Post-hoc explainable AI method to improve accuracy and understanding of cardiac image classification models for arrhythmia detection and other heart diseases

Arnav Jhingran

Project Hypothesis

Is it possible to balance the tradeoff of *understanding* and *accuracy* ---thus understand a deep learning model for cardiac image classification and improve its accuracy?

My goal is to define a new approach, XAI2ROI, to include explainable AI techniques to understand the behavior of published black box deep learning models for cardiac image classification and apply the *explanation* learning approaches to identify the *regions of importance* (ROI) which can then be used to retrain the original model to improve its accuracy. Thus making it possible to increase the *accuracy* of the original blackbox model for cardiac image classification while *understanding* its behavior to explain it better.

Background

Today the most successful deep learning models applied in healthcare to diagnose diseases using diagnostic images (e.g., ECG, echocardiograms, X-rays, MRIs) are called “black-boxes” making it difficult to *explain* why the model reached a certain classification decision. Especially in medicine, a doctor needs to be able to explain, when using a deep learning tool to assist in diagnosis, what features and factors played a role in the final outcome. This has led to a lot of new work on explainable AI techniques (XAI).

Deep learning is based on a “deep” multi-layered neural network which is a machine learning technique patterned after the human brain. It consists of an input layer, multiple hidden layers and an output layer. Multiple types of deep learning networks have been researched for various applications from image classification, natural language processing to self driving cars. Some of the models most useful for wave or image classification are generic Feed forward networks (FFN) or a class of them called convolution neural networks (CNN). CNNs are used

most commonly for image classification where a filter (a convolution function) is applied to fixed size subset of an image to identify components of an image and then classify the image. Both FFN and CNNs do not have cycles or loops. Other more complex network types like Recurrent neural networks (RNN) and long short term memory (LSTM) have loops and hidden state that is useful to "remember" previous input but do not apply that well to wave and image classification. CNN based deep learning models have become common in medical diagnostics where the data is from diagnostic images generated from X-rays, MRIs, ECG, echocardiograms.

To *understand* and explain how a deep learning model reached a certain medical diagnosis various XAI techniques are being developed. Most practical among these are post-hoc techniques where the explanation can be added after the original model analysis is done. With post-hoc analysis, for example, we can go back and understand which input features or image regions contributed most to the final result. Some known post-hoc XAI techniques are: LIME (locally interpretable model-agnostic explanation), LRP (layer-wise relevance propagation), BETA (Black Box Explanation through Transparent Approximation). For image processing using CNN models, the explainability method **LIME** is most often cited for being easy to use and has good performance to understand complex models unlike the other approaches.

The basic idea of LIME is to alter the images systematically and see the impact it has on the blackbox model's classification. Once the blackbox deep learning model is trained on a set of images, a subset of images are taken and perturbed. This perturbation is done by graying out a small region of the image. The perturbed image is then tested with the original model to identify which region had the most impact on the final classification. The LIME based explanation can be viewed using a heat map on the original image which helps human understanding.

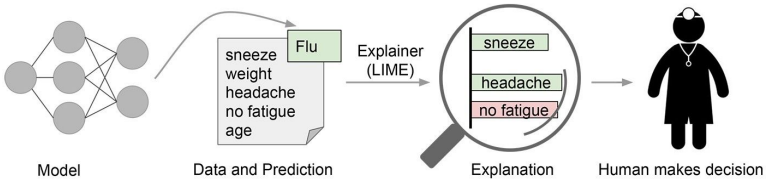

Fig 1: LIME application in medical diagnostic models. Image courtesy:
https://towardsdatascience.com/understanding-model-predictions-with-lime

The application of deep learning in healthcare is across various fields of diagnostic medicine. Given that cardiovascular disease is the leading cause of death (1 in 4 deaths are due to cardiovascular diseases), one of the most widely studied areas for applying deep learning is for cardiac imaging, especially analyzing ECG (electrocardiogram) waveforms and echocardiograms (echo) images which are the first set of diagnostics tests used for any heart ailment.

Electrocardiogram (ECG) can be used to detect arrhythmias (irregular heart beats) and Atrial fibrillation (Afib) which is a type of arrhythmia that can cause heart failures.
Echocardiogram (echo) is the series of ultrasound images capturing the flow of blood during a cardiac cycle (heart beat). Echo images are used to detect the pumping effectiveness of the heart especially the ejection fraction (EF) of the left ventricle (LVEF), lower values of which can indicate serious heart ailments like cardiomyopathy.

Method

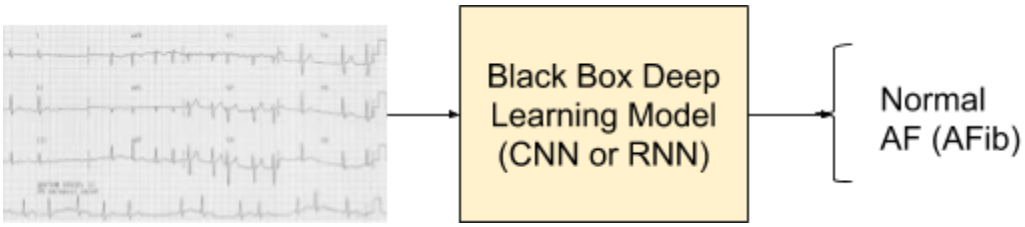

Figure 1: An ECG waveform showing Artirial fibrilation (Afib). Image courtesy: ecgwaves.com. A black deep learning model can classify Afib in the ECG waveform with pretty high accuracy. A published box model is used as the baseline.

To prove my hypothesis I will use two published models for cardiac disease classification as my baseline control.

The first model is for ECG classification for AFib detection. For this I will use multiple datasets of ECG waveforms that identify normal and Afib waveforms. Using these datasets I will use a published model (https://stanfordmlgroup.github.io/projects/ecg/) as the baseline, train the model using the ECG waveform datasets and measure the accuracy using that model for classifying the validation/test data for Afib (as in Fig 1).

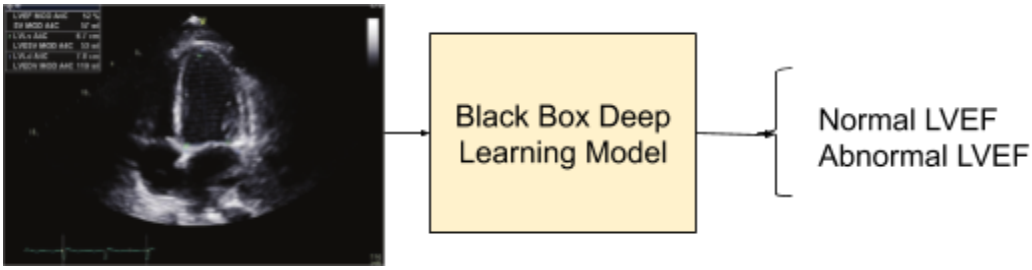

Figure 2: An echocardiogram dicom image sample showing left ventricular ejection fraction (LVEF) computation. Image courtesy: researchgate.net. A blackbox deep learning model can classify the abnormal LVEF in the echo dicom videos with pretty high accuracy. A published box model is used as the baseline.

The second model is for echo video analysis to estimate left ventricular ejection fraction and identify abnormal values. For this I will use a dataset of echocardiogram videos that has the left ventricular ejection fraction computed to predict cardiomyopathy. Using this dataset I will use another published model (https://douyang.github.io/EchoNetDynamic/) as the baseline, train the model using the Echo dicom video dataset and measure the accuracy of that model for determining the LVEF (as in Fig 2) . The published models will be the control giving the baseline values of Accuracy and F1 scores for the corresponding datasets.

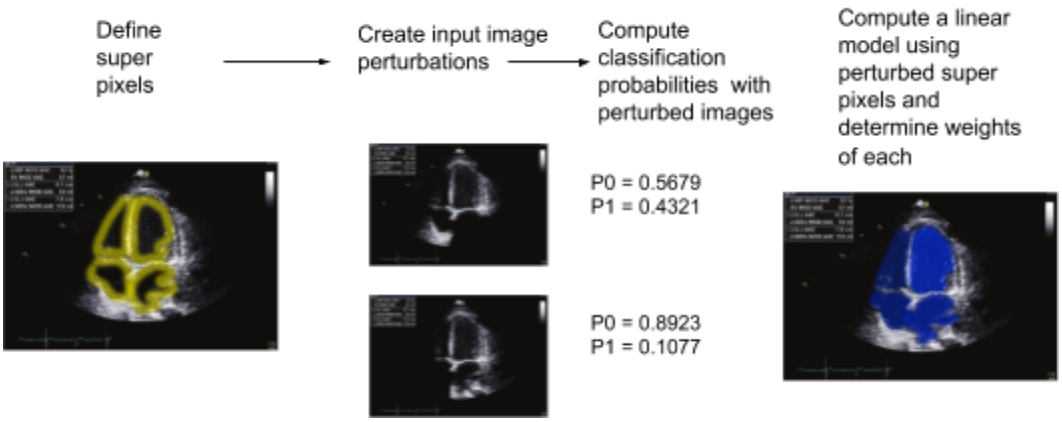

Figure 3: The XAI method called Locally interpretable model-agnostic explanation (LIME) identifies the weights of the super pixels in an echocardiogram image. The new linear classifier tries to explain the published black model classification by determining the weights of the super pixels which are regions of pixels in the original image. The weighted superpixels can now be used by my approach to identify regions of importance (ROI) in the image, the left ventricle region had 64% weight while the right had 31%.

My approach does the following steps:

1. Use the published models to create the baseline blackbox deep learning models for each type of cardiac classification: Afib identification for the ECG images and abnormal LVEF identification for the echocardiogram dicom video images.
2. Post-hoc analysis for local interpretations of the published models using the explainable AI techniques (XAI) called Locally Interpretable Model-agnostic explanation (LIME) which is a model agnostic approach to understand the black-box deep learning models. The key concept underlying this approach is perturbing the inputs and watching how doing so affects the model's output.
3. For each dataset used in the control I will pick a subset of the dataset (varying the fraction selected for each experiment) and apply the XAI technique to identify and assign weights to the input features or superpixels/slices used by the baseline models.

4. Using these XAI generated indicators of the value of each super pixel/slice to the black box model's classification my approach will define a weight threshold and generate a set of super pixels/slices that are important to the final classification.
5. These set of super pixels/slices above the threshold weight are the "***regions of importance***" in the image. Consider a super pixel like a bounding box defined by the <Xmin, Ymin, Xmax, Ymax>
6. My approach then retrains the baseline blackbox models with a focus on the region of importance (ROI) using the ROI pooling layer approach. ROI pooling has been used in image analysis by using identified objects as regions of interest that are made the "focus" area for the deep learning layers.
7. Using the same validation data set as the baseline we recompute the accuracy and F1 score with my retrained model.
8. Compare the accuracy scores with the baseline by varying the explained dataset fraction and the ROI weight threshold.

Materials

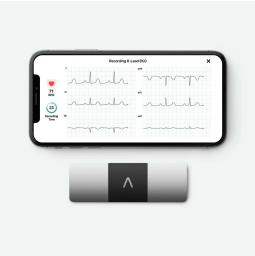

Figure: The hand-held personal ECG device (Image courtesy: AlivCor)

I used the following hardware and software packages for various stages of data processing.

1. KardiaMobile (from AlivCor) finger-based home EKG device to view an EKG on a smart phone.
2. Anaconda 5.0 Python distribution which includes the Jupyter notebook and the Spyder Python dev environment.
3. Python included the libraries:
    a. matplotlib (to plot the data in the intermediate stages),
    b. numpy (package for various scientific and statistical functions),
    c. TensorFlow 1.2.0 (the deep learning/neural network package),
    d. PyTorch (another deep learning framework)
    e. Pydicom (to read and view dicom images),
    f. Keras library (https://keras.io/).
    g. Scikit-learn 0.22.2 (machine learning package for regression)

h. Lime library (explainable AI functions for LIME)
i. Roi-pooling library (for ROI pooling functions)

Detailed Procedure

For the baseline blackbox model (ECG_BASE) of the ECG image classification to identify Afib I use the ECG Physionet dataset (ECG-1) and the MIT_BIH dataset (ECG-2). The published model (https://stanfordmlgroup.github.io/projects/ecg/) which I call ECG_BASE uses a 34 layer convolutional neural network (CNN) model to detect arrhythmias in arbitrary length ECG time-series. The model can classify noise, normal sinus rhythm and segment twelve arrhythmia types present in the time-series and was trained and validated using a proprietary dataset from irhythm technologies. I compiled the mode and trained it on the ECG-1 and ECG-2 datasets sampled at 200Hz and a sequence of annotations for every second of the ECG as supervision. While the original model is trained to classify 12 different arrhythmia classes in my baseline it is used to only identify 4 classes: noise, other, normal sinus rhythm and AFib. Each ECG signal is sampled at 200Hz. The ECG_BASE model consists of 33 layers of convolution followed by a fully connected layer and a softmax activation filter. The network consists of 16 residual blocks with 2 convolutional layers per block. The convolutional layers all have a filter length of 16 and have 64k filters. Before each convolutional layer the model includes a Batch Normalization (BN) step and a rectified linear activation (ReLU) filter following a pre-activation block design. The first and last layers of the network are special cased due to this pre-activation block structure. The final layer consists of a fully connected layer and softmax activation. The output is a distribution over the 4 output classes for each time-step. In the final step I introduce the ROI pooling layer after the convolutions steps using a Keras library function to add layers to a CNN model.

My approach consists of 3 phases as shown in the flowchart below.

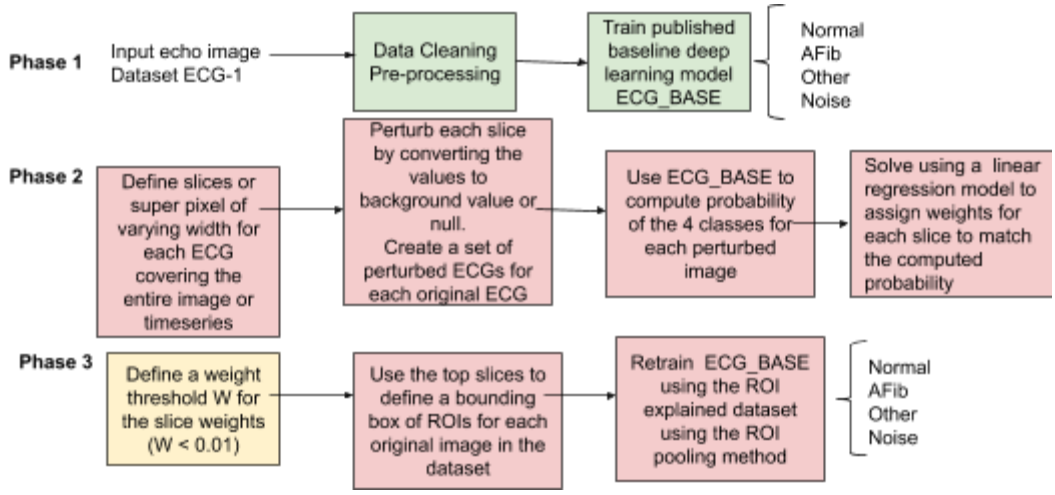

Figure 4: My ROI (regions of importance) generation using the LIME explainable approach. The Phase 1 trains the original published blackbox model with the dataset(s). In Phase 2 the LIME explainable AI approach is used to find the slices that contribute the most to classify the image for Afib. In Phase 3 using the weights from Phase 2 the regions of importance (ROIs) are created in the original image and used as input to the ROI pooling layer added to retrain the blackbox model.

Phase 1 is used to train the ECG_BASE with the training datasets. I use the normal input dataset of ECG-1 and ECG-2 to train the ECG_BASE model with 90% of the data and use 10% of the dataset for validation to produce a probability distribution over the classes sinus, noise, Afib and other. These will be the baseline control scores for comparing my XAI2ROI approach with.

In *Phase 2*, I use the post-hoc XAI technique called the LIME method for explaining the ECG_BASE models. For 10% of the dataset images in ECG-1 which I call "explained dataset", ECG-1-EXP, I pre-process the images to create slices each of varying widths. Then for each image I create 64 perturbed images by setting one of the slices to the background value of null. For each perturbed image I compute the classification generated by ECG_BASE. Using the perturbed images and the slice controlled classification value generated by ECG_BASE, I use a standard linear regression model to assign weights to the slices based on what they contributed to the final classification probability by ECG_BASE.

In *Phase 3*, I determine the "regions of importance" ROI in the original images of the "explained dataset" ECG-1-EXP. For each super pixel or slice whose weight was below the threshold W_t

(varying W_t from 0.001 to 0.01) the bounding block is masked by making it a null value in the ECG time series. The super pixels or slices of weight > W_t are the regions of importance of the image as determined by the explainable AI approach. This ROI identified image subset is called ECG-1-EXP-ROI. I retrain the ECG_BASE model with this post-processed dataset using the ROI pooling layer approach as the regions of importance of the images are already pre-identified. ROI pooling allows to focus the deep learning model's training on the data within the bounding box of the ROIs. I recompute the accuracy and confusion matrix using the validation dataset and compare it with the baseline ECG_BASE.

I generate results varying two variables: the size of the explained data set (from 5% to 10%) and the threshold ROI weight, W_t. For each dataset I use a fraction of the original dataset to create explained dataset (varying from 1% to 5% of the total images) and then retrain the models and compute the accuracy and F1 scores. For each of the explained dataset size, I vary the value of W_t from 0.001 to 0.01 (in 10 increments) and determine the final accuracy scores and the confusion matrix. The confusion matrix plots the TP (true positive), TN (true negative), FP (false positive), FN (false negative) values which are used to determine accuracy (TP+TN/TP+TN+FP+FN) and F1 score. The F1 score is the harmonic mean of the precision and recall 2*precision*recall/(precision + recall), where precision = TP/(TP+FP) and recall = TP/(TP+FN).

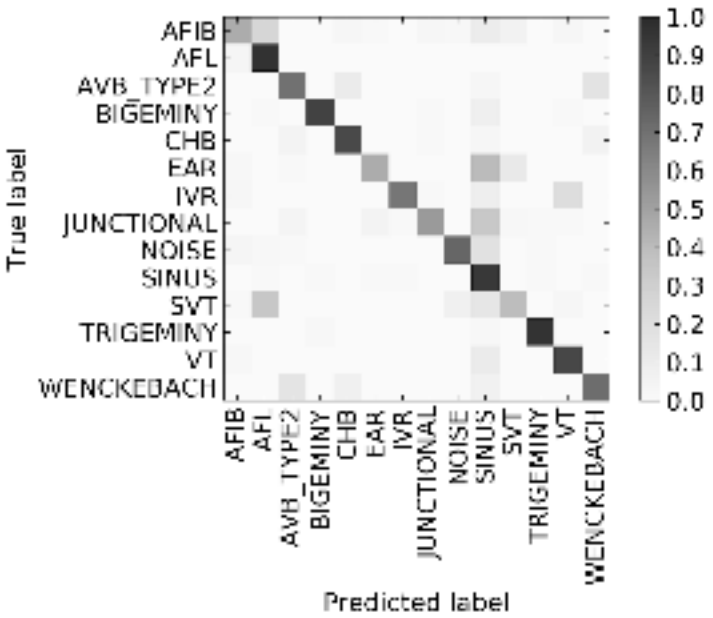

Figure 5: The original confusion matrix plot using the ECG_BASE deep learning model. This plot shows 12 classifications for various arrhythmia types using the non-public irhythm dataset. In my tests I focus only on the AFib class and the normal SINUS class from public datasets like Physionet. Data Courtesy: https://stanfordmlgroup.github.io/projects/ecg/

Experimental Results

Training Datasets

My project uses the training dataset of cardiac images from multiple publicly available sources. The Stanford Echo-Net dataset (ECHO-1) includes the Echocardiogram videos and images from over 10,025 unique patients from 2016 and 2018 from the Stanford Hospital. The echo data is in the form of Dicom videos each containing 50-100 dicom images. Each dataset label includes the age, gender, image size and the measured value of the left ventricular Ejection Fraction (EF). The focus of the deep learning model is to determine the value of EF and validate that with value in the dataset label. EF is an important measure of cardiac function and is used to predict and diagnose a number of heart diseases including heart failure.

The Physionet ECG (ECG-1) dataset consisted of 12,186 ECG recordings of a single lead ECG recordings collected from AliveCor from their customers using the finger-based ECG device for personal use. Each recording was taken by AliveCor's customers by pressing the two electrodes using the left and right hand fingers creating a single lead ECG recording equivalent of the Left-arm Right-arm value (LA-RA). Four classes of data were labelled normal rhythm, AF rhythm, other rhythm and noisy recordings.

MIT-BIH arrhythmia database (ECG-2) provided by the Massachusetts Institute of Technology. It comes from 47 clinical patients and contains 48 annotated ECG records. Each group is approximately 30 minutes long and is sampled at a rate of 360 Hz by a 0.1–100 Hz band pass filter, for a total of approximately 650,000 sample points.

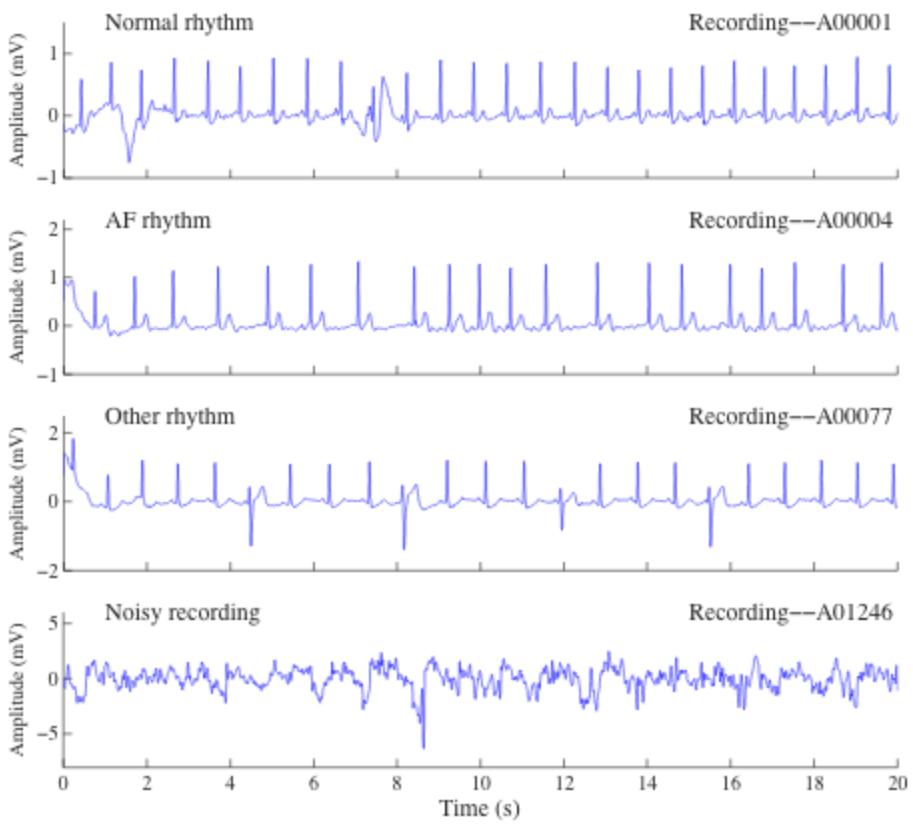

Figure 6: A sample recording plots of the ECG data from ECG-1 dataset (from physionet.org) showing normal and AFib waveforms along with noise.

For field testing I use AlivCor's (https://www.alivecor.com/) KardiaMobile device that can collect 6-lead or single lead ECG data.

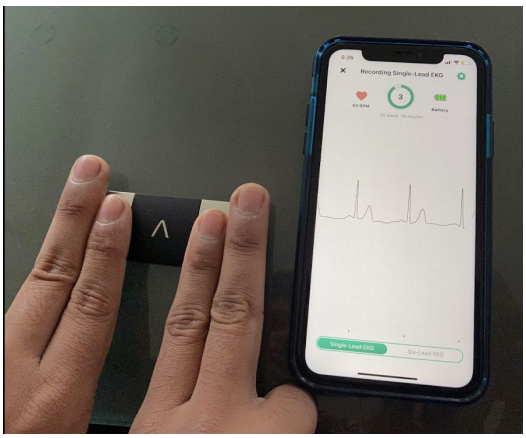

Figure 7: The hand-held personal 6-lead ECG device from AlivCor for field testing as needed.

Metrics

We use the confusion matrix for evaluation and it can be used to compute the five metrics to evaluate the performance of the baseline models and my approach, namely: accuracy, recall, precision, specificity, and *F*1 score. Accuracy is the proportion of correctly classified images (e,g, ECG waveforms or echo image) to the total number of images in the dataset. The *F*1 score is the harmonic mean of precision and recall.

Validation

The dataset is split into the training and validation dataset as 90-10%. Then another fraction of the validation dataset is used to create the XAI based approach to determine the input data "impact". The model is retrained with the new labelled input and the new accuracy and F1 measures are determined. The XAI-labeled classification performance is compared by evaluating the confusion matrix and the accuracy and F1 score for the baseline deep learning algorithm for the different datasets.

Class	Precision	Recall	F1-score	Support
A	0.843	0.875	0.859	80
N	0.902	0.909	0.906	508
O	0.765	0.755	0.760	233
~	0.741	0.645	0.690	31
Avg/Total	0.853	0.854	0.854	852

ECG-1 Average: 0.841679

Table 1: The precision and recall statistics of the Afib classification using my retraining with the ECG-1 dataset of the original model from https://stanfordmlgroup.github.io/projects/ecg/. The class labels are A=Afib, N=Normal, O=Other and ~=Noise. Training dataset had 7676 ECG samples and the validation dataset had 852 ECG samples. My

training was over 100 epochs with each epoch having 239 steps.

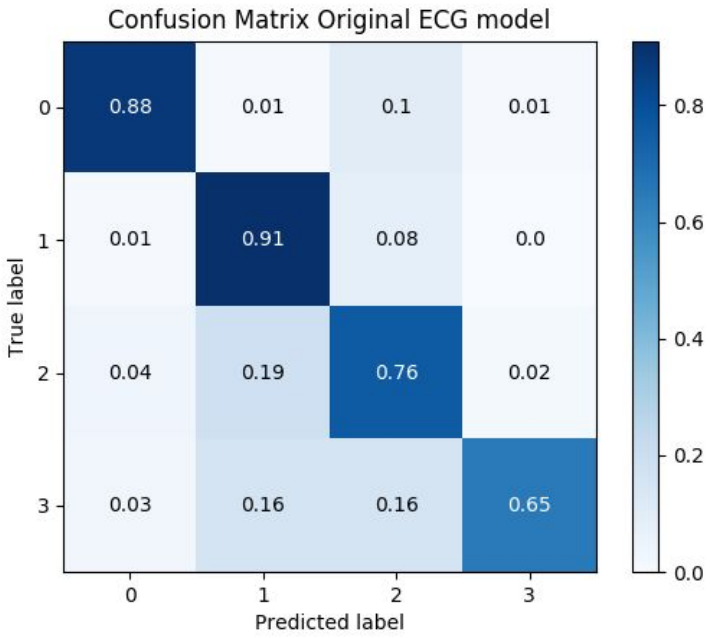

Figure 6: The confusion matrix of Afib classification using my retraining with the ECG-1 dataset of the original model from https://stanfordmlgroup.github.io/projects/ecg/. The labels are 0='A', 1='N', 2='O', 3='~', where A=Afib, N=Normal, O=Other and ~=Noise.

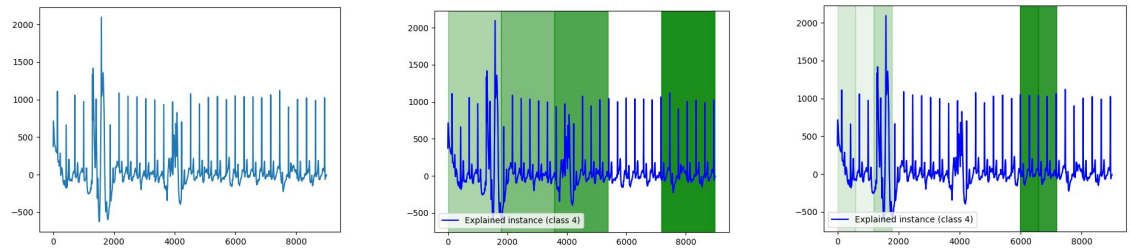

Fig7: **The normal ECG**. The LIME XAI technique is used to identify the heatmap of the various slices (also called super pixels for images). As ECG is a time series of mV values using the LIME explainable AI approach I just dampen the value in the slice to null and compute the weights. The different slice widths (num_slices=5 and 15) are shown. The color density is based on the weight of the slice on the original deep learning model's prediction.

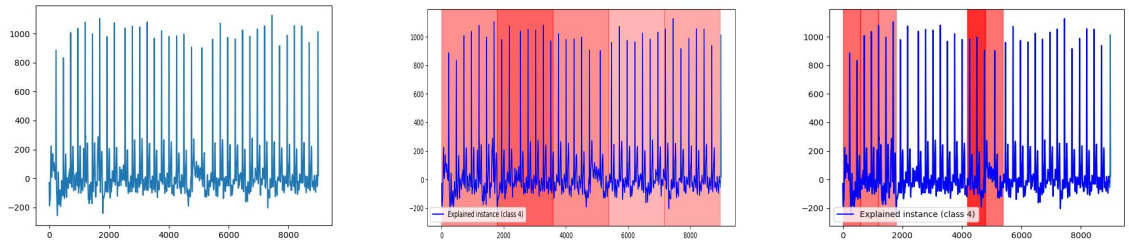

Fig8: **The AFib ECG**. The LIME XAI technique is used to identify the heatmap of the various slices. As ECG is a time series of mV values using the LIME explainable AI approach I just dampen the value in the slice to null and compute the weights. The different slice widths are shown. The color density is based on the weight of the slice on the model's prediction.

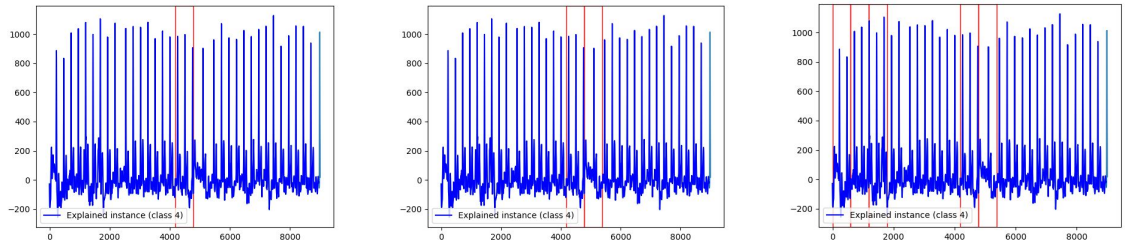

Fig8: **The AFib ECG ROI**. Post the LIME identification of the heat map per slice thresholding is used to identify the regions of interest (ROI) using the num slices =15. The left most threshold picked the max weight, the middle picked the top 2 weights and the other picked all slices. This ROI bounding box is used by the ROI pooling layer.

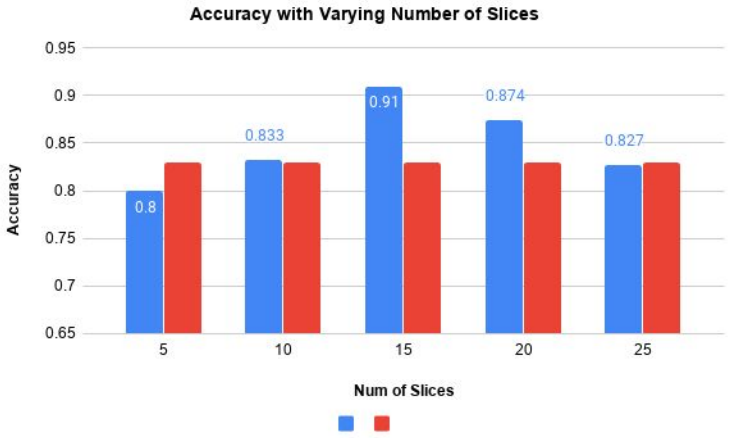

Fig9: XAI2ROI technique is used to map the LIME identified regions (the ones above the threshold weight) into a ROI pooling layer and retrain the original deep learning model. The overall accuracy on the validation set is measured based on the slice width of the LIME based region. Here the num of slices =15 had the best accuracy of 0.91. The blue bars are with the new training and the red bars are with the original model.

Class	Precision	Recall	F1-score	Support
A	0.859	0.912	0.885	80
N	0.914	0.923	0.919	508
O	0.803	0.785	0.794	233
~	0.731	0.613	0.667	31
Avg/Total	0.872	0.873	0.872	852

ECG-1 Average with XAI2ROI: 0.865827

Table 2: The precision and recall statistics of the Afib classification using XAI2ROI with new ROI pooling layers on the ECG-1 dataset. The class labels are A=Afib, N=Normal, O=Other and ~=Noise. The same training dataset was used which had 7676 ECG samples and the validation dataset had 852 ECG samples. My training was repeated over 100 epochs with each epoch having 239 steps. The overall accuracy increased to 0.865. The num slices for LIME was = 20. Only the slice with max weight was used to define the ROI bounding box.

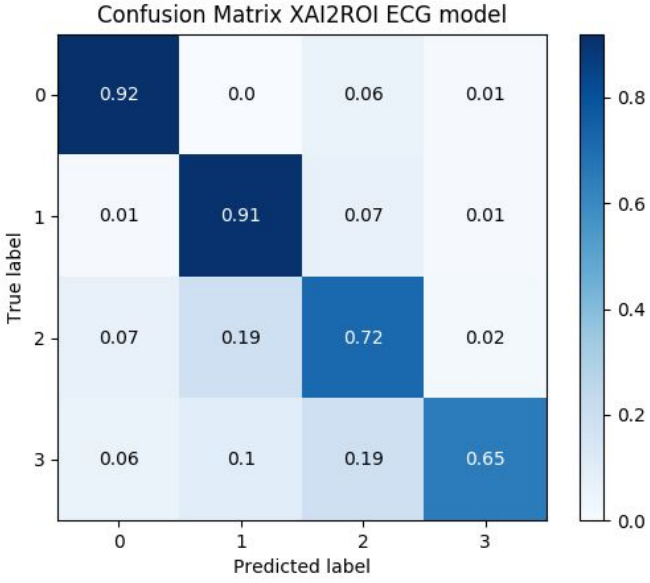

Figure 7: The corresponding confusion matrix of Afib classification using XAI2ROI with new ROI pooling layers on the ECG-1 dataset. The labels are 0='A', 1='N', 2='O', 3='~', where A=Afib, N=Normal, O=Other and ~=Noise. The ROI focus improves the accuracy of the Afib classification from 0.88 in the original to 0.92 with XAI2ROI

Overall I proved that XAI2ROI has the potential to explain a complex published deep learning model and increase its accuracy. In the ECG data set using the ECG_BASE model the accuracy from 84% to 91% in the best case and to 86.5% in the average case.

Bibliography

1. L. Gilpin et al, Explaining Explanations: An Overview of Interpretability of Machine Learning, Feb 2019, https://arxiv.org/pdf/1806.00069.pdf
2. A. Ratner et al, Weak Supervision: The New Programming Paradigm for Machine Learning, https://hazyresearch.github.io/snorkel/blog/ws_blog_post.html
3. V. Chen at el, Slice-based Learning: A Programming Model for Residual Learning in Critical Data Slices, https://arxiv.org/abs/1909.06349
4. J. Roney et al, Deep weakly-supervised learning methods for classification and localization in histology images: a survey, https://arxiv.org/abs/1909.03354
5. S. Petersen et al, Artificial Intelligence Will Transform Cardiac Imaging—Opportunities and Challenges, Frontiers in Cardiovascular Medicine, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6746883/
6. D. Ouyang et al, Interpretable AI for beat-to-beat cardiac function assessment, https://www.medrxiv.org/content/10.1101/19012419v2.full.pdf
7. P. Rajpurkar et al, Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks, https://arxiv.org/abs/1707.01836
8. M. Ribeiro et al, Why should I trust you? Explaining the predictions of any classifier, https://arxiv.org/pdf/1602.04938.pdf
9. A. Isin et al, Cardiac arrhythmia detection using deep learning, https://www.sciencedirect.com/science/article/pii/S187705091732450X