Utilizing Signal Representation Techniques on Electrocardiogram Data to Identify for Lung Obstruction using Computer Vision Models

Mehlam Shabbir, *UNF, School of Computing,* Dr. Liu Xudong, *UNF, School of Computing,* Dr. Mona, Nasseri, *UNF, School of Engineering,* and Dr. Helgeson, Scott, *Mayo Clinic*

Abstract

Electrocardiogram (ECG) data is crucial for diagnosing cardiovascular disorders. In this study, we employed different convolutional neural network (CNN) architectures, including VGG16, LeNet5, AlexNet, and a simple custom-designed CNN model, to classify patients with obstructive cardiovascular conditions. For the majority of the models, we generated images using the Short-Time Fourier Transform (STFT) signal representation technique for each of the 12-lead ECG data samples and trained the CNNs on these image representations. We also applied transfer learning on the VGG16 model to leverage pre-trained features for our classification task. In addition, we trained a simple CNN directly on the raw 12-lead ECG data for comparison purposes. Our results suggest that the current image generation approach may require further refinement. The models demonstrated a tendency to overfit to the training set, learning the noise instead of recognizing meaningful patterns, leading to poor generalization and inaccurate predictions on the test set. Future work will focus on optimizing the image generation process and exploring alternative deep learning models, such as simple Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRU), as well as further investigating the potential of training CNNs directly on raw ECG data and the effectiveness of transfer learning techniques to enhance diagnostic accuracy and clinical utility.

I. Introduction

Cardiovascular disorders are among the leading causes of morbidity and mortality worldwide, making accurate and timely diagnosis crucial for improving patient outcomes. Electrocardiogram (ECG) data, which records the electrical activity of the heart, is widely used as an essential diagnostic tool for identifying cardiovascular conditions. However, manual interpretation of ECG data can be time-consuming and prone to error. In recent years, deep learning techniques, specifically convolutional neural networks (CNNs), have shown promise in automating and enhancing ECG data analysis for cardiovascular disorder classification.

This paper investigates the performance of different CNN architectures, including VGG16, LeNet5, AlexNet, and a custom-designed CNN model, in classifying patients with obstructive cardiovascular conditions. In order to facilitate the use of CNNs on the ECG data, we employed the Short-Time Fourier Transform (STFT) signal representation technique to generate images for each of the 12-lead ECG samples. This image-based approach aimed to leverage the innate strength of CNNs in image classification tasks.

Furthermore, we applied transfer learning to the VGG16 model, which allowed us to utilize the pre-trained features of this architecture in our classification task. This technique has been effective in reducing the amount of training data and computational resources required for achieving high performance in various deep learning tasks. In contrast, we also trained a simple CNN directly on raw 12-lead ECG data to assess the feasibility and performance of this approach.

Despite the potential advantages of CNNs and transfer learning in ECG data analysis, our findings suggest that the current image generation approach may require further optimization. The models displayed a tendency to overfit to the training data, learning noise rather than recognizing meaningful patterns. This resulted in poor generalization and inaccurate predictions on the test set. Consequently, future work will focus on improving the image generation process and exploring alternative deep learning models, such as simple Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRU). Additionally, we aim to further investigate the potential of training CNNs directly on raw ECG data and assess the effectiveness of transfer learning techniques in enhancing diagnostic accuracy and clinical utility. Overall, this study contributes to the growing body of research on leveraging deep learning techniques for improving cardiovascular disorder classification and diagnosis, with the ultimate goal of enhancing patient care and outcomes.

II. Research Methodology

The study aimed to investigate the use of signal representation techniques and convolutional neural network (CNN) models to analyze a large dataset of electrocardiogram (ECG) recordings from patients at the Mayo Clinic. The dataset comprised 47,525 patients, and each 12-lead ECG recording had been labeled 0 for non-obstruction and 1 for obstruction in patients. To process the ECG data, a signal representation technique, specifically the short time Fourier transform, was applied. This technique helped to analyze the time-varying frequency content of the ECG signal, which can provide insights into the patient's cardiac health. Signal representation methods are techniques used to represent signals like time-series information such as an

audio file, in a concise and meaningful manner. The goal of signal representations is to extract the most important features of a signal, which can be used for further analysis and processing.

Next, the dataset was split into three subsets: training, validation, and testing. The training set was used to train the CNN models, while the validation set was used to adjust the hyperparameters of the model and prevent overfitting. The test set was used to evaluate the performance of the trained model, mainly based on categorical accuracy and loss. Different CNN models were used in the study, including VGG16 transferred learning model, LeNet5, and AlexNet. The dataset initially has an imbalance between the positive and negative class labels since in the real-world, there are more patients who do not suffer obstruction compared to those who do. Keeping this in mind, we wanted to factor this imbalance of classes out of the equation, hence the dataset that we trained on, balances the class label to have equal number of positive and negative samples.

After the models were trained and tested, the results were analyzed. The accuracy and loss metrics were used to evaluate the performance of each model. Based on the results, the hyperparameters of the model were adjusted to further validate and enhance the findings. The methodology used in this research project was a systematic and rigorous approach that provided a reliable and robust way to analyze a large dataset of ECG recordings and assess the performance of different CNN models. The study findings can have important implications for the development of diagnostic tools and treatments for cardiovascular diseases.

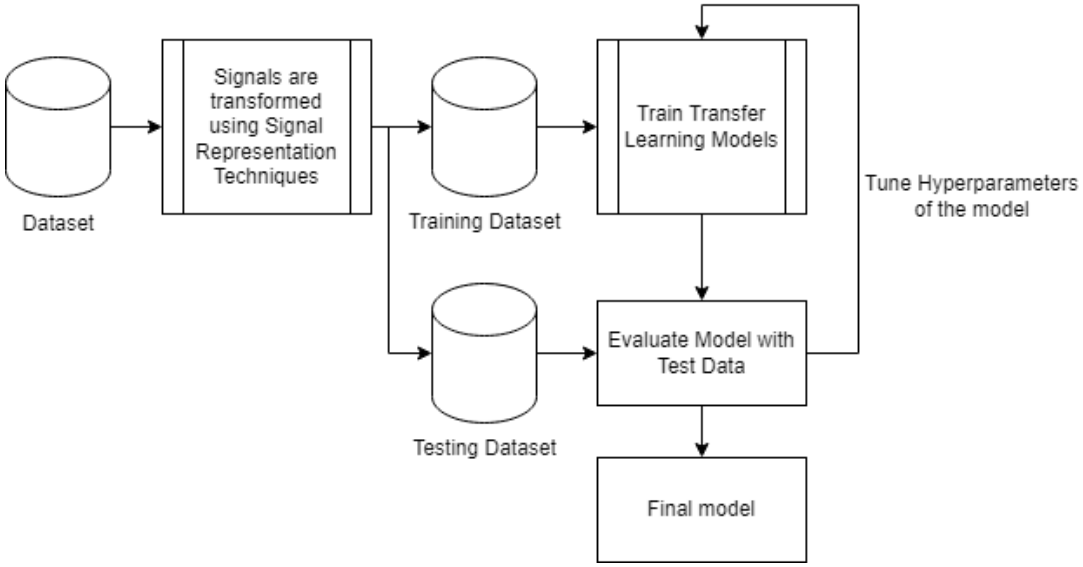


Fig. 1: Overview of Research Methodology for this project

Signal representation methods are techniques used to represent signals like time-series information such as an audio file, in a concise and meaningful manner. The goal of signal representations is to extract the most important features of a signal, which can be used for further analysis and processing. We focus on two types of signal representation, the spectrogram and MFCC, while for spectrogram we tried two variations of different hop length and number of points in the Fast Fourier Transform (FFT). An example of each of them is included in Figure 2.

A. Spectrogram & Short Time Fourier Transform

A spectrogram is a visual representation of the spectrum of a signal over time. It shows how the frequency content of a signal changes over time, by plotting the amplitude of different frequency components of the signal as a function of time. In other words, it is a 2D plot where the x-axis represents time, the y-axis represents frequency, and the color or brightness of each point represents the amplitude of the corresponding frequency component.

To compute a spectrogram, we use the short-time Fourier transform (STFT) which is a technique for analyzing non-stationary signals. The STFT works by dividing the signal into small overlapping segments, called windows, and then applying the Fourier transform to each window to obtain a time-frequency representation of the signal.

The signal is first multiplied by a window function, such as a Hamming or Hanning window, which tapers the signal at the edges to minimize spectral leakage before being used to calculate the STFT. The Fourier transform is then applied to each segment of the windowed signal after it has been separated into overlapping segments. The resulting complex values are then squared to obtain the power spectrum, which represents the distribution of power over frequency. By stacking the power spectra over time, we obtain a spectrogram of the signal.

The trade-off between temporal and frequency resolution in the spectrogram is determined by the window size and overlap selection. In contrast to larger window sizes, which offer better frequency resolution but worse temporal resolution, smaller

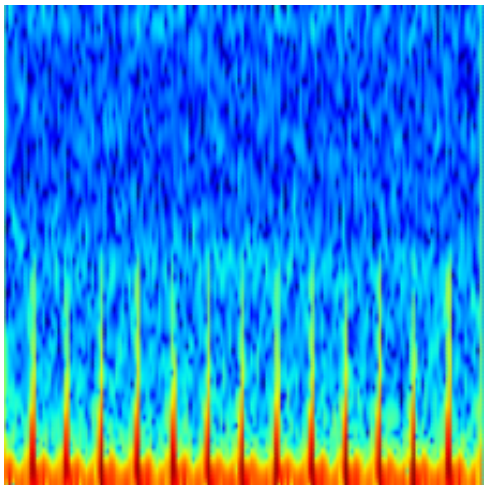
window sizes offer better temporal resolution but worse frequency resolution. The smoothing of the spectrogram is influenced by the overlap between adjacent windows, which can also affect the temporal and frequency resolution.

The formula for the short-time Fourier transform (STFT) can be written as follows:

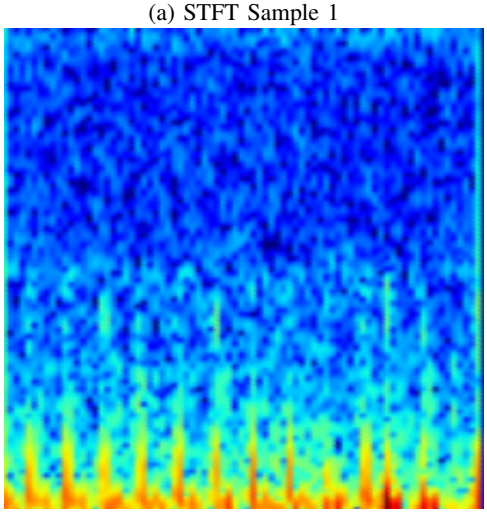
$$X(m,\omega) = \sum_{n=-\infty}^{\infty} x(n)w(n-m)e^{-j\omega n}$$

where $X(m,\omega)$ is the complex-valued STFT coefficient at time index m and frequency index ω, $x(n)$ is the input signal, $w(n)$ is the window function centered at time index m, and $e^{-j\omega n}$ is the complex exponential at frequency index ω.

Spectrograms are widely used in audio signal processing for tasks such as music analysis [1], and sound classification [2]. They are also used in other fields such as the medical field, where research has been conducted in decoding leg muscle activities of people who suffer Parkinson's disease. The spectrograms were used to identify patterns in the Local Field Potentials (LFP), which are electrical signals that are generated by the collective activity of neurons in a small region of the brain that corresponds to different aspects of walking, such as initiation, termination, and vigor of leg muscle activation. [3]



(a) STFT Sample 1



(b) STFT Sample 2

Fig. 2: Two samples of images generated from an ECG signal

III. Transfer Learning

IV. Experiments and Results - Mayo Clinic Electrocardiogram (ECG) Data

In the research, we aim to explain the experiments conducted in order to analyze the performance of different models on ECG data.

A. VGG16 Model

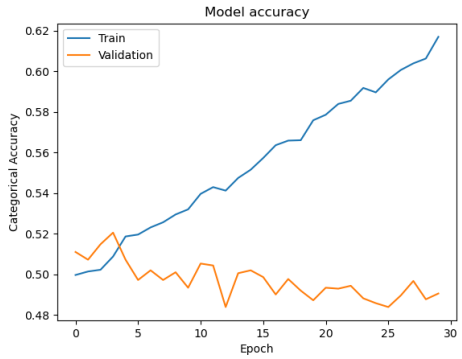
1) Experiment 1 - Initial 30 epochs: In the first experiment, we employed transfer learning using a pre-trained VGG16 model to analyze STFT images of 12-lead ECG data. The VGG16 model is a well-established deep learning architecture known for its strong performance in image classification tasks. We fine-tuned only the last few layers of the VGG16 model to adapt it to our specific dataset and problem.

The input data for this experiment consisted of Short-Time Fourier Transform (STFT) images generated from the 12-lead ECG signals. STFT is a widely-used technique for time-frequency analysis of signals, which allows us to visualize the spectral content of the ECG data. Each lead was trained on its corresponding STFT image, resulting in a total of 12 separate training processes.

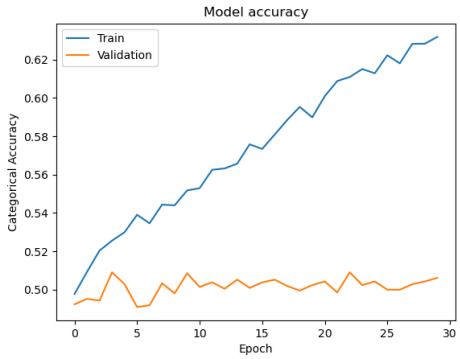
The training process was conducted over 30 epochs, with the model's performance evaluated in terms of accuracy and loss metrics for both training and validation sets. The results showed that the training accuracy increased consistently throughout the epochs, suggesting that the model was learning from the training data. However, the validation accuracy exhibited a different behavior: it experienced a slight increase during the initial epochs but then plateaued around 50%. This could indicate that the model might be overfitting the training data, as it is not able to generalize well to the validation set.

In terms of the loss metric, the training loss continued to decrease throughout the epochs, further supporting the notion that the model was learning from the training data. On the other hand, the validation loss initially decreased but eventually started to flatten, which could also be a sign of overfitting or the model reaching its capacity to learn from the given data.

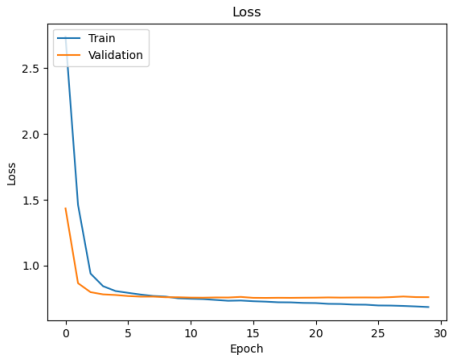
It is important to note that the training accuracy reached only about 62%, which led us to believe that the model might benefit from additional training. In order to address these issues, Experiments 2 and 3 were designed to investigate the potential benefits of training the model for more epochs.



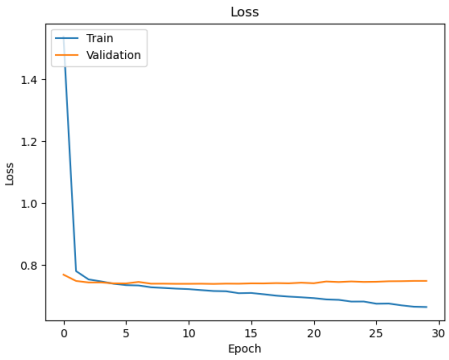
(a) Lead 1: Categorical Accuracy over 30 epochs



(b) Lead 2: Categorical Accuracy over 30 epochs



(c) Lead 1: Categorical Loss over 30 epochs

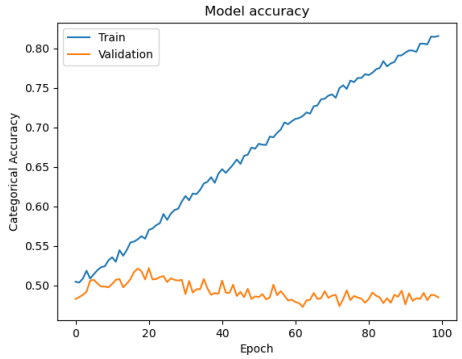


(d) Lead 2: Categorical Loss over 30 epochs

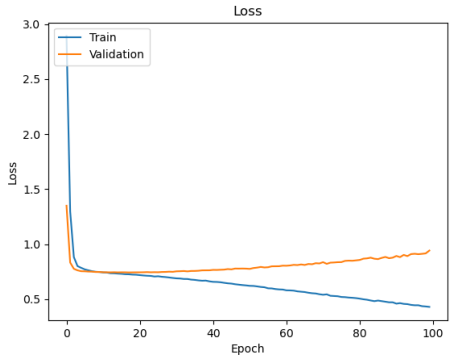
Fig. 3: Experiment 1 - Categorical Accuracy & Loss Performance for the first 2 leads.

2) Experiment 2: Increasing Epochs to 100: In Experiment 2, we aimed to address the relatively low training accuracy observed in Experiment 1 by increasing the number of training epochs from 30 to 100. This modification allowed the model to learn more from the training data, ultimately reaching a training accuracy of slightly above 80%. However, the results for the validation set remained unchanged, with the validation accuracy plateauing at around 50%.

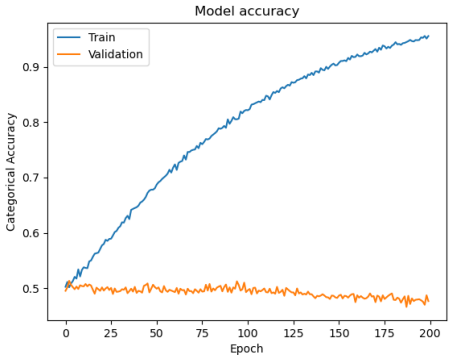
The training loss continued to decrease, indicating that the model was still learning from the training data. However, the validation loss exhibited a different trend, as it started increasing after every epoch. This behavior suggests that the model might be overfitting even more as the number of training epochs increased, leading to a decrease in its generalization capability.



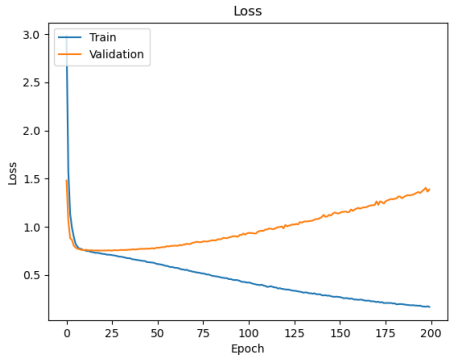
(a) Lead 1: Categorical Loss over 100 epochs



(b) Lead 2: Categorical Loss over 100 epochs



(c) Lead 1: Categorical Loss over 200 epochs



(d) Lead 2: Categorical Loss over 200 epochs

Fig. 4: Experiment 2 & 3 - Categorical Accuracy & Loss Performance for the first 2 leads .

3) Experiment 3: Increasing Epochs to 200: In Experiment 3, we sought to further explore the effect of increasing the number of training epochs by extending the training period from 100 to 200 epochs. This change led to a substantial improvement in the training accuracy, which reached above 90%. The training loss also continued to decrease, reinforcing the idea that the model was still learning from the training data.

However, the results for the validation set were not as promising. The validation accuracy remained stagnant at around 50%, and the validation loss continued to increase. These trends indicate that the model was becoming increasingly overfit to the training data as the number of epochs increased, making it less effective at generalizing to new, unseen data.

In conclusion, Experiments 2 and 3 demonstrated that increasing the number of training epochs can lead to improvements in training accuracy but does not necessarily translate to better performance on the validation set.

4) Experiment 4: Stitching the 12-lead ECG images into one single image: In Experiment 4, we aimed to investigate the impact of combining the STFT images generated for each lead into a single image. This approach allows us to train just one model for all leads, as opposed to the separate models for each lead used in the previous experiments. The stitched image contains information from all 12 leads, which could potentially improve the model's ability to capture relevant patterns in the ECG data.

Due to the increased size of the resulting images and the memory limitations of the machine used for training, we had to decrease the batch size during the training process. This change might affect the training dynamics and the overall performance of the model.

The results of Experiment 4 show that the model achieves a high level of training accuracy in a lower number of epochs compared to the previous experiments. This finding suggests that the model is capable of learning more efficiently when provided with combined information from all leads in a single image. However, the validation accuracy remains stagnant at around 50%, indicating that the model still struggles to generalize to unseen data.

In terms of the loss metrics, the training loss drops quickly, which is consistent with the rapid increase in training accuracy. On the other hand, the validation loss increases quickly, further reinforcing the notion that the model is overfitting the training data.

In conclusion, Experiment 4 provided valuable insights into the potential benefits of training a single model on combined STFT images from all 12 leads. While this approach led to faster learning and higher training accuracy, the persistent issue of overfitting and stagnant validation accuracy suggests that additional strategies are needed to improve the model's generalization

capabilities. Future experiments could explore the use of regularization techniques, data augmentation, or alternative model architectures, as well as potentially combining these approaches to address the overfitting issue more effectively.

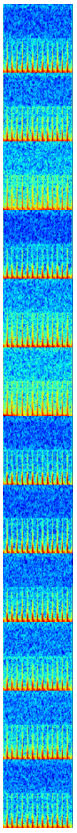


Fig. 5: A sample of 12-lead ECG stitched image

5) Experiment 5: Ensemble Model of 12 Separate Lead Models: In Experiment 5, we investigated the potential benefits of creating an ensemble model using the 12 models that were trained on each separate lead. Ensemble methods aim to combine multiple weak classifiers to build a stronger classifier, which can often lead to improved overall performance. The rationale behind this approach was to capitalize on the strengths of each model and potentially mitigate the limitations of individual models.

However, the results of Experiment 5 did not meet our expectations. Due to the weak classifiers' inconsistent categorization of positive samples as positive, the ensemble model consistently anticipated negative results. This tendency led to a model that was strongly biased towards making bad predictions, which is undesirable and restricts the model's application in a clinical setting. These results can be summarized in the confusion matrix that is represented in Table I. It is seen that the model classifies every sample in the test set as negative (non-obstruction) but also classifies all positive (obstruction) samples as the negative class as well.

Finally, Experiment 5 showed that the ensemble approach based on 12 distinct lead models did not result in the desired performance improvements. A model with a substantial bias towards negative predictions was created as a result of the poor classifiers' lack of agreement when recognizing positive data. Alternative ensemble methods, such as weighted voting or stacking, as well as more complex methods for combining the data from each lead could be investigated in future work in order to increase the model's accuracy in identifying positive samples.

		Actual	
		Negative	Positive
Predicted	Negative	7,341 (TN)	0 (FN)
	Positive	2,164 (FP)	0 (TP)

TABLE I: Confusion Matrix of the Ensemble Model

Given the overfitting we saw in our trials with the VGG16 model, we have chosen to test the efficacy of more straightforward CNN models for the classification of ECG data. Our aim is to investigate if the VGG16 architecture may be overly complex for the current challenge, and whether less complex models, such as LeNet-5 and AlexNet, may be able to reduce the overfitting problem. These less complex models have been useful in a number of image classification tasks, and they could offer important

information on how well other CNN designs perform when analyzing ECG data. In the sections that follow, we'll go over the experiment done using AlexNet and LeNet-5 models and evaluate how they performed in comparison to VGG16-based methods, paying particular attention to how well they generalized to new data.

B. AlexNet

AlexNet has a deeper architecture compared to LeNet-5 but shallower than VGG16. It consists of 8 layers: 5 convolutional layers and 3 fully connected layers. The convolutional layers use varying filter sizes. Max-pooling layers are added after some of the convolutional layers to reduce spatial dimensions. The network finishes with three fully connected layers and a final softmax layer for classification. AlexNet's complexity lies between LeNet-5 and VGG16. It has more layers and parameters than LeNet-5, resulting in better performance on complex tasks. It is less complex than VGG16, which makes it computationally more efficient and faster to train than VGG16.

C. LeNet-5

LeNet-5 has a simpler and shallower architecture compared to the other models. It consists of 7 layers in total, 2 convolutional layers, 2 average pooling layers, and 3 fully connected dense layers. The convolutional layers use 4x4 and 5x5 filter size for the respective 2 convolutional layers, and the subsampling layers are used to reduce spatial dimensions. The network ends with a final fully connected layer for classification. LeNet-5 is the least complex of the three models due to its fewer layers and smaller size. This results in fewer parameters, making it computationally more efficient and faster to train. However, its simplicity may limit its performance on more complex tasks.

In this experiment, we trained a LeNet-5 model for each of the 12 leads, aiming to evaluate the performance of a simpler CNN architecture in comparison to the VGG16 model. The results obtained from the LeNet-5 models exhibited similar trends to those observed with VGG16. Specifically, we observed continuous improvement in training accuracy and decreasing training loss, indicating that the models were effectively learning from the training data. However, the validation loss worsened, and the validation accuracy remained stagnant at around 50%, suggesting that the models still struggled to generalize to unseen data.

It is worth noting that the LeNet-5 models trained faster and reached a higher level of training accuracy in fewer epochs compared to the VGG16 models. This observation demonstrates the potential benefits of using a simpler architecture, such as reduced computational requirements and faster training times. However, the the issue of overfitting persists even with the simplest of the three CNN models. Further investigation and modifications are needed to improve the models' generalization capabilities by improving the STFT images quality. The results on the model performance for the first 2 leads is summarized in Figure 6.

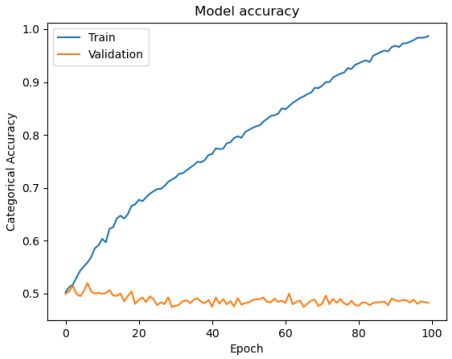
D. Sequential Convolutional Neural Network Model

The convolutional neural network (CNN) architecture presented above is a sequential model designed for processing raw two-dimensional 12-lead ECG data, as opposed to transforming the ECG data into STFT images and training on those images. This experimental approach aims to directly leverage the inherent spatiotemporal information within the ECG signals. The model begins with a reshaping layer that transforms the input into the desired shape, specifically a 12x5000 matrix with a single channel, where each row represents a lead and each column represents a time point.

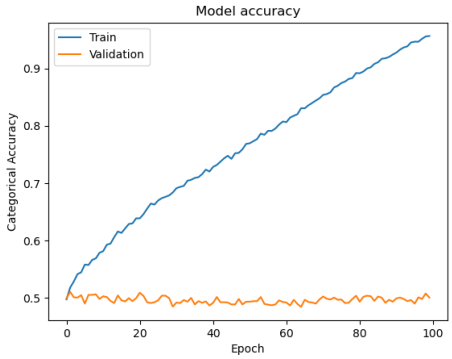
Next, the first convolutional layer is applied with 32 filters, a kernel size of 1x5, activated by the ReLU activation function, and with valid padding. This is followed by a max-pooling layer with a pool size of 1x50 and valid padding, reducing the spatial dimensions of the feature maps. The model then has another convolutional layer with 64 filters, a kernel size of 3x3, ReLU activation, and valid padding. This is paired with a max-pooling layer using a pool size of 2x2 to further reduce the spatial dimensions.

The output is then flattened, converting the two-dimensional feature maps back into a one-dimensional array. The fully connected part of the architecture consists of two dense layers, with 128 and 64 units respectively, and ReLU activation functions. Dropout and batch normalization layers are interspersed between these dense layers to prevent overfitting and improve training stability. Finally, the output layer comprises a dense layer with a number of units equal to the number of classes and a softmax activation function, which yields class probabilities for the given input. By working with raw ECG data, this CNN architecture attempts to capture the underlying features and patterns directly from the signals, potentially leading to more accurate and interpretable results.

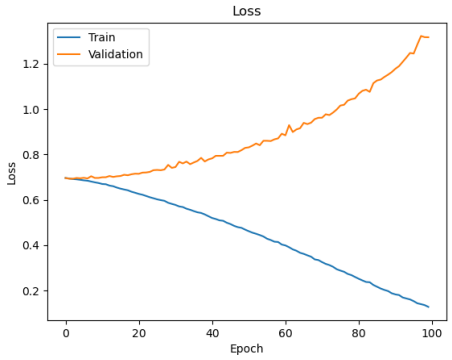
The described model was trained for a total of 200 epochs to investigate its performance on both the training and validation datasets. The results, as illustrated by the categorical accuracy and loss plots in Figure 7, demonstrate promising outcomes. Specifically, the model achieves close to 70% accuracy on the training set, while the validation set exhibits an accuracy of approximately 67.5%. In terms of loss, both training and validation losses display a decreasing trend, eventually plateauing as the number of epochs increases. However, it is noteworthy that the lines representing these metrics are rather jagged, indicating that the model's learning progression is not entirely smooth. Despite this, the overall performance of the model is encouraging,



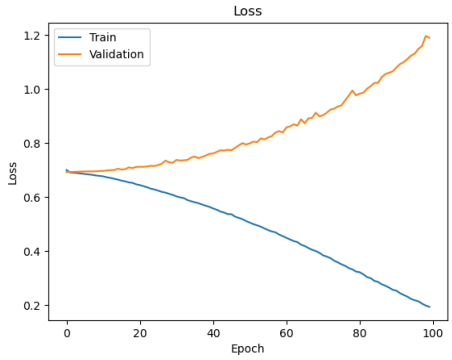
(a) Lead 1: Categorical Accuracy over 100 epochs



(b) Lead 2: Categorical Accuracy over 100 epochs



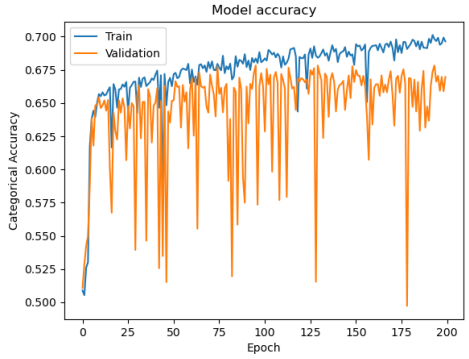
(c) Lead 1: Categorical Loss over 100 epochs



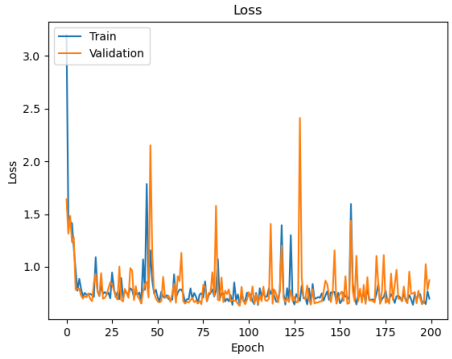
(d) Lead 2: Categorical Loss over 100 epochs

Fig. 6: Experiment 1 - LeNet-5 Categorical Accuracy & Loss Performance for the first 2 leads.

as it demonstrates a reasonable level of generalization, with the validation accuracy closely following the training accuracy. The jagged nature of the learning curves could potentially be addressed through further optimization, such as employing learning rate schedules or advanced optimization algorithms to improve convergence and model stability.



(a) Categorical Accuracy over 200 epochs



(b) Categorical Loss over 200 epochs

Fig. 7: Experiment: Training sequential convolutional neural network on raw 12-lead ECG data.

V. Future Work

By addressing a number of crucial issues, we want to improve the performance of our models and broaden the area of our research in our upcoming work. Firstly, the upper half of the STFT images, which offers little insight into the ECG signal, will be removed first in order to improve the images. With this adjustment, the model will be able to concentrate on the important signal information and avoid picking up noise from the images' less useful regions. This is shown in Figure 8, where we can see the top half of the image containing information that is not useful or provides any insights to the signal for the model to

train on. Second, we want to investigate several approaches to working with unprocessed ECG signals, such as running the data through different recurrent neural network (RNN) models, such as simple RNNs, Long Short-Term Memory (LSTM) networks, and Gated Recurrent Unit (GRU) models. These methods might make it easier for the model to efficiently recognize temporal patterns in the ECG data. Last but not least, we want to expand the scope of our research to include other classifications offered by the Mayo Clinic, such as patients falling under the ”Normal complete” or ”Normal spirometry” categories. We will be able to gain a deeper understanding of the ECG data thanks to this expansion, which may also improve the clinical usefulness of our models. We intend to address these issues in order to get over the constraints found in the current studies and offer more insightful information about the processing and classification of ECG data.

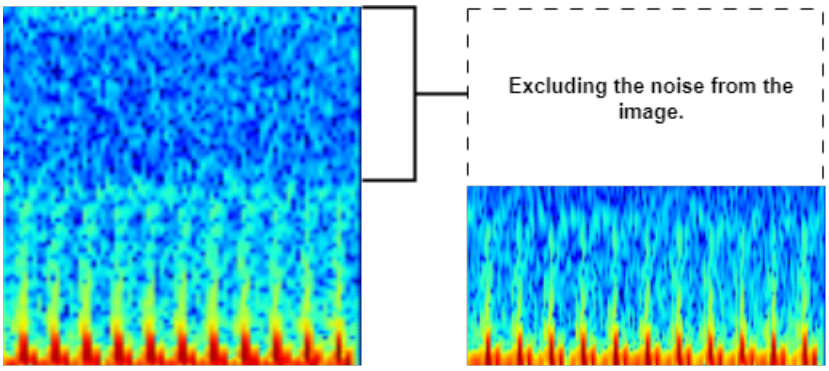


Fig. 8: Refining STFT images to exclude noisy upper half.

VI. Conclusion

In conclusion, we evaluated multiple methods for classifying ECG data during our studies utilizing various CNN architectures, including VGG16, LeNet-5, and ensemble models. All the models struggled to generalize to unseen data despite obtaining high training accuracies and rapid improvements in training loss, as seen by stagnant validation accuracies and deteriorating validation losses. The persistent problem of overfitting emphasizes the need for additional research and refinement of our models. Future work will focus on improving input data and broadening the scope of our research as we investigate ways for reducing overfitting, such as data augmentation, regularization techniques, and alternative model architectures. By addressing some of these factors, we seek to enhance the functionality of our models and offer more insightful information on the processing and categorization of ECG data in relation to patients that suffer from obstruction, hence increasing their clinical relevance and possible influence on patient treatment and diagnosis.

References

[1] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in *Proceedings of the 14th python in science conference*, vol. 8, 2015, pp. 18–25.

[2] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, “Deep learning for audio signal processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[3] Y. Thenaisie, K. Lee, C. Moerman, S. Scafa, A. Gálvez, E. Pirondini, M. Burri, J. Ravier, A. Puiatti, E. Accolla *et al.*, “Principles of gait encoding in the subthalamic nucleus of people with parkinson’s disease,” *Science Translational Medicine*, vol. 14, no. 661, p. eabo1800, 2022.