Generative Adversarial Networks for PCG Arrhythmia Detection

Aditya Kendre
12th Grade
Year 5

Cumberland Valley HS
Mechanicsburg, PA, USA

Introduction

Background

Around **33% of all deaths** are the result of Cardiovascular diseases, these diseases often cause **arrhythmia**[1]. Traditional arrhythmia diagnosis requires Electrocardiogram[2] (**ECG**) analysis, this **limits the use case** to hospitals and clinics with **specialized equipment.** Hence, **13.1%** of patients have **undiagnosed Atrial Fibrillation**[3] (**AF**). Heart sounds or Phonocardiograms[4] (**PCGs**) provide a distinct **advantage** over traditional ECGs, in that it records the acoustic properties of the heart's movement. This allows for greater **versatility** and **ease of use**.

Hypothesis

If a novel heart sound analysis system is developed to detect a variety of arrhythmias, then arrhythmias will have a decreased undiagnosed rate.

Engineering Goals

1. **Increase the Number of Cardiovascular Pathologies analyzed in heart sounds**- Develop a model to construct heart sounds from pre-existing datasets that cover the complete range of pathologies that are likely to be encountered in clinical settings.
2. **Develop an End-to-End System** - Create a system that is able to record, analyze, and predict (end-to-end system) heart sounds for Cardiovascular modalities without specialized equipment.
3. **Real-World Testing** - Test the end-to-end system in a real-world environment to ensure the practicality and generality of the system.

Constraints

Constraints include the use of a small dataset and noisy recordings of heart sounds auscultated[5] at different locations.

Related Works

Although heart sound databases do exist, these **datasets are limited by the number of pathologies** that are collected, often having to divide the dataset into two categories: normal and abnormal. Currently, only three major supervised PCG datasets exist PhysioNet Classification of Heart Sound Challenge dataset, PASCAL Heart Sound Challenge dataset, and the Littman Heart Sound and Murmur Library. Additionally, **little effort has been done to increase the development and labeling of comprehensive datasets of PCG** signals that cover the complete range of pathologies.

In diagnosing heart sounds, two major challenges arise: localization and classification. Localization aims to find the position of biomarkers in heart sounds. By doing this, heart sounds can be segmented into signals containing a single heart sound. Furthermore, classification attempts to categorize heart sounds into normal and abnormal groups by exploiting the information extracted from localization. Conventional heart sound **localization and classification methods involve time and/or frequency and** are typically **dependent on machine learning algorithms to enhance the results**. These algorithms typically include artificial neural networks (ANNs), support vector machines (SVMs), self-organizing maps (SOMs), and are limited to the number of samples and pathologies covered in a given dataset. **This leads to a surface-level analysis of the heart sounds.**

[1] Irregular heartbeat
[2] Recording of the electrical signals produced by the heart
[3] A type of arrhythmia that produces an irregularly irregular heartbeat
[4] Recordings of the sounds created by the physical movement of the heart
[5] Listening to sounds created from the heart

Methods

Data Management

Although heart sounds are analyzed more often than ECG recordings, a better variety of ECG datasets exist. Thus, we proposed using both ECG and heart sound (PCG) datasets for arryhthmia classification. PCG recordings often are recording in non-ideal environments that are filled with unwanted background noise and interference. Hence, we preprocess the data by denoising, normalizing, standardizing, and transforming the signal. This allows a model to extract meaningful features efficiently, revealing the physiological structure of the heart sounds. The data is then augmented to enable a significant increase in the diversity of data available while training a model, this is done without collecting new data. It aims to slightly alter existing data to a point where the model cannot recognize the augmented data as one it has trained on before, but still maintains important distinguishing characteristics.

Dataset	Dataset Type	Lengths	Environment & Recording Quality	Pathologies Ratios
Classification of Heart Sound Recordings - PhysioNet 2016	PCG & ECG	5-120 seconds	Extremely noisy and low signal quality	**Normal**: 3541 (77.1%) **Abnormal**: 551 (12.0%) **Noisy**: 501 (10.9 %)
PASCAL 2011	PCG	1-30 seconds	Noisy and taken from iStethoscope and digital stethoscopes	**Normal**: 351 (40.0%) **Murmur**: 129 (14.7%) **Extra**: 65 (7.4%) **Artifact**: 86 (9.8%) **Unlabeled**: 247 (28.1%)
Littman Heart Sound & Murmur Library	PCG	2 seconds	Clean and taken from digital stethoscope	**Stenosis**: 5 (31.3%) **Septal**: 1 (6.3%) **Ejection**: 3 (18.8%) **Coarctation**: 1 (6.3%) **Prosthetic**: 1 (6.3%) **Regurgitation**: 2 (12.5%) **Pericarditis**: 1 (6.3%) **Gallop**: 2 (12.5%)
PTB-XL	ECG	10 seconds	n/a	**Normal**: 9528 (34.2%) **CD**: 5486 (19.7%) **MI**: 5250 (18.9%) **HYP**: 4907 (17.6%) **ST/TC**: 2655 (9.5%)

Table 1: Dataset table that shows the dataset type (PCG or ECG), the lengths of the recordings (in seconds), the environment in which the signals were recorded, the recording quality and the number of categories in the dataset (Table taken by Kendre, 2021).

Cross-validation

Cross-validation is used to estimate how accurately a predictive model will perform in the real-world. In a prediction problem, a model is usually given a dataset of known data on which training is done (training set). The training process optimizes the model parameters via backpropagation to make the model fit the training set as well as possible. Additionally, we need to assess the best place to stop training to ensure the model does not overfit. Thus, we need to test the model on an independent set (validation set) to validate the model's success. However, this creates a bias towards the samples in the training and validation set. Hence, we need a third dataset (testing set) that the model has no bias to evaluate the model. Here, the data was split into 70% of training, 10% of validation, and 20% of testing.

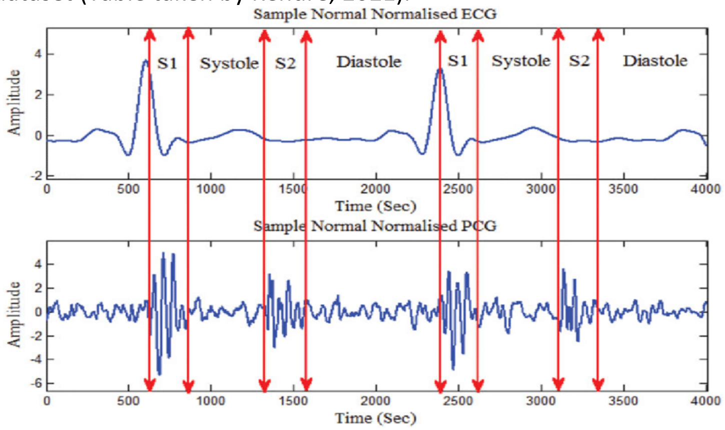


Figure 1: Visualizes the ECG and PCG signals recorded simultaneously, both PCG (bottom) and ECG (top) show evidence of systole and diastole (Figure taken by Behbahani et al).

Methods

Model Development

We propose using Generative Adversarial Networks (GANs) along with Convolutional Neural Network (CNNs) and Transformers[1] for both heart sound analysis and heart sound synthesis based on ECGs. GANs pose a unique advantage over traditional machine learning and deep learning methods, in that the model learns to mimic a dataset by creating its own data while classifying the generated data and the data from the dataset. Furthermore, Transformers introduce attention mechanisms that give context to all inputs, allowing the model to enhance important non-linear relationships within in each signal.

In heart sound analysis, the PCG CNN Encoder is pre-trained[2], to compress the PCG signals into a latent space[3], and then reconstructed into PCGs using the PCG CNN Decoder. After pre-training, the PCG signal's latent spaces are fed into the GAN discriminator (Transformer Encoder). Here, the Transformer creates a prediction based on latent space. Additionally, the PCG Generator tries to mimic PCGs to fool the Transformer into predicting the generated PCG as normal/abnormal, rather than as noisy.

In heart sound synthesis, the process of ECG reconstruction is identical to that of PCG reconstruction. However, during pre-training the ECG's latent spaces are optimized towards the PCG's latent space. This novel concept forces the Generators to produce equivalent latent spaces for ECG and PCG (given both signals were recorded simultaneously). After pre-training, the ECG CNN Encoder is fed ECG data from categorized arrhythmia datasets.

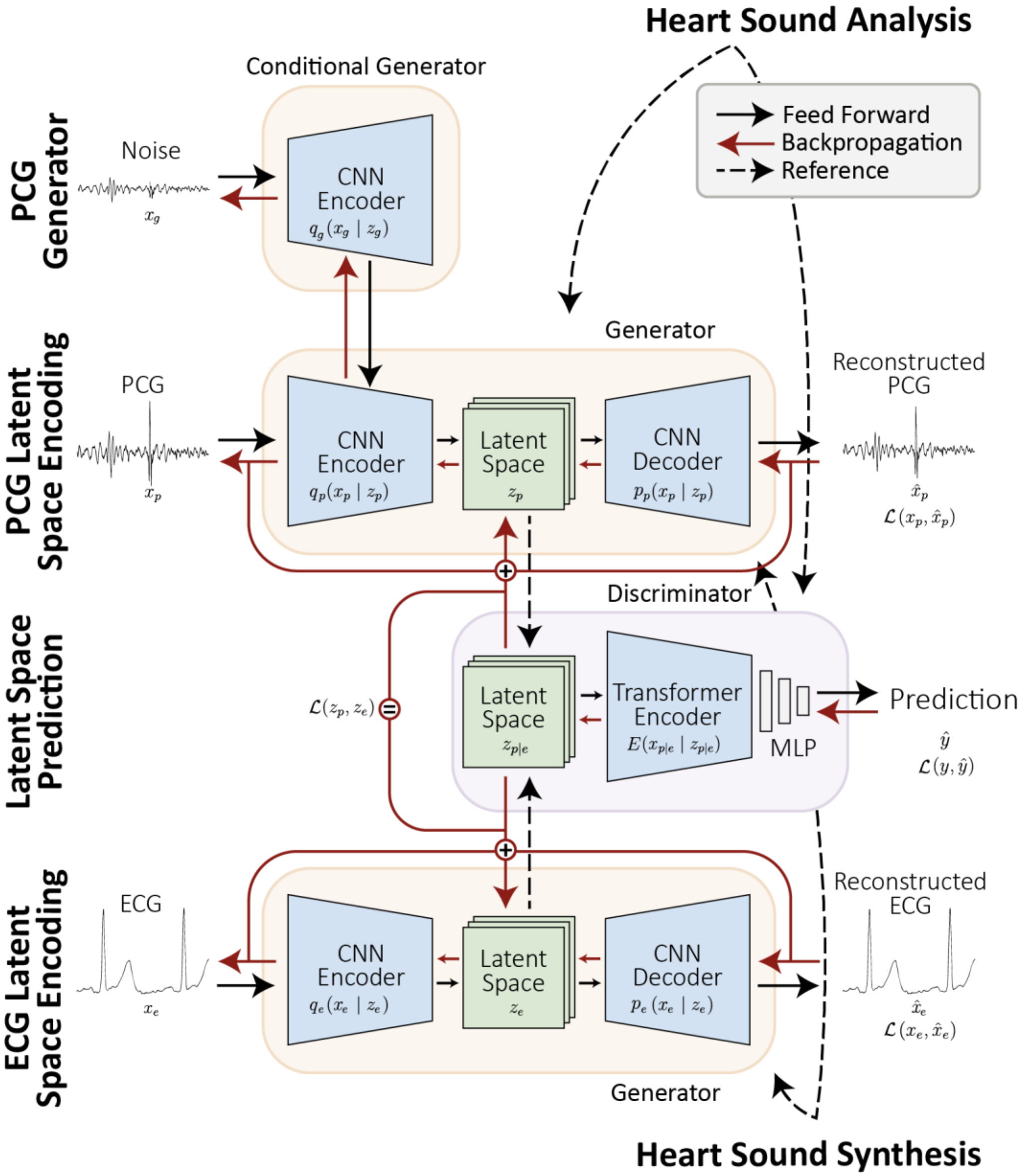


Figure 2: The pipeline of the CNN based generator and Transformer based discriminator. This shows the process of using both PCG and ECG datasets for heart sound (PCG) analysis and synthesis (Figure taken by Kendre, 2021).

1. Machine Learning architectures used for feature extraction
2. Initial training that is done to aid the model better performance in a shorter amount of time and computational resources
3. A representation of compressed signals

Methods

Model Learning

During the training phase, backpropagation is used to optimize the model's weights and biases. A cost function calculates an error term (loss) based on the model's prediction. Using the error, a gradient is computed with respect to all the parameters in the model. An optimizer then updates the weights and biases based on the loss's gradient. The goal of the optimizer is to minimize the cost function's error, such that it correctly classifies the heart sounds or correctly synthesizes the heart sounds. In this study, we used the Adam optimizer in union with Cross-Entropy Loss and Mean Squared Error (Table 2). The model is only trained on the training set; thus, backpropagation only occurs on the training set. However, while the model is training, the validation set is used to assess the model's ability to generalize (this process does not include backpropagation). The model stops learning when the validation set approaches a limit and no longer learns new information, this is called early stopping.

Model Evaluation

Model metrics aid in quantifying the model’s performance, allowing us to compare our methods with existing techniques. The model is evaluated only on the testing set, as to provide a non-biased evaluation. All metrics are calculated based on the true positive (TP) rate, false positive (FP) rate, true negative (TN) rate, and false negative (FN) rate. Table 2 shows the metrics used to evaluate the model.

Real-World Testing & Model Deployment

The model's viability is crucial for ensuring the model's success in the real world. Thus, we need to conduct trials with the end-to-end system to ensure we can deploy and integrate the model with ease.

Mean Squared Error:

$$MSE(x, y) = \{l_1, \ldots, l_N\}^\top, l_n = (x_n - y_n)^2$$

Cross Entropy Loss:

$$\mathrm{CE}(x, class) = -\log(\frac{\exp(x[\ class\ ])}{\sum_j \exp(x[j])}) = -x[\ class\ ] + \log(\sum_j \exp(x[j]))$$

Accuracy:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

Sensitivity:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR}$$

Specificity:

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} = 1 - \mathrm{FPR}$$

Positive Predictive Value:

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = 1 - \mathrm{FDR}$$

Negative Predictive Value:

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} = 1 - \mathrm{FOR}$$

F1 Score:

$$\mathrm{F}_1 = 2 \times \frac{\mathrm{PPV} \times \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

Time Complexity:

$$\mathrm{O(n)} = \frac{1}{n}\sum_{i=0}^{n} m(|input| \sim n)_t$$

Table 2: Metrics shows in the table are used to evaluate the model on the testing set to ensure generalization and comparability (Table taken by Kendre, 2021).

Results

Distribution Table

The proposed method introduces significant features and architectures that aid in abnormality detection. Using such techniques, the classification metrics showed extremely promising results as shown in Table 3. Specifically, the model achieved 73.7% accuracy on the PASCAL dataset and 89.5% accuracy on the PhysioNet dataset. Hence, the model proved to be very efficient for classification of normal and murmur-filled heart sounds. Furthermore, we also propose creating heart sounds from ECGs for additional arrhythmia-specific training. Training the model on the synthesized heart sounds provided a mean accuracy of 93.0%, ECG to PCG conversion is a feasible approach. Additionally, training the model on all 3 datasets indicated excellent results, predicting 95.0% of abnormalities correctly.

Model Metrics on Datasets

Dataset	Accuracy	F1 Score	Sensitivity	Specificity	PPV	NPV
PASCAL	73.7±.02	75.0±.04	75.0±.05	72.2±.04	75.0±.02	72.2±.04
PhysioNet 2016	89.5±.03	64.4±.05	48.0±.06	85.7±.02	96.5±.04	85.7±.03
Synthesized PTB-XL (AF)	93.0±.03	94.3±.03	99.5±.03	90.3±.03	99.3±.03	91.8±.03
Combined	**95.0±.03**	**94.6±.04**	**94.3±.04**	**99.5±.03**	**90.3±.03**	**99.3±.02**

Table 3: The table displays the metrics collected on the testing set for all classes in the dataset (Table taken by Kendre, 2021).

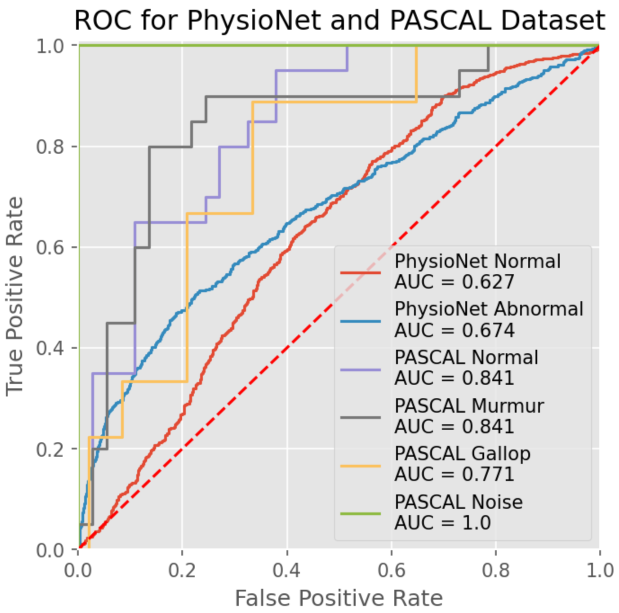


Figure 3: The ROC curve shows the TPR and FPR for different thresholds for the model's prediction and the AUC for each class, which shows the model's efficiency (Figure taken by Kendre, 2021).

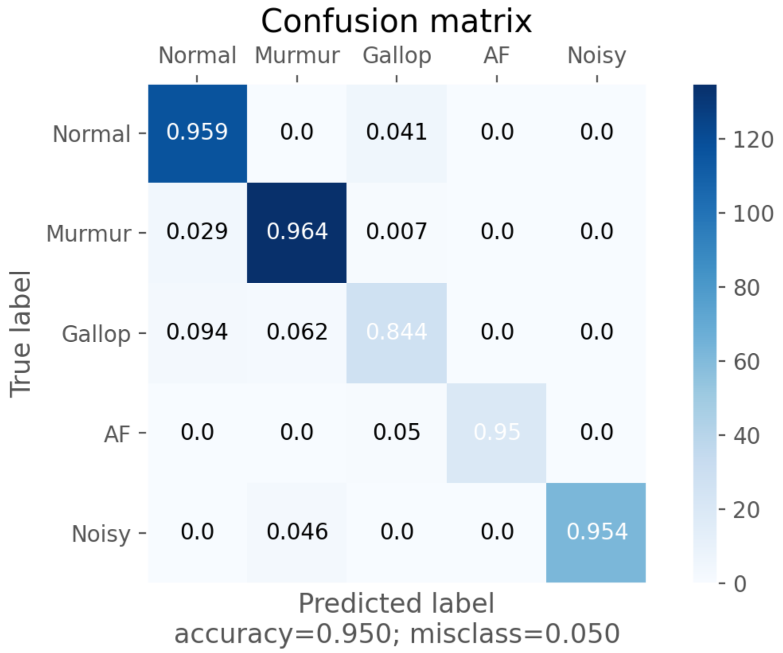
ROC/AUC

The receiver operating characteristic curve (ROC) aids in understanding the model's ability to predict the correct heart sound class. The curve plots the true positive rate against the false positive rate at various prediction thresholds. Based on the ROC, we choose the optimal threshold for all classes. However, the optimal threshold depends on a subjective trade-off between the true and false positive rates. Here, we choose to optimize for increased true positive rate, as we want to ensure all potential subjects with a disease are sent for further examination. Additionally, the graph illustrates the area under the curve (AUC), this analysis provides an aggregate measure of the model's performance over all classification thresholds. All AUC scores are above the 0.5 threshold; this suggests that the model's ability to distinguish between classes is high.

Results

Confusion Matrix

The confusion matrix in Figure 4 illustrates the performance for each class of the proposed method. Specifically, we evaluated the model's success on the grounds of accuracy, specificity, and sensitivity of the classification. We calculated the average true positives, false positives, false negatives and false positives for all testing sets. Using these floored values, we normalized each label along the y-axis. The matrix reveals that the most common misunderstanding occurs between Gallops and Normal rhythms. This is expected as Gallops are heart sounds that contain an extra sound like S3 or S4, which are often lacking in amplitude. Overall, we conclude that the average accuracy of abnormal heartbeat detection is ~95% with a misclassification rate of just ~5%. Thus, the model is extremely accurate in detecting abnormalities in heart sounds and displays the capability to classify abnormal heart sounds into arrhythmia and abnormality types.



Confusion matrix

True label \ Predicted label	Normal	Murmur	Gallop	AF	Noisy
Normal	0.959	0.0	0.041	0.0	0.0
Murmur	0.029	0.964	0.007	0.0	0.0
Gallop	0.094	0.062	0.844	0.0	0.0
AF	0.0	0.0	0.05	0.95	0.0
Noisy	0.0	0.046	0.0	0.0	0.954

accuracy=0.950; misclass=0.050

Figure 4: Matrix of accuracy between categories (pathologies) classified by the model (predicted label) and the true categories (true label) (Figure taken by Kendre, 2021).

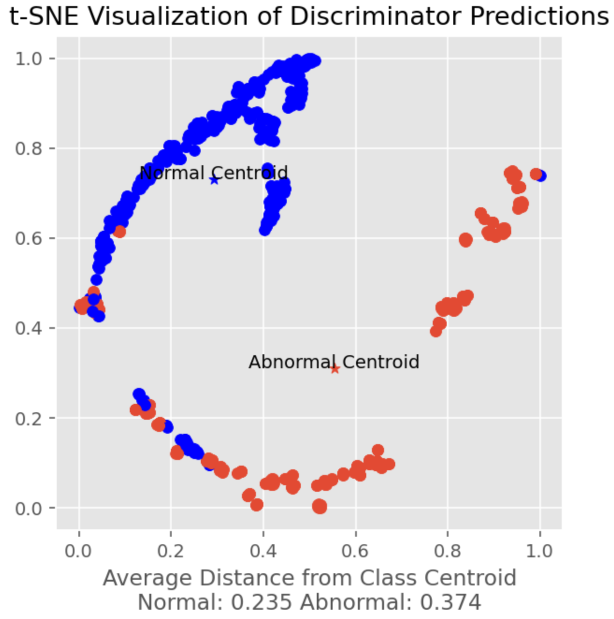


Figure 5: t-SNE visualization of testing dataset after model training (Figure taken by Kendre, 2021).

t-SNE Visualization

Dataset visualization is critical in understanding the dataset's complexity and model's effectiveness. Here, we use t-distributed stochastic neighbor embedding (t-SNE), a statistical method for visualizing multidimensional data with less computational expense. The method is presented with the raw prediction values for each input of the validation set and maps the corresponding predictions into a 2-dimensional space (x, y). Tracked over the best epoch, the visualization allows us to view the differentiation between heart sounds from the model's prediction. The visualization highlights clear clustering within the dataset, which suggests the model is stable. Though, it is evident that there is overlapping between abnormal and normal signals in the y=0.4-0.6 range. Assuming these signals as ground truth, this implies that additional feature engineering is required to adequately classify heart sounds.

Results

Time Complexity

Model complexity is used to gauge and evaluate the efficacy of a model against an increase in data (n). We mainly focus on the discriminator's time complexity as it is most relevant to the problem at hand (space complexity is O(1)). Depending on model deployment and integration, the complexity can vary. For example, GPUs have parallel processing capabilities, which allow them to process multiple signals at once, effectively decreasing the discriminator complexity to O(1). For this reason, we use the worst-case scenario (a CPU), for analysis of the proposed method's time complexity. The discriminator's time complexity is directly and linearly correlated to the input size, suggesting the complexity is O(n). This means that the discriminator's efficiency enables real-time heart sound analysis.

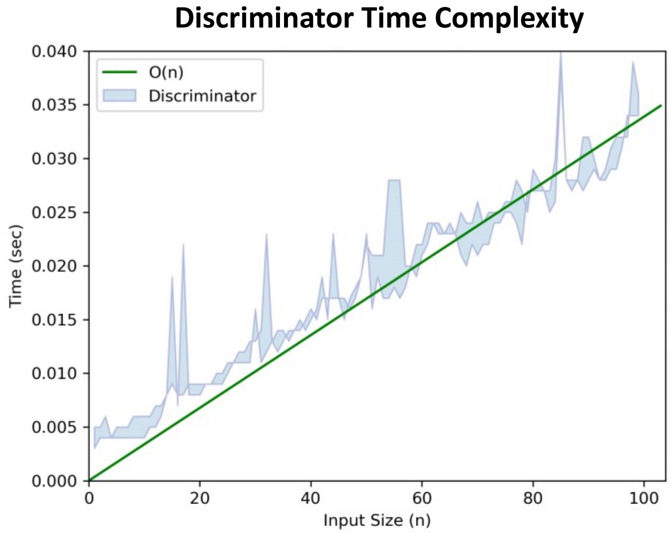


Figure 6: Time complexity of the PCG/ECG Encoder and Transformer Discriminator in classifying heart sounds (Figure taken by Kendre, 2021).

End-to-End System

Testing the model's viability is crucial for ensuring the model's success in the real world. Ideally, recording heart sounds are recorded with digital stethoscopes. These tools use transducer technology to convert sound into an electrical signal. Over the past decade, this technology has grown immensely (by cause of speech recognition). Modern phones have the potential to record the sounds at a high resolution, given the microphone is located at the correct position relative to the heart. Such a device will prove extremely beneficial in providing a diagnosis without the need for specialized equipment. Figure 7 shows a heart sound recording from the phone microphone. The plot shows important biomarkers like S1 and S2, which suggest the microphone suitable for the classification task. This confirms smartphone microphones do not record excessive amounts of noise that may hinder the performance of the detection system. The smartphone app not only allows anyone to record their heart sounds with any device with a microphone but does so without needing any external equipment; such as a case or a stethoscope. Additionally, the app allows the users to download and email the sound recordings, providing physicians and cardiologists with deeper insight. This approach allows the app to also track heart activity over time, allowing for more awareness of your fitness level, heart health, and emotional health.

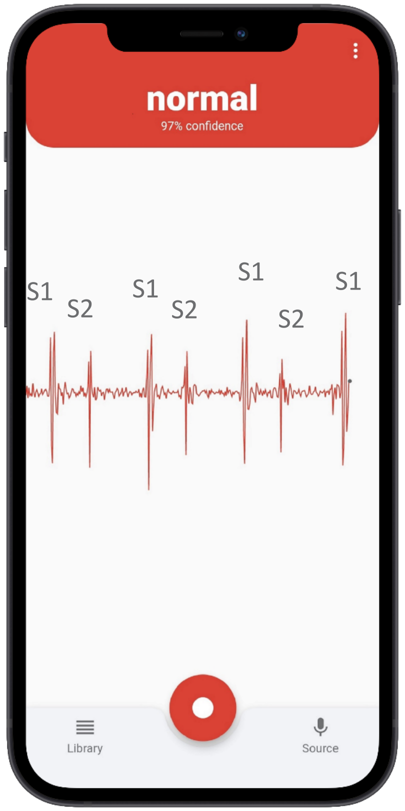


Figure 7: Screenshot of created app recording a heart sound from the built-in microphone (Figure taken by Kendre, 2021).

Discussion

Model Interpretation

Figure 8 visualizes the channels of the Convolutional Encoder layers which are responsible for extracting features from heart sounds. The color of the line represents the importance of the feature relative to other features. Meaning, the lighter the color, the more important the feature. The map illustrates that the model pays more attention to peaks of higher amplitude in the heart sound. This indicates that the layers are extracting latent features from the signal. Specifically, the plot depicts the extraction of important biomarkers such as S1 and S2. Furthermore, the extractions make it clear that the S1 sound is more important, as those are the brightest throughout all layers. This parallels medical knowledge, as most cardiovascular anomalies occur in Systole, or at the start of S1.

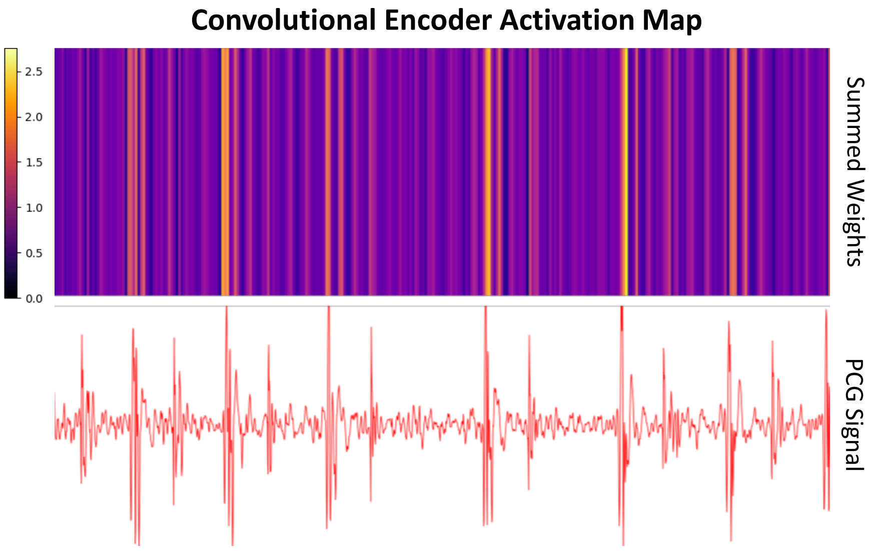


Figure 8: Visualization of CNN encoder when fed a heart sound (Figure taken by Kendre, 2021).

Comparative System Evaluation

In a clinical setting, physicians often use digital stethoscopes for listening to heart sounds. By monitoring PCG characteristics such as amplitude, pitch, and cyclic patterns, physicians differentiate between abnormalities in the heart sound. Additionally, these devices allow their users to significantly amplify heart sounds by eliminating ambient noises by filtering background noise and amplifying the sounds recorded by the sensor. However, the average heartbeat is between 60 and 100 bpm, meaning a heart sound can occur anywhere from once a second to twice a second. Hence, digital stethoscopes require high sampling rates for collecting heart sounds. Table 4 conveys most digital stethoscopes have a sampling rate of 4000-8000 Hz; however, modern phones have sampling rates of 16000-48000. This high sampling rate provides more accurate and granular control over traditional digital stethoscopes.

Digital Stethoscope Comparison

Device	Frequency Range (Hz)	Sample Rate (Hz)	Amplification	Cardiac Landmark Guide	Cost
3M Littman 3200	20 – 2000	8000	Up to 24x	No	$499
Eko Core	20 – 2000	4000	Up to 40x	No	$349
Jabes	20 – 1000	8000	Up to 20x	No	$229
Smartphone	**20 – 2000**	**16000-48000**	**Up to 40x**	**Yes**	**n/a**

Table 4: The table displays different devices that record heart sounds digitals with their respective features (Table taken by Kendre, 2021).

Discussion

Comparative Model Evaluation

Comparing different methods, we observe the 85.7% was the best accuracy reached on the PhysioNet dataset and 61.1% was the best accuracy reached on the PASCAL dataset. Conversely, our proposed method archived 89.5% accuracy on the PhysioNet dataset and 73.7% accuracy on the PASCAL dataset. The increase in performance is attributed to our semi-supervised approach, where we used adversarial Conditional Generators to generate heart sounds. This exposed the Discriminator to a wide range of heart sounds which assisted the model in optimizing for generalized features. Additionally, we conducted a statistical significance test (t-test) to show the probability our proposed method's results are due to random chance. The test concluded the results were statistically significant as all p-values were less than 0.05. This implies that the null hypothesis can be rejected, and the results are statistically significant.

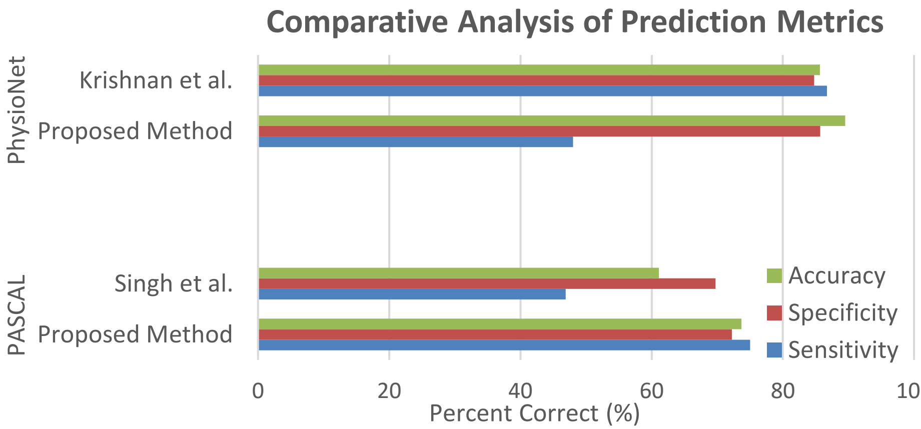


Figure 9: Comparison between proposed method and state-of-the art (Figure taken by Kendre, 2021).

Related Works

Study	Classification techniques	Beat types	Dataset	Time Efficiency	Results
Nogueira et al.	SVM	N & A	PhysioNet 2016	-	Sensitivity: 96.47% Specificity: 72.65% Overall Score: 84.56%
Krishnan et al.	DNN	N & A	PhysioNet 2016	-	Sensitivity: 86.73% Specificity: 84.75% Accuracy: 85.65%
Rubin et al.	CNN	N & A	PhysioNet 2016	-	Specificity: 95% Sensitivity: 73% Overall Score: 84%
Singh et al.	Bayes Net and Logit Boost	N & M	PASCAL	-	Specificity: 46.9% Sensitivity: 69.73% Accuracy: 61.1%
Proposed Method	**GAN, CNN, Transformer**	**N, M, G, No**	**PASCAL**	**~1000 cps**	**Sensitivity: 75.0.3%** **Specificity: 72.2%** **Accuracy: 73.7.0%**
		N & A	**PhysioNet 2016**		**Sensitivity: 48.0%** **Specificity: 85.7%** **Accuracy: 89.5%**
		N, M, G, AF, No	**Combined**		**Sensitivity: 94.3%** **Specificity: 99.5%** **Accuracy: 95.0%**

Table 5: The table displays studies with their proposed classification techniques, beat types (types of heart sounds), dataset, time efficient, and results. N(ormal), A(bnormal), M(urmur), G(allop), No(isy) (Table taken by Kendre, 2021).

Conclusions

We proposed a Generative Adversarial Network (GAN), composed of a Convolution Transformer Generator and a Transformer Discriminator to detect abnormal heart sounds in a recording. The results from model testing and evaluation, along with results from the t-test revealed the proposed method reached better performance than the previous state-of-the-art methods. The introduction of heart sounds analysis with ECGs allowed for increased arrhythmia labels for classification and in a time-efficient manner. Furthermore, the proposed method showed real-world deployment capabilities for autonomous heart sound abnormality detection with recordings collected from a phone microphone.

In terms of future development, we propose conducting prospective clinical trials with patients that have different types of arrhythmias. This will allow us to truly test the generalization capabilities of the model and smartphone app in the real world. Depending on these results, we may opt to develop a low-cost DIY and clinical solution for increased sensitivity in heart recordings. Also, applicable fields include medical emergencies that are time constraint (ER) and developing rural communities that don't have access to arrhythmia expertise.

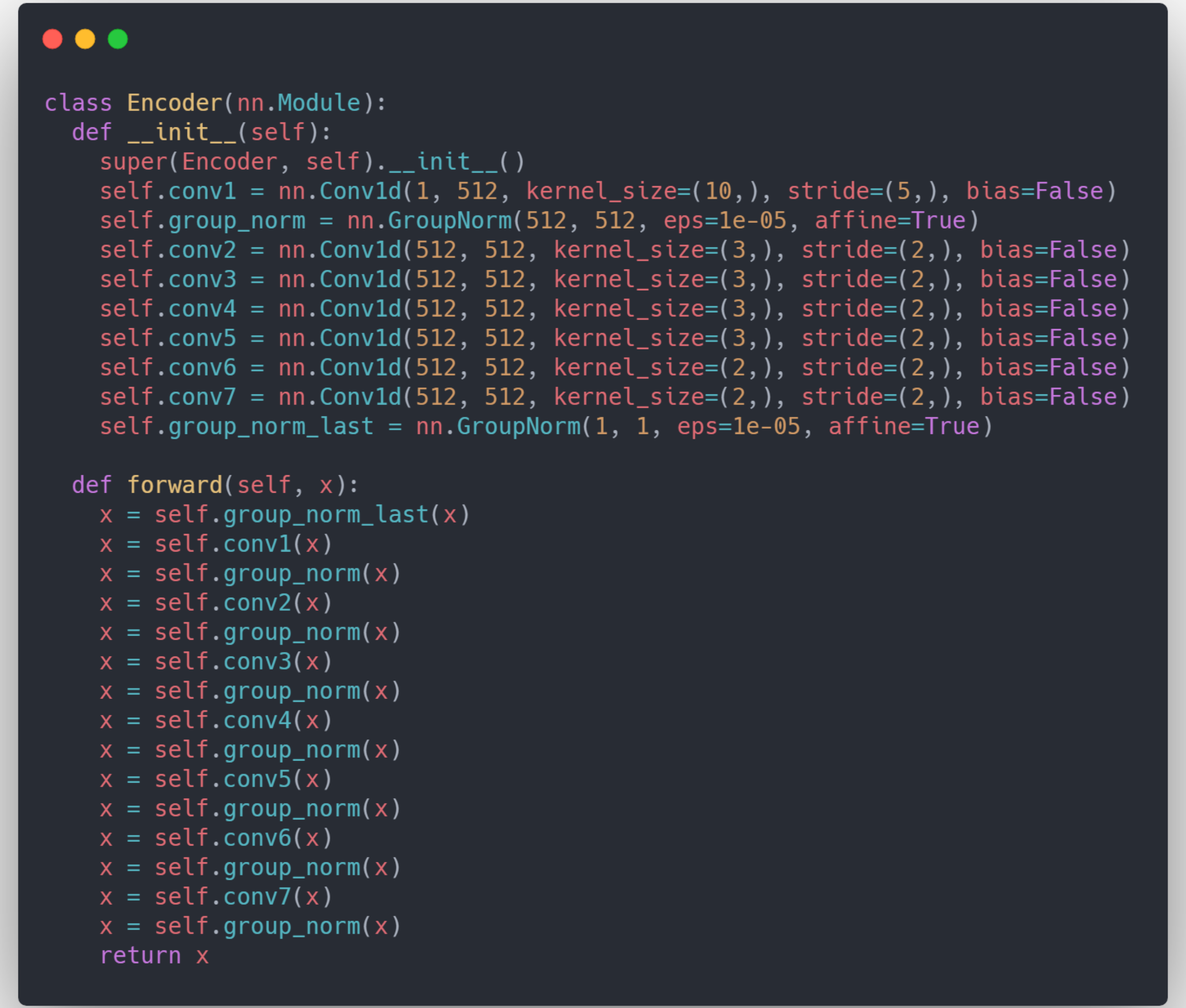
Applications with the development of the multiview approach include language and time series processing. Specifically, we can train models to convert language to speech and speech to language without the need for a supervised dataset of language A and language B. Rather, the model can be trained to convert language A to an intermediary language (language C), this language can then be converted into language B. Moreover, we can train the model to reconstruct speech recording directly from electrical signals (EEGs) from the auditory cortex or reconstruct vision from the visual cortex.

The object of this study was to create a fast and accurate end-to-end heart sound arrhythmia detection system, capable of detecting abnormalities in real-time without specialized equipment. While also increasing the number of cardiovascular pathologies classified. Our proposed method accomplishes exemplary statistics in abnormality detection and shows promising results in increased heart sound synthesis. Hopefully, this study will shed light on abnormality detection techniques and give birth to applications with signal construction.

Further Exploration

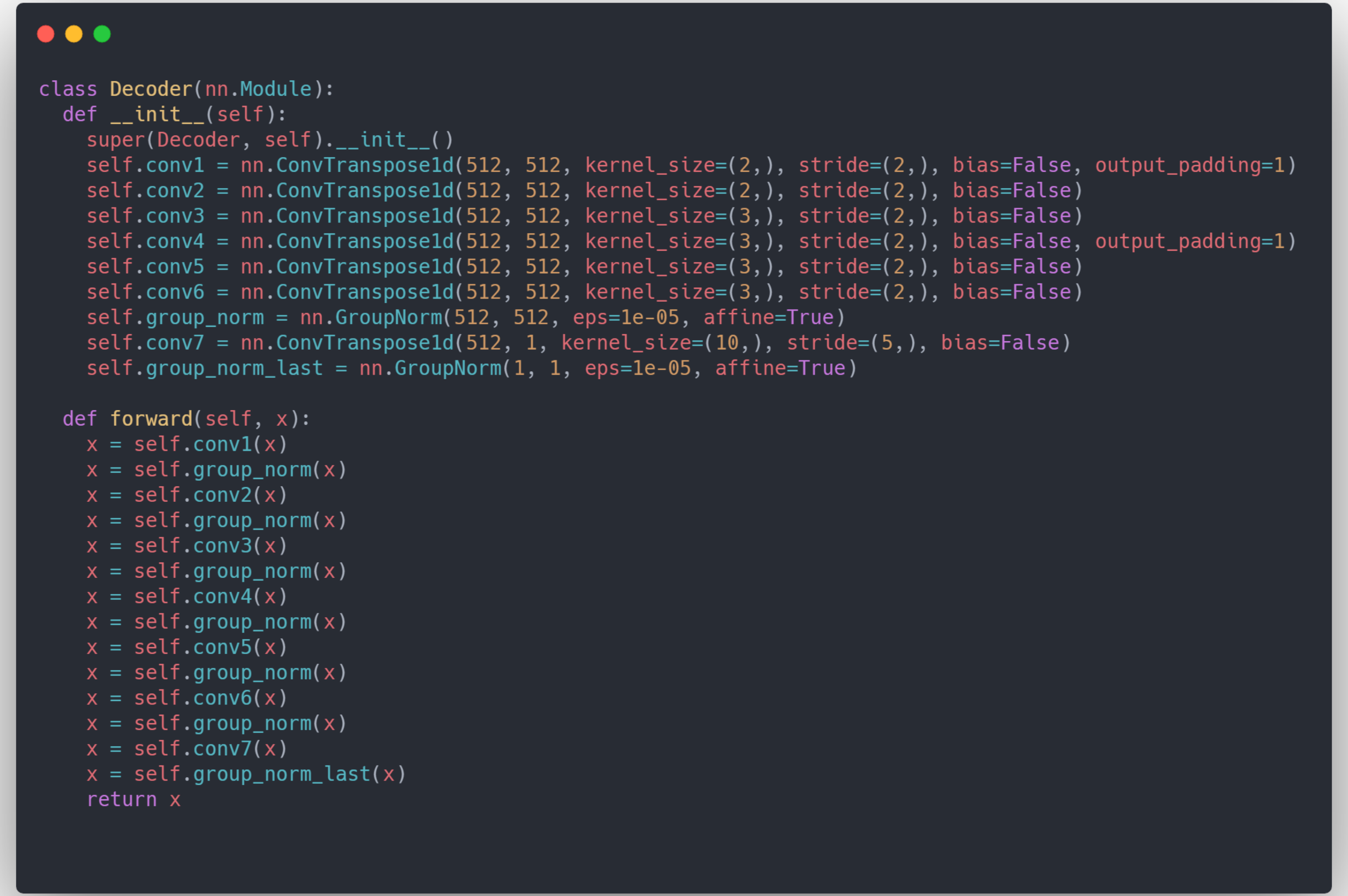
- Deploying the model with an app that is available to 3rd world countries that can't afford to conduct in-depth testing regularly
  - Integrate and serve the model using FastAPI
- Use abnormal heart sound unsupervised datasets as a basis of categorical arrhythmia classification
  - Using low dimensional visualization techniques like t-SNE or UMAP
  - Cluster data using methods like K means and hierarchical clustering
- Create a classifiable latent representation of PCG signal biomarkers that can be represented with accuracy and precision
  - Created by VAE that are fed the PCG signals directly instead of a spectrogram
- Investigate training a heart sound discriminator from generated PCG data from ECG datasets
- Reconstructing speech (wav) recordings from the human auditory cortex (EEG) using techniques used for PCG construction

Code Implementation



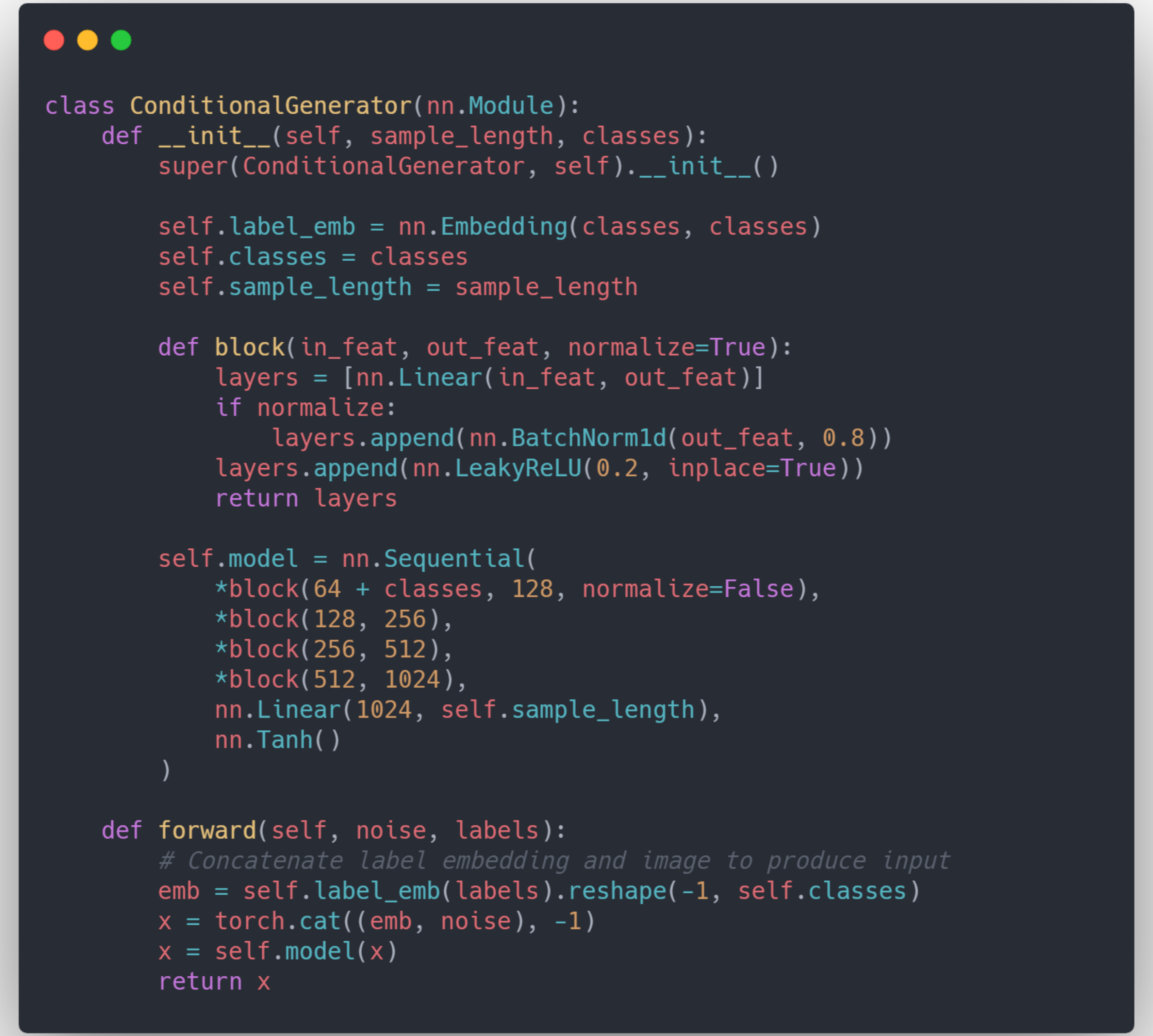
```python
class Encoder(nn.Module):
  def __init__(self):
    super(Encoder, self).__init__()
    self.conv1 = nn.Conv1d(1, 512, kernel_size=(10,), stride=(5,), bias=False)
    self.group_norm = nn.GroupNorm(512, 512, eps=1e-05, affine=True)
    self.conv2 = nn.Conv1d(512, 512, kernel_size=(3,), stride=(2,), bias=False)
    self.conv3 = nn.Conv1d(512, 512, kernel_size=(3,), stride=(2,), bias=False)
    self.conv4 = nn.Conv1d(512, 512, kernel_size=(3,), stride=(2,), bias=False)
    self.conv5 = nn.Conv1d(512, 512, kernel_size=(3,), stride=(2,), bias=False)
    self.conv6 = nn.Conv1d(512, 512, kernel_size=(2,), stride=(2,), bias=False)
    self.conv7 = nn.Conv1d(512, 512, kernel_size=(2,), stride=(2,), bias=False)
    self.group_norm_last = nn.GroupNorm(1, 1, eps=1e-05, affine=True)

  def forward(self, x):
    x = self.group_norm_last(x)
    x = self.conv1(x)
    x = self.group_norm(x)
    x = self.conv2(x)
    x = self.group_norm(x)
    x = self.conv3(x)
    x = self.group_norm(x)
    x = self.conv4(x)
    x = self.group_norm(x)
    x = self.conv5(x)
    x = self.group_norm(x)
    x = self.conv6(x)
    x = self.group_norm(x)
    x = self.conv7(x)
    x = self.group_norm(x)
    return x
```

Code Implementation



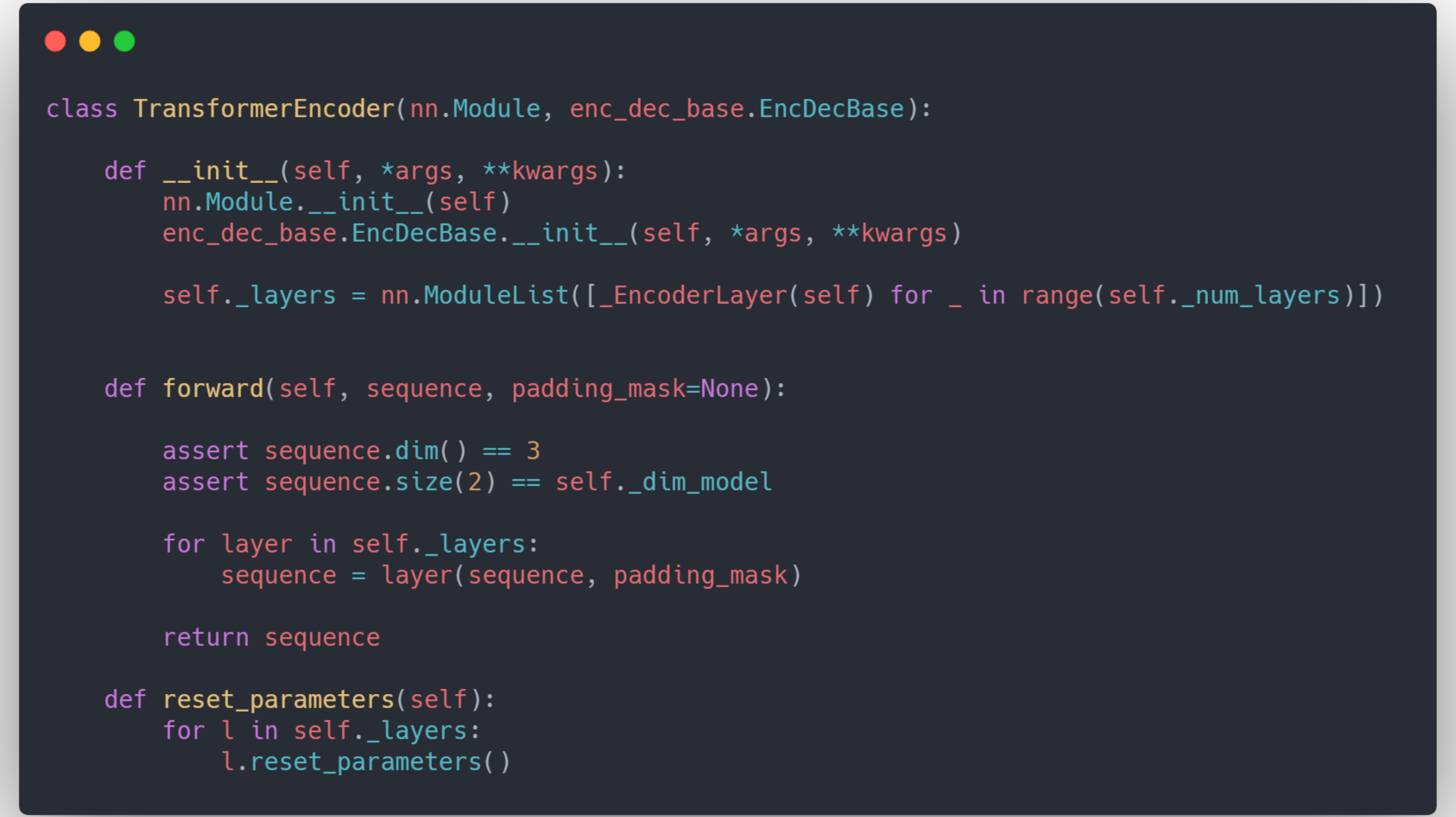
```python
class Decoder(nn.Module):
  def __init__(self):
    super(Decoder, self).__init__()
    self.conv1 = nn.ConvTranspose1d(512, 512, kernel_size=(2,), stride=(2,), bias=False, output_padding=1)
    self.conv2 = nn.ConvTranspose1d(512, 512, kernel_size=(2,), stride=(2,), bias=False)
    self.conv3 = nn.ConvTranspose1d(512, 512, kernel_size=(3,), stride=(2,), bias=False)
    self.conv4 = nn.ConvTranspose1d(512, 512, kernel_size=(3,), stride=(2,), bias=False, output_padding=1)
    self.conv5 = nn.ConvTranspose1d(512, 512, kernel_size=(3,), stride=(2,), bias=False)
    self.conv6 = nn.ConvTranspose1d(512, 512, kernel_size=(3,), stride=(2,), bias=False)
    self.group_norm = nn.GroupNorm(512, 512, eps=1e-05, affine=True)
    self.conv7 = nn.ConvTranspose1d(512, 1, kernel_size=(10,), stride=(5,), bias=False)
    self.group_norm_last = nn.GroupNorm(1, 1, eps=1e-05, affine=True)

  def forward(self, x):
    x = self.conv1(x)
    x = self.group_norm(x)
    x = self.conv2(x)
    x = self.group_norm(x)
    x = self.conv3(x)
    x = self.group_norm(x)
    x = self.conv4(x)
    x = self.group_norm(x)
    x = self.conv5(x)
    x = self.group_norm(x)
    x = self.conv6(x)
    x = self.group_norm(x)
    x = self.conv7(x)
    x = self.group_norm_last(x)
    return x
```

Code Implementation



```python
class ConditionalGenerator(nn.Module):
    def __init__(self, sample_length, classes):
        super(ConditionalGenerator, self).__init__()

        self.label_emb = nn.Embedding(classes, classes)
        self.classes = classes
        self.sample_length = sample_length

        def block(in_feat, out_feat, normalize=True):
            layers = [nn.Linear(in_feat, out_feat)]
            if normalize:
                layers.append(nn.BatchNorm1d(out_feat, 0.8))
            layers.append(nn.LeakyReLU(0.2, inplace=True))
            return layers

        self.model = nn.Sequential(
            *block(64 + classes, 128, normalize=False),
            *block(128, 256),
            *block(256, 512),
            *block(512, 1024),
            nn.Linear(1024, self.sample_length),
            nn.Tanh()
        )

    def forward(self, noise, labels):
        # Concatenate label embedding and image to produce input
        emb = self.label_emb(labels).reshape(-1, self.classes)
        x = torch.cat((emb, noise), -1)
        x = self.model(x)
        return x
```

Code Implementation



```python
class TransformerEncoder(nn.Module, enc_dec_base.EncDecBase):

    def __init__(self, *args, **kwargs):
        nn.Module.__init__(self)
        enc_dec_base.EncDecBase.__init__(self, *args, **kwargs)

        self._layers = nn.ModuleList([_EncoderLayer(self) for _ in range(self._num_layers)])


    def forward(self, sequence, padding_mask=None):

        assert sequence.dim() == 3
        assert sequence.size(2) == self._dim_model

        for layer in self._layers:
            sequence = layer(sequence, padding_mask)

        return sequence

    def reset_parameters(self):
        for l in self._layers:
            l.reset_parameters()
```

References

Abo-Zahhad, M., Farrag, M., Abbas, S. N., & Ahmed, S. M. (2016). A comparative approach between cepstral features for human authentication using heart sounds. *Signal, Image and Video Processing*, *10*(5), 843–851. https://doi.org/10.1007/s11760-015-0826-9

Ali, M. N., El-Dahshan, E.-S. A., & Yahia, A. H. (2017). Denoising of Heart Sound Signals Using Discrete Wavelet Transform. *Circuits, Systems, and Signal Processing*, *36*(11), 4482–4497. https://doi.org/10.1007/s00034-017-0524-7

Almasi, A., Shamsollahi, M. B., & Senhadji, L. (2011). A dynamical model for generating synthetic Phonocardiogram signals. *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 5686–5689. https://doi.org/10.1109/IEMBS.2011.6091376

Aziz, S., Khan, M. U., Alhaisoni, M., Akram, T., & Altaf, M. (2020). Phonocardiogram Signal Processing for Automatic Diagnosis of Congenital Heart Disorders through Fusion of Temporal and Cepstral Features. *Sensors*, *20*(13), 3790. https://doi.org/10.3390/s20133790

Babu, K. A., Ramkumar, B., & Manikandan, M. S. (2017). S1 and S2 heart sound segmentation using variational mode decomposition. *TENCON 2017 - 2017 IEEE Region 10 Conference*, 1629–1634. https://doi.org/10.1109/TENCON.2017.8228119

Bashar, Md. K., Dandapat, S., & Kumazawa, I. (2018). Heart Abnormality Classification Using Phonocardiogram (PCG) Signals. *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, 336–340. https://doi.org/10.1109/IECBES.2018.8626627

Bau, D., Zhu, J.-Y., Strobelt, H., Zhou, B., Tenenbaum, J. B., Freeman, W. T., & Torralba, A. (2018). GAN Dissection: Visualizing and Understanding Generative Adversarial Networks. *ArXiv:1811.10597 [Cs]*. http://arxiv.org/abs/1811.10597

D, P., T, U. M., K, P., & A, S. (n.d.). *Detection of Heart Diseases by Mathematical Artificial Intelligence Algorithm Using Phonocardiogram Signals*.

Dao, A. T. (2015). Wireless laptop-based phonocardiograph and diagnosis. *PeerJ*, *3*, e1178. https://doi.org/10.7717/peerj.1178

Das, A., Agrawal, H., Zitnick, C. L., Parikh, D., & Batra, D. (2016). Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions? *ArXiv:1606.03556 [Cs]*. http://arxiv.org/abs/1606.03556

Deperlioglu, O., Kose, U., Gupta, D., Khanna, A., & Sangaiah, A. K. (2020). Diagnosis of heart diseases by a secure Internet of Health Things system based on Autoencoder Deep Neural Network. *Computer Communications*, *162*, 31–50. https://doi.org/10.1016/j.comcom.2020.08.011

Electrocardiogram_generation_with_a_bidirectional_.pdf. (n.d.).

Fernando, T., Ghaemmaghami, H., Denman, S., Sridharan, S., Hussain, N., & Fookes, C. (2020). Heart Sound Segmentation using Bidirectional LSTMs with Attention. *IEEE Journal of Biomedical and Health Informatics*, *24*(6), 1601–1609. https://doi.org/10.1109/JBHI.2019.2949516

Garg, V., Mathur, A., Mangla, N., & Rawat, A. S. (2019). Heart Rhythm Abnormality Detection from PCG Signal. *2019 Twelfth International Conference on Contemporary Computing (IC3)*, 1–5. https://doi.org/10.1109/IC3.2019.8844950

Ger, S., & Klabjan, D. (2020). Autoencoders and Generative Adversarial Networks for Imbalanced Sequence Classification. *ArXiv:1901.02514 [Cs, Stat]*. http://arxiv.org/abs/1901.02514

Ismail, S., Siddiqi, I., & Akram, U. (2018). Localization and classification of heart beats in phonocardiography signals—A comprehensive review. *EURASIP Journal on Advances in Signal Processing*, *2018*(1), 26. https://doi.org/10.1186/s13634-018-0545-9

Jia, Y., Weiss, R. J., Biadsy, F., Macherey, W., Johnson, M., Chen, Z., & Wu, Y. (2019). Direct speech-to-speech translation with a sequence-to-sequence model. *ArXiv:1904.06037 [Cs, Eess]*. http://arxiv.org/abs/1904.06037

Jordaens, L. (2018). A clinical approach to arrhythmias revisited in 2018: From ECG over noninvasive and invasive electrophysiology to advanced imaging. *Netherlands Heart Journal*, *26*(4), 182–189. https://doi.org/10.1007/s12471-018-1089-1

Kalaivani, V. (2014). DIAGNOSIS OF ARRHYTHMIA DISEASES USING HEART SOUNDS AND ECG SIGNALS. *Russian Journal of Cardiology*, *1-ENG*, 35–41. https://doi.org/10.15829/1560-4071-2014-1-ENG-35-41

Kiranyaz, S., Zabihi, M., Rad, A. B., Ince, T., Hamila, R., & Gabbouj, M. (2020). Real-time phonocardiogram anomaly detection by adaptive 1D Convolutional Neural Networks. *Neurocomputing*, *411*, 291–301. https://doi.org/10.1016/j.neucom.2020.05.063

Krishnan, P. T., Balasubramanian, P., & Umapathy, S. (2020). Automated heart sound classification system from unsegmented phonocardiogram (PCG) using deep neural network. *Physical and Engineering Sciences in Medicine*, *43*(2), 505–515. https://doi.org/10.1007/s13246-020-00851-w

Latif, S., Usman, M., Rana, R., & Qadir, J. (2020). Phonocardiographic Sensing using Deep Learning for Abnormal Heartbeat Detection. *ArXiv:1801.08322 [Cs]*. http://arxiv.org/abs/1801.08322

Liu, C., Springer, D., Li, Q., Moody, B., Juan, R. A., Chorro, F. J., Castells, F., Roig, J. M., Silva, I., Johnson, A. E. W., Syed, Z., Schmidt, S. E., Papadaniil, C. D., Hadjileontiadis, L., Naseri, H., Moukadem, A., Dieterlen, A., Brandt, C., Tang, H., … Clifford, G. D. (2016). An open access database for the evaluation of heart sound algorithms. *Physiological Measurement*, *37*(12), 2181–2213. https://doi.org/10.1088/0967-3334/37/12/2181

Lubaib, P., & Muneer, K. V. A. (2016). The Heart Defect Analysis Based on PCG Signals Using Pattern Recognition Techniques. *Procedia Technology*, *24*, 1024–1031. https://doi.org/10.1016/j.protcy.2016.05.225

Machine Learning Visualization. A collection of a few interesting… _ by Pier Paolo Ippolito _ Towards Data Science.pdf. (n.d.).

Messner, E., Zohrer, M., & Pernkopf, F. (2018). Heart Sound Segmentation—An Event Detection Approach Using Deep Recurrent Neural Networks. *IEEE Transactions on Biomedical Engineering*, *65*(9), 1964–1974. https://doi.org/10.1109/TBME.2018.2843258

Model Interpretability and Visualization_ Looking Inside the Neural Network Black Box _ by Avinash _ Heartbeat.pdf. (n.d.).

PhysioNet/CinC Challenge 2016: Training Sets. (n.d.). Retrieved January 11, 2021, from https://archive.physionet.org/pn3/challenge/2016/

Potes, C., Parvaneh, S., Rahman, A., & Conroy, B. (2016, September 14). *Ensemble of Feature:based and Deep learning:based Classifiers for Detection of Abnormal Heart Sounds*. 2016 Computing in Cardiology Conference. https://doi.org/10.22489/CinC.2016.182-399

Rajpurkar, P., Hannun, A. Y., Haghpanahi, M., Bourn, C., & Ng, A. Y. (2017). Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks. *ArXiv:1707.01836 [Cs]*. http://arxiv.org/abs/1707.01836

Rawther, N. N., & Cheriyan, J. (2015). *Detection and Classification of Cardiac Arrhythmias based on ECG and PCG using Temporal and Wavelet features. 4*(4), 6.

S1 and S2 Heart Sound Recognition Using Deep Neural Networks. (2017). *IEEE Transactions on Biomedical Engineering*, *64*(2), 372–380. https://doi.org/10.1109/TBME.2016.2559800

Smilkov, D., Thorat, N., Kim, B., Viégas, F., & Wattenberg, M. (2017). SmoothGrad: Removing noise by adding noise. *ArXiv:1706.03825 [Cs, Stat]*. http://arxiv.org/abs/1706.03825

Subasi, A. (2019). Biomedical Signals. In *Practical Guide for Biomedical Signals Analysis Using Machine Learning Techniques* (pp. 27–87). Elsevier. https://doi.org/10.1016/B978-0-12-817444-9.00002-7

Surukutla, D., Bhanushali, K., & Patil, T. (2020). Cardiac Arrhythmia Detection Using CNN. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3572237

Thoms, L.-J., Collichia, G., & Girwidz, R. (2019). Real-life physics: Phonocardiography, electrocardiography, and audiometry with a smartphone. *Journal of Physics: Conference Series*, *1223*, 012007. https://doi.org/10.1088/1742-6596/1223/1/012007

Yupapin, P., Wardkein, Yupapin, P., Phanphaisarn, Koseeyaporn, Roeksabutr, Roeksabutr, Wardkein, & Koseeyapon. (2011). Heart detection and diagnosis based on ECG and EPCG relationships. *Medical Devices: Evidence and Research*, 133. https://doi.org/10.2147/MDER.S23324

Zeiler, M. D., & Fergus, R. (2013). Visualizing and Understanding Convolutional Networks. *ArXiv:1311.2901 [Cs]*. http://arxiv.org/abs/1311.2901

Zhang, W., Han, J., & Deng, S. (2017). Heart sound classification based on scaled spectrogram and partial least squares regression. *Biomedical Signal Processing and Control*, *32*, 20–28. https://doi.org/10.1016/j.bspc.2016.10.004

Zhao, C., Sun, Q., Zhang, C., Tang, Y., & Qian, F. (2020). Monocular Depth Estimation Based On Deep Learning: An Overview. *Science China Technological Sciences*, *63*(9), 1612–1627. https://doi.org/10.1007/s11431-020-1582-8

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning Deep Features for Discriminative Localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2921–2929. https://doi.org/10.1109/CVPR.2016.319