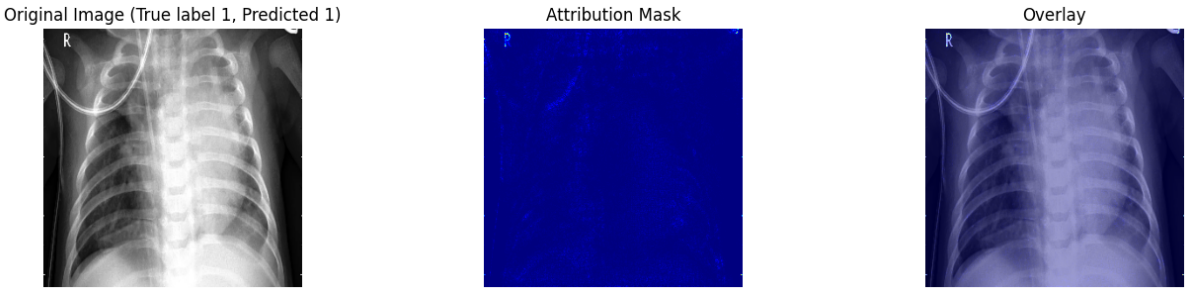
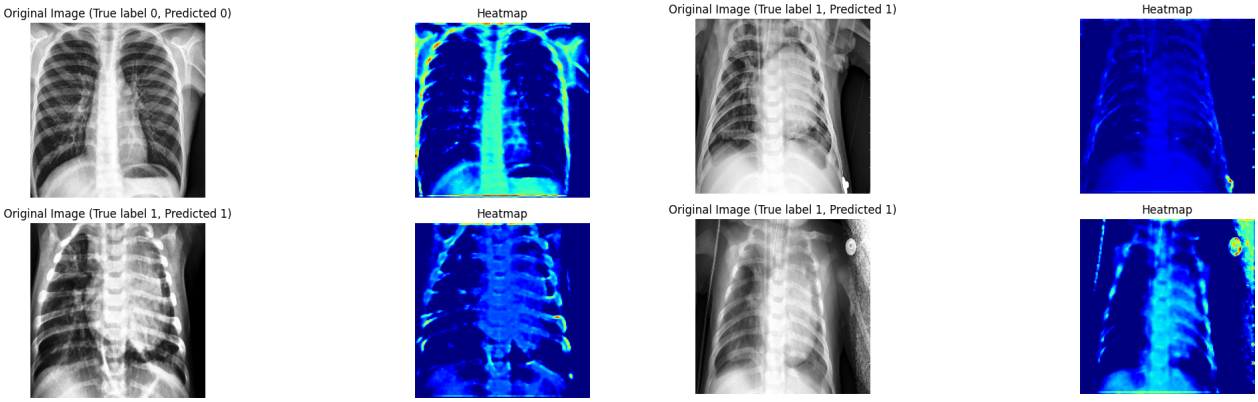


Figure 7: Integrated Gradients attribution revealing CNN reliance on spurious correlation with the ’R’ marking



(a) Sensible heatmaps

(b) Irrelevant or inconsistent elements

Figure 8: Visualization of Grad-CAM saliency heatmaps