EMPLOYING ADVERSARIAL MACHINE LEARNING AND COMPUTER AUDITION FOR SMARTPHONE-BASED REAL-TIME ARRHYTHMIA CLASSIFICATION IN HEART SOUNDS

Aditya Kendre
Cumberland Valley High School
Mechanicsburg, PA, USA

March 16, 2021

ABSTRACT

We propose a novel approach to detect arrhythmias in Phonocardiograms (PCGs). Typically, many arrhythmia conditions are unknown until a patient is suggested an ECG/EKG test. PCGs provide ease of access to everyone who has a device capable of recording audio, allowing medical professionals to treat arrhythmias in the developmental stages. The new design is comprised of two subsystems; one is based on the relationship between Electrocardiograms (ECGs) and PCGs, and the other between PCGs and arrhythmias. The first subsystem uses a Generative Adversarial Networks (GAN), in which both generated and real PCG signals are fed into the discriminator for classification. In subsystem two, ECG spectrograms are dimensionally reduced, then constructed into PCG spectrograms using a transGAN. These constructed PCG spectrograms, when converted back into time series, should be identical to the ground truth. This would allow the transGAN to convert ECG datasets into PCG datasets, providing subsystem one to train on both ECG and PCG datasets. After testing, the GAN model (subsystem one) should achieve an accuracy of 86.02%, a specificity of 77.81%, and a sensitivity of 94.24% on the testing set. Furthermore, the transGAN should show promising results, in that the transGAN discriminator should be able to construct the PCG spectrogram accurately. With this data, we should be able to use subsystem one to create a smartphone-based app to detect arrhythmias in heart sound recordings. Our proposed method should accomplish exemplary statistics in abnormalities detection and show promising results in increased arrhythmia construction.

***K*eywords** Arrhythmias · Phonocardiograms · Electrocardiograms · Biomarkers

Contents

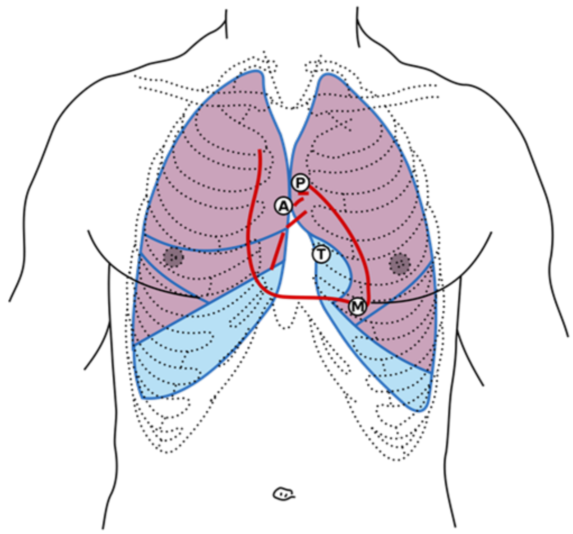

Figure 1: Representation of heart sound recording positions.

1 Rationale

Motivation To create a fast and accurate model capable of detecting Cardiovascular modalities in real-time, specifically a variety arrhythmia's in heart sound recordings (PCGs) without the need for specialized equipment.

An estimated three million cases of arrhythmia occur in the United States yearly (Mayo Clinic), with 300,000 sudden deaths per year – an incidence rather higher than stroke, lung cancer, or breast cancer (American Heart Association). Traditionally, non-invasive arrhythmia analysis is based on multiple electrodes that reflect the electrical activity on ECGs. This method, despite being accurate, limits the use case to hospitals and clinics with specialized equipment; thus, limiting the portability of diagnosing, let alone classification of the type of pathology.

Phonocardiograms (PCGs) are sounds that are created by the mechanical movement of the heart. This physical movement produces four distinct sounds: S1, S2, S3, S4, and murmurs. S1 and S2 are sounds created by a healthy heart; whereas, S3, S4, and murmurs refer to diseases or anomalies.The first heat sound, S1, marks the start of Systole. Systole occurs when the heart muscle contracts and pumps blood from the chambers into the arteries. The second heart sound, S2, marks the end of Systole and the start of Diastole. Diastole is a phase of the heartbeat when the heart muscle relaxes and allows the chambers to fill with blood.

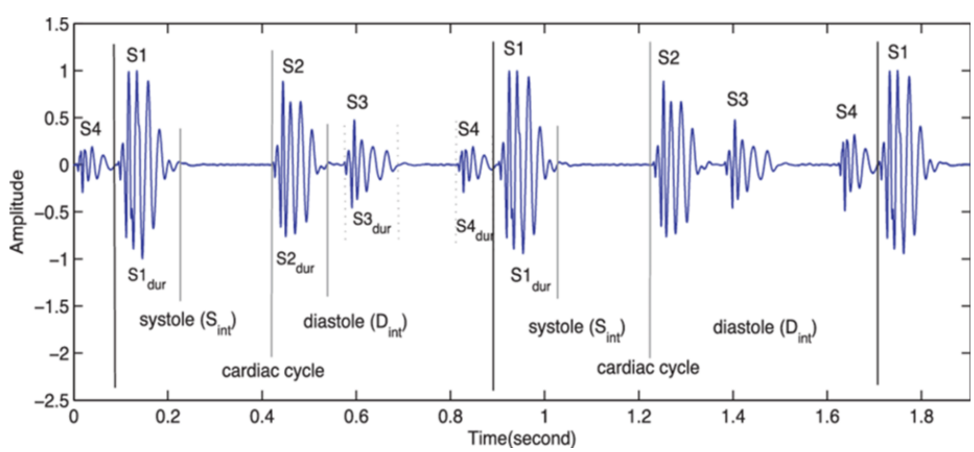

Figure 2: Illustrates the S1, S2, S3, and S4 biomarkers of heart sounds.

Although heart sound databases do exist, these datasets are still 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 Recording Challenge dataset, PASCAL Heart Sound Challenge dataset, and the Heart Sound and Murmur Library. The presently available PCG datasets have a limited number of samples and do not cover the complete range of pathologies that are likely to be encountered in clinical settings.

In diagnosing heart sounds, two major challenges arise: localization and classification. Localization aims to find the position of the aforementioned 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, frequency, or both, 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.

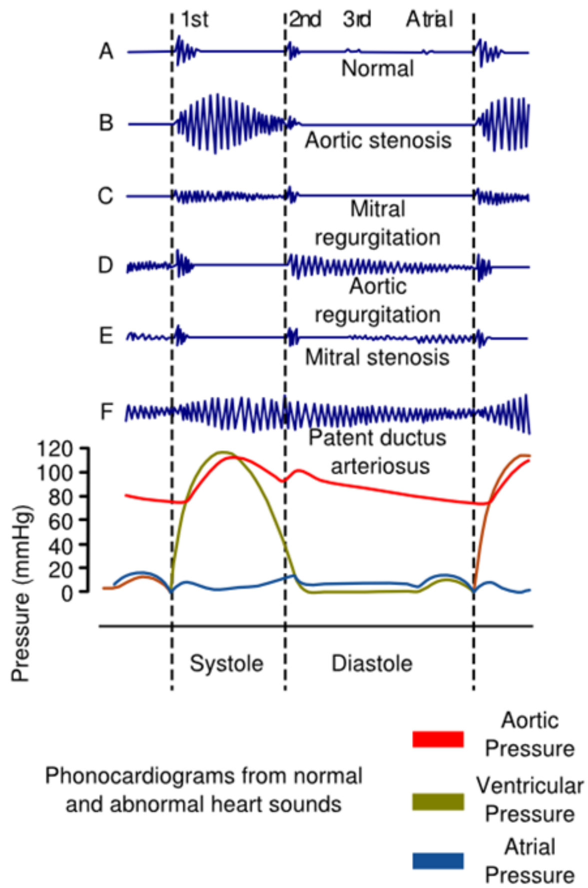

Figure 3: Representation of different abnormalities in sound and pressure.

The main challenge of ineffective heart sound detection stems from an analysis of noisy heartbeats, e.g., background noise. For clean datasets, e.g., the PhysioNet Challenge dataset, a varieties time and frequency of methods converged on localization accuracy of 96.9% (Fernando et al.) and 86.02% classification accuracy (Potes et al.). From the viewpoint of practical applications, the development of computationally efficient solutions is extremely important to the success of a model's deployment. Many studies have negated to comment on the practicality of their proposed methods. From our research, we have concluded only two studies have noted their time efficiency, (Fernando et al.) and (Messner et al.). The fastest model processed 1000 heart state classifications in 56.88 seconds (Fernando et al.), suggesting the model can process 18 bps. Thus, current models need severe optimization to achieve near to real-time analysis. These results are excluding the classification of heart arrhythmias.

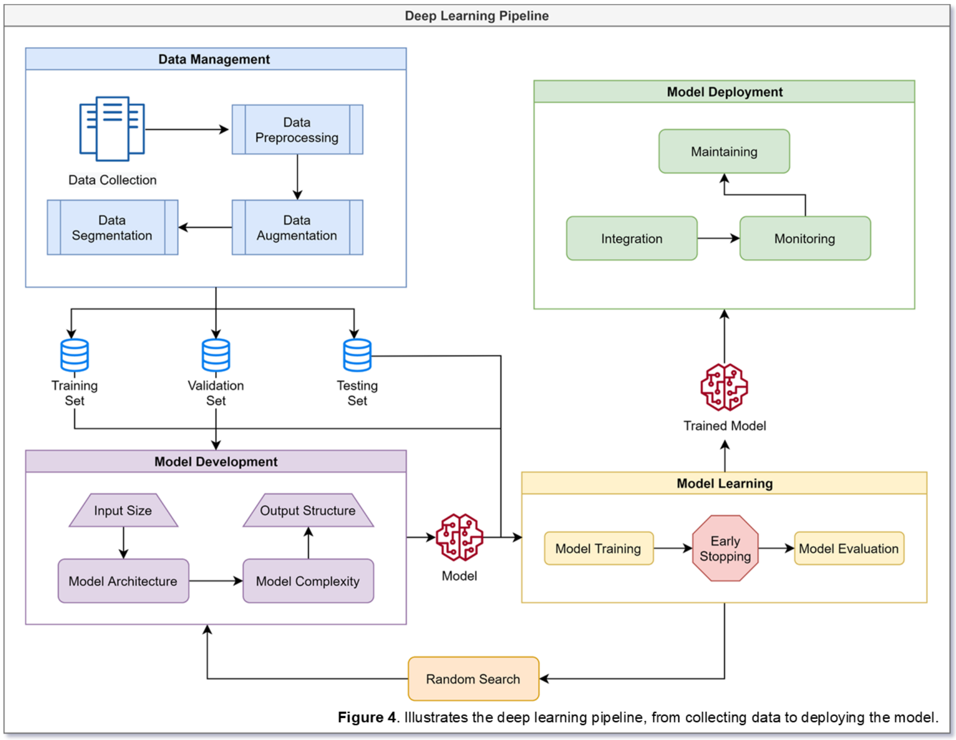

Figure 4: Illustrates the deep learning pipeline, from collecting data to deploying the model.

Thus, the problem of computationally efficient and accurate classification of noisy heartbeats, especially with datasets with a variety of pathologies still remains a problem.

2 Objectives

Problem Current detection methods have limited performance in pathologies and lack real-time classification capabilities.

Question Is it possible to use Generative Adversarial Networks (GANs) to accurately detect arrhythmias in PCGs and surpass previous methods in detection tasks?

Hypothesis If a Generative Adversarial Network is used to create spurious data, then the model will outperform previous state-of-the-art methods in classification, because the specious data will aid the model in extracting significant features from a ground truth dataset.

Engineering Goals

1. **Develop a System for End-to-End Heart Sound Arrhythmia Detection** — Create a system that is able to record and analyze heart sounds for Cardiovascular modalities. The system should implement an adversarial model that is both time and space-efficient, and accurate.
2. **Increase the Number of Cardiovascular Pathologies** — Develop a model to construct heart sounds from pre-existing data.
3. **Real-World Testing** — Test the end-to-end system in a real-world environment to ensure practicality and generality of the system.

Constraints Constraints include a time complexity complexity of O(n) and an accuracy above 90%. These constraints are to ensure that the model is comparable to the current state-of-the-art performance and has computationally efficient for real-world use.

Expected Outcomes Constraints include a time complexity complexity of O(n) and an accuracy above 90%. These constraints are to ensure that the model is comparable to the current state-of-the-art performance and has computationally efficient for real-world use.

3 Data Management

3.1 Data Collection

Although PCG signals are analyzed less often than ECG signals, these signals are rather analyzed in real-time by physicians and healthcare workers. Preliminary studies were done on PCG segmentation and classification primarily used private datasets. Hence, there existed no publicly available datasets until recently. Since then, many public datasets have been developed aiding researchers in their studies and creating open benchmarks for researchers to use in comparing similar findings. However, these datasets are still limited by the number of classes that are collected, when compared to ECG datasets.

Currently, only three major supervised PCG datasets exist: PhysioNet Classification of Heart Sound Recording Challenge dataset, PASCAL Heart Sound Challenge dataset, and the Heart Sound and Murmur Library. These datasets are all anonymized and de-identified for the safety of their subjects, and thus includes no personal information such as name, income, age, etc.

The PhysioNet Classification of Heart Sound Recording Challenge dataset was produced as a part of the 2016 PhyisoNet Computing in Cardiology Challenge. The heart sounds were collected from both clinical and non-clinical environments (in-home visits). The challenge focused on creating an accurate dataset of normal and abnormal heart sound recordings, especially in real-world (extremely noisy and low signal quality) scenarios. These recordings were sourced from nine independent databases and in total, contain 4,593 heart sound recordings from 1072 subjects, lasting from 5-120 seconds. Of which, 409 recordings that were collected from 121 patients contain one PCG lead and one simultaneously recorded ECG. Though, all recordings were resampled to 2,000 Hz using an anti-alias filter. Furthermore, the dataset is comprised of 3 classes: normal, abnormal, and unsure (this is due to poor recording quality), and have the following proportion respectively: 77.1%, 12,0%, 10.9%.

The PASCAL Classifying Heart Sounds Challenge dataset was released to the general public in 2011. The challenge consisted of two sub-challenges: heart sound segmentation, and heart sounds classification; these sub-challenges corresponded with dataset A, and dataset B respectively. Both datasets have recordings of varying lengths, between 1 second and 30 seconds. Dataset A was collected via the iSethoscope Pro iPhone app, and contained 176 heart sound recordings. 124 of which are divided into four classes: Normal (31 recordings), Murmur (34 recordings), Extra heart sound (19 recordings), and Artifact (40 recordings); the rest of the records are unlabeled for testing purposes. Dataset B was collected using a DigiScope (a digital stethoscope), and included 656 heart sounds. All except 370 were separated into three classes: Normal (320 recordings), Murmur (95 recordings), and Extra-systole (46 recordings). Both datasets A and B vary in sound recordings between lengths of 1 second and 30 seconds.

More than 300 million ECG recordings are analyzed yearly, and thus create an exceptional tool for arrhythmia classification. Coupled with the recent surge in research interest in 2015, many massive publicly available datasets have been published, notable by PhysioNet - the moniker of the Research Resource for Complex Physiologic Signals. Numerous, datasets ECG exist, however, many are limited to few classes (Normal and Abnormal). At present, three public datasets exist that have more than 4 classes: AF Classification Challenge 2017, PTB Diagnostic ECG, and PTB-XL dataset. Additionally, iRhythm Technologies have developed a semi-public dataset, that is available upon request, that contains 12 classes.

The PTB-XL is the largest publicly available dataset for ECGs and contains 21,837 clinical 12-lead ECG recordings from 18,885 patients of 10 second length. These recordings are separated into 5 super-classes: Normal, Myocardial Infraction, Hypertrophy, ST/T-Change, and Conduction Disturbance. These super-classes are further split into 71 sub-classes that range from AV Block to Posterior Myocardial Infraction. The raw signal data were downsampled to 100 Hz and annotated by up to two cardiologists, who assigned potentially multiple ECG statements to each record. iRhythm Technologies developed a large, 12 classes ECG dataset using raw single-lead ECG inputs. The 12 classes include Atrial fibrillation and flutter, AVB, Bigeminy, EAR, IVR, Junctional rhythm, Noise, Sinus rhythm, SVT, Trigeminy, Ventricular tachycardia, and Wenckebach. The dataset consists of 91,232 ECG recordings from 53,549 patients. This training dataset is available upon request under license from iRhythm Technologies, Inc. The publicly available test dataset contains 328 records collected from 328 unique patients, split between 6 classes. Both datasets were recorded using a Zio monitor, which monitors the heart through a single-lead sensor at 200 Hz. The annotation was done by a consensus committee of expert cardiologists.

The PhysioNet AF Classification database, presented in 2017 for the Computing in Cardiology Challenge, contains 8,528 ECG recordings, divided into 4 classes: Normal (5154 recordings), Atrial Fibrillation (771 recordings), Other arrhythmias (2557 recordings), and Noisy (46 recordings). The single-lead recordings last from 9 - 61 seconds, with a mean of 32.5 seconds and a standard deviation of 10.9 seconds. The ECG recordings were sampled to 300 Hz and provided in MATLAB V4 WFDB-compliant format.

3.2 Data Preprocessing

PCG recordings often are recording in non-ideal environments that are filled with unwanted background noise and interference. Data preprocessing is the process of altering the data in the signal, often by denoising, normalizing, standardizing, and transforming the signal. These steps are crucial for automatic localization and classification tasks. Preprocessing the data allows a model to extract meaning features efficiently and reveals the physiological structure of the heart sounds [Latif et al.]. Furthermore, preprocessing helps ensure that the data that is fed into the model is always in the same domain. This allows the model to generalize more easily.

We first resample the data to 500 Hz, to decrease the spatial resolution of the heart sound recordings, but still retain important features. Thus, helping the model to converge faster. The resampled data is standardized using the standard score equation. This scales the mean of the distribution to 0, artificially scaling all data into similar ranges, thus, helping combat the exploding gradient problem.

The standardized data is then fed into a CycleGAN that has learned to denoise data. The CycleGAN is fed synthetically noised PCG signal and attempts to construct the denoised data from the noisy data. This synthetic noise consists of white noise, pink noise, and real background noise collected from audio recordings. The noise is added to each PCG signal recording and then treated as the input to the CycleGAN. The CycleGAN's output is compared to the original, non-noise, PCG recording. In this way, the CycleGAN eventually learns to denoise PCG recordings.

3.3 Data Segmentation

Data segmentation refers to the process of creating cross-validation datasets. This process assists in validating if the model is overfitting to the dataset. These datasets include the training set, validation set, and testing set. Typically, the training set is 70%-80% of the dataset, the reset of the dataset is split among the validation set and testing set. Here, we split the data 80% training, 10% validation, and 10% testing.

3.4 Data Augmentation

Data augmentation is a strategy that enables a significant increase in the diversity of data available while training a model, without actually collecting new data. Data augmentation techniques aim 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 retains the characteristics of the data's category. This helps in reinforcing important features within the data and is only done during the training portion of the workflow.

A common misconception arises when comparing preprocessing and augmentation. To be clear, preprocessing aims to clean the data of unwanted artifacts that are not meant for classification and is done in place. Augmentation, on the other hand, is solely done for expanding the dataset's size, often to combat overfitting. Augmenting the data before preprocessing further obscures the data unrealistically, and beyond classification.

We use to resample the heart sound recordings to different frequencies to simulate slower and faster beats per minute (bpm). The normal bpm for a human is between 60-100 bpm. Thus, measuring the sample distance between the first S1 (the start of systole) and the second S1, we calculate the bps and resample accordingly.

Furthermore, we use noise injection directly to preprocessed PCG recordings [Messner et al.]. This process is identical to the process of synthetically adding noise to PCG recordings described in the preprocessing step. A variety of noises, like white noise, is added to the signal to increase the sample of recordings per class. This method is extremely beneficial for training on small datasets, like the PASCAL dataset.

4 Model Development

4.1 Model Architecture

Here we propose using Generative Adversarial Networks (GANs) for increased success in PCG heart sound detection. GANs pose a unique advantage over traditional machine learning and deep learning methods, in that a model learns to mimic a dataset by creating its own data, and tries to fool a discriminator into thinking the generated data is real. In a supervised approach, a GAN consists of two parts, a generator and a discriminator. The generator is responsible for creating fake heart sound data, while the discriminator tries to predict where the incoming data is fake or real. In a semi-supervised approach, however, the is fed data from a real dataset and the generator. Here, the discriminator tries to classify the generator's fake data, as well as predict the classes form the real dataset.

4.2 Model Complexity

Traditionally, generators are dense layers that slowly increase the dimensionality of the generated data to match that of the real dataset. Discriminators, on the other hand, are commonly CNNs because the majority of their applications work with images. However, it is possible to use a wide variety of architectures; such as LSTMs, RNNs, SVMs, DNNs, ANNs, Transformers. As mentioned previously, there are many types of model architecture, some are used for classification, and others for feature extraction. Optimizing the combination of feature extraction layers and classification layers is extremely time-consuming and computationally taxing. This is because there exist many combinations of hyperparameters, thus making it difficult to optimize each parameter. To optimize hyperparameters, we used hyperparameter sweeps to make the optimization process more efficient. This method involves using one of three methods: grid search, random search, and Bayesian search. Grid search computes each possible combination of all hyperparameters and tests them all. Although this is very effective, it can be computationally costly. Random search selects a new combination at random, provided a distribution of values. This method is surprisingly effective and scales very well. Bayesian search creates a probabilistic model of metrics and suggests parameters that have a high probability of improving metrics. This works well for small-scale projects, but scales poorly as the complexity of parameter relationships increases. Here, we used a random search to optimize our hyperparameters.

5 Model Learning

5.1 Model Training

During the training phase, the model is trained using backpropagation in conjunction with a cost function. Backpropagation attempts to calculate the gradient of the cost function with respect to the weight and biases of the model. This process involves an optimizer, which optimizes the model's parameters and a cost function that measure the correctness or incorrectness of the model. The goal of the optimizer is to minimize the cost function's error by adjusting the parameters to the given label. In this study, we used the Adam optimizer in union with Cross-Entropy Loss. The Adam optimizer uses a hyperparameter that dictated the change in the model's parameters on each backpropagation step, this is called the learning rate. Here we choose a learning rate of 0.0001.

The model is only trained on the training set; thus, backpropagation only occurs on the training set. Additionally, for each step in the training set, the optimizer backpropagates and optimizes the parameters and calculates metrics to further evaluate the model. The amount of steps in the training set is dictated by the batch size, the number of signals the model is trained on, in a single forward pass. Here we use a batch size of 32, meaning that the model is fed 32 signals per input. This significantly speeds up the process of training as more signals are passed through the model every time the model is optimized. A full pass of the training set is called an Epoch, here we train the model on 100 Epochs.

5.2 Model Training

To ensure the model is not overfitting, but generalizing to the training set, we use a validation set to track the metrics of the model. In theory, the metrics on the training set equal to that of the validation set. In practicality, after many epochs of training the metrics of the validation set become static, but the metrics of the training set still increase. This suggests that the model is overfitting. Thus, we stop training the model on the training set and test it as a testing set.

5.3 Model Evaluation/Data Analysis

Testing sets or hold-out sets are used to validate the metrics of the model, this is because both the validation set and the testing set have been tested by the model; thus, the model has developed a latent bias to both sets. Therefore, a third set is needed to assess the model's ability to generalize on an independent dataset. The metrics calculated on the testing set include the Accuracy, Sensitivity, Specificity, and ROC/AUC (and MSELoss in the case of the VQGAN).

6 Model Deployment

Model Deployment is one of the last stages of any machine learning project and involves releasing the model to the public.

6.1 Integration

Integration consists of implementing the model in a system, whether it happens on the client-side or the backend. The most popular backend model integration tools involve Flask, Azure, and FastAPI. These tools create APIs that encapsulate the model prediction, given a GET request with the desired input.

6.2 Monitoring & Maintaining

Following model integration and deployment, we move onto the next phase, monitoring and maintaining the system. As more and more data passes through the model, it increases the opportunity for the model to learn from a more generalized dataset. Though such data would be unsupervised, we could use unsupervised techniques to categories the data. Based on the improvement of the model, the model and be reintegrated and deployed. In essence, looping the whole process from data management to model learning.

6.3 Potential Problems

6.4 Risk and Safety

The equipment used in this research include mobile devices such as phones and laptops, these devices pose no risk.

6.5 Overfitting

One of the largest problems in Deep Learning overall, which possesses a threat to our model is overfitting. Overfitting typically happens when the model metrics of the training and validation set diverge. This suggests that the model is not generalizing, but rather memorizing the training dataset. To combat overfitting, researchers typically implement data argumentation techniques to reinforce important features in a dataset.

6.6 Domain Shift

A domain shift occurs when a source dataset performs well but on a different dataset distribution, the performance drastically decreases. Typically, domain adaptation is often used to improve performance on target datasets. This is done by training the model itself on multiple datasets to improve the model's capacity to generalize.

6.7 Traning Time

With large multi-model architectures, it becomes tough to train models on a single GPU. This can happen for a number of reasons, but the main reason is because the model takes up too much memory of the GPU. Generally, parallel processing is used to split tasks and assign them to different GPUs. For instance, the discriminator model will run on a single GPU, while the generator will run on another GPU.

References

[1] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. GAN Dissection: Visualizing and Understanding Generative Adversarial Networks. *arXiv:1811.10597 [cs]*, December 2018. arXiv: 1811.10597.

[2] Prakash D, Uma Mageshwari T, Prabakaran K, and Suguna A. *Detection of Heart Diseases by Mathematical Artificial Intelligence Algorithm Using Phonocardiogram Signals*.

[3] PhysioNet/CinC Challenge 2016: Training Sets.

[4] Nabina N Rawther and Jini Cheriyan. Detection and Classification of Cardiac Arrhythmias based on ECG and PCG using Temporal and Wavelet features. 4(4):6, 2015.

[5] Shahid Ismail, Imran Siddiqi, and Usman Akram. Localization and classification of heart beats in phonocardiography signals —a comprehensive review. *EURASIP Journal on Advances in Signal Processing*, 2018(1):26, December 2018.

[6] Omer Deperlioglu, Utku Kose, Deepak Gupta, Ashish Khanna, and Arun Kumar Sangaiah. Diagnosis of heart diseases by a secure Internet of Health Things system based on Autoencoder Deep Neural Network. *Computer Communications*, 162:31–50, October 2020.

[7] Dinesh Surukutla, Karan Bhanushali, and Trupti Patil. Cardiac Arrhythmia Detection Using CNN. *SSRN Electronic Journal*, 2020.

[8] Abdulhamit Subasi. Biomedical Signals. In *Practical Guide for Biomedical Signals Analysis Using Machine Learning Techniques*, pages 27–87. Elsevier, 2019.

[9] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, Las Vegas, NV, USA, June 2016. IEEE.

[10] Palani Thanaraj Krishnan, Parvathavarthini Balasubramanian, and Snekhalatha Umapathy. Automated heart sound classification system from unsegmented phonocardiogram (PCG) using deep neural network. *Physical and Engineering Sciences in Medicine*, 43(2):505–515, June 2020.

[11] Abhishek Das, Harsh Agrawal, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions? *arXiv:1606.03556 [cs]*, June 2016. arXiv: 1606.03556.

[12] Pranav Rajpurkar, Awni Y. Hannun, Masoumeh Haghpanahi, Codie Bourn, and Andrew Y. Ng. Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks. *arXiv:1707.01836 [cs]*, July 2017. arXiv: 1707.01836.

[13] Tharindu Fernando, Houman Ghaemmaghami, Simon Denman, Sridha Sridharan, Nayyar Hussain, and Clinton Fookes. Heart Sound Segmentation using Bidirectional LSTMs with Attention. *IEEE Journal of Biomedical and Health Informatics*, 24(6):1601–1609, June 2020. arXiv: 2004.03712.

[14] A. Almasi, M. B. Shamsollahi, and L. Senhadji. A dynamical model for generating synthetic Phonocardiogram signals. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5686–5689, Boston, MA, August 2011. IEEE.

[15] V. Kalaivani. DIAGNOSIS OF ARRHYTHMIA DISEASES USING HEART SOUNDS AND ECG SIGNALS. *Russian Journal of Cardiology*, (1-ENG):35–41, January 2014.

[16] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature Visualization. *Distill*, 2(11):10.23915/distill.00007, November 2017.

[17] M Finlay. The "mobile-phonocardiogram", a new tool in the arrhythmia clinic. *Heart*, 92(7):898–898, May 2006.

[18] Varsha Garg, Arpit Mathur, Nishant Mangla, and Aman Singh Rawat. Heart Rhythm Abnormality Detection from PCG Signal. In *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pages 1–5, Noida, India, August 2019. IEEE.

[19] L. Jordaens. 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, April 2018.

[20] Serkan Kiranyaz, Morteza Zabihi, Ali Bahrami Rad, Turker Ince, Ridha Hamila, and Moncef Gabbouj. Real-time phonocardiogram anomaly detection by adaptive 1D Convolutional Neural Networks. *Neurocomputing*, 411:291–301, October 2020.

[21] P. Lubaib and K.V. Ahammed Muneer. The Heart Defect Analysis Based on PCG Signals Using Pattern Recognition Techniques. *Procedia Technology*, 24:1024–1031, 2016.

[22] Lars-Jochen Thoms, Giuseppe Collichia, and Raimund Girwidz. Real-life physics: phonocardiography, electrocardiography, and audiometry with a smartphone. *Journal of Physics: Conference Series*, 1223:012007, May 2019.

[23] Preecha Yupapin, Wardkein, Preecha Yupapin, Phanphaisarn, Koseeyaporn, Roeksabutr, Roeksabutr, Wardkein, and Koseeyapon. Heart detection and diagnosis based on ECG and EPCG relationships. *Medical Devices: Evidence and Research*, page 133, August 2011.

[24] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. *arXiv:1311.2901 [cs]*, November 2013. arXiv: 1311.2901.

[25] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. *arXiv:1609.03499 [cs]*, September 2016. arXiv: 1609.03499.

[26] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise. *arXiv:1706.03825 [cs, stat]*, June 2017. arXiv: 1706.03825.

[27] Siddique Latif, Muhammad Usman, Rajib Rana, and Junaid Qadir. Phonocardiographic Sensing using Deep Learning for Abnormal Heartbeat Detection. *arXiv:1801.08322 [cs]*, July 2020. arXiv: 1801.08322.

[28] Stephanie Ger and Diego Klabjan. Autoencoders and Generative Adversarial Networks for Imbalanced Sequence Classification. *arXiv:1901.02514 [cs, stat]*, August 2020. arXiv: 1901.02514.

[29] Chaoqiang Zhao, Qiyu Sun, Chongzhen Zhang, Yang Tang, and Feng Qian. Monocular Depth Estimation Based On Deep Learning: An Overview. *Science China Technological Sciences*, 63(9):1612–1627, September 2020. arXiv: 2003.06620.

[30] Amy T. Dao. Wireless laptop-based phonocardiograph and diagnosis. *PeerJ*, 3:e1178, August 2015.

[31] Sumair Aziz, Muhammad Umar Khan, Majed Alhaisoni, Tallha Akram, and Muhammad Altaf. Phonocardiogram Signal Processing for Automatic Diagnosis of Congenital Heart Disorders through Fusion of Temporal and Cepstral Features. *Sensors*, 20(13):3790, July 2020.

[32] Md. Khayrul Bashar, Samarendra Dandapat, and Itsuo Kumazawa. Heart Abnormality Classification Using Phonocardiogram (PCG) Signals. In *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pages 336–340, Sarawak, Malaysia, December 2018. IEEE.

[33] Cristhian Potes, Saman Parvaneh, Asif Rahman, and Bryan Conroy. Ensemble of Feature:based and Deep learning:based Classifiers for Detection of Abnormal Heart Sounds. September 2016.

[34] S1 and S2 Heart Sound Recognition Using Deep Neural Networks. *IEEE Transactions on Biomedical Engineering*, 64(2):372–380, February 2017.

[35] Grzegorz Redlarski, Dawid Gradolewski, and Aleksander Palkowski. A System for Heart Sounds Classification. *PLoS ONE*, 9(11):e112673, November 2014.

[36] Chengyu Liu, David Springer, Qiao Li, Benjamin Moody, Ricardo Abad Juan, Francisco J Chorro, Francisco Castells, José Millet Roig, Ikaro Silva, Alistair E W Johnson, Zeeshan Syed, Samuel E Schmidt, Chrysa D Papadaniil, Leontios Hadjileontiadis, Hosein Naseri, Ali Moukadem, Alain Dieterlen, Christian Brandt, Hong Tang, Maryam Samieinasab, Mohammad Reza Samieinasab, Reza Sameni, Roger G Mark, and Gari D Clifford. An open access database for the evaluation of heart sound algorithms. *Physiological Measurement*, 37(12):2181–2213, December 2016.

[37] Ye Jia, Ron J. Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu. Direct speech-to-speech translation with a sequence-to-sequence model. *arXiv:1904.06037 [cs, eess]*, June 2019. arXiv: 1904.06037.

[38] Mohammed Nabih Ali, EL-Sayed A. El-Dahshan, and Ashraf H. Yahia. Denoising of Heart Sound Signals Using Discrete Wavelet Transform. *Circuits, Systems, and Signal Processing*, 36(11):4482–4497, November 2017.

[39] K. Ajay Babu, Barathram Ramkumar, and M. Sabarimalai Manikandan. S1 and S2 heart sound segmentation using variational mode decomposition. In *TENCON 2017 - 2017 IEEE Region 10 Conference*, pages 1629–1634, Penang, November 2017. IEEE.

[40] Wenjie Zhang, Jiqing Han, and Shiwen Deng. Heart sound classification based on scaled spectrogram and partial least squares regression. *Biomedical Signal Processing and Control*, 32:20–28, February 2017.

[41] M. Abo-Zahhad, Mohammed Farrag, Sherif N. Abbas, and Sabah M. Ahmed. A comparative approach between cepstral features for human authentication using heart sounds. *Signal, Image and Video Processing*, 10(5):843–851, July 2016.

[42] Elmar Messner, Matthias Zohrer, and Franz Pernkopf. Heart Sound Segmentation—An Event Detection Approach Using Deep Recurrent Neural Networks. *IEEE Transactions on Biomedical Engineering*, 65(9):1964–1974, September 2018.

[43] Patrick Esser, Robin Rombach, and Björn Ommer. Taming Transformers for High-Resolution Image Synthesis. *arXiv:2012.09841 [cs]*, February 2021. arXiv: 2012.09841.

[44] Photo Posters - Create Custom Photo Posters | Walgreens Photo.

[45] Ye Jia, Ron J. Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu. Direct speech-to-speech translation with a sequence-to-sequence model. *arXiv:1904.06037 [cs, eess]*, June 2019. arXiv: 1904.06037.