HIGH SCHOOL EDITION
Journal of Student Research

Volume 12 Issue 4 (2023)

Cardiac Auscultation Using Artificial Intelligence with Different Intake Methods

Nicholas Turner[1] and Sophia Barton[#]

[1]Regis Jesuit High School
[#]Advisor

ABSTRACT

Having to pay to go to a doctor's office and pay for a medical professional to use a stethoscope is costly and inconvenient. A mobile solution that is cheap and uses a medium that is widespread will make diagnoses more accessible. The objective of our research was to assess how feasible and accurate a mobile device solution to cardiac auscultation is, compared to a digital stethoscope. We used a convolutional neural network-based solution, which used the heart sound audio, collected with a digital stethoscope and smartphone, graphed out on a spectrogram for input. We trained two convolutional neural networks, one on the digital stethoscope audio and the other on the smartphone audio. To analyze the outputs, we used the metrics accuracy, recall, precision, and f1 score. We then compared the outputs of the model trained on the digital stethoscope audio versus the model trained on the smartphone audio. The model trained on the smartphone data typically performed 15% worse than the model trained on stethoscope data in terms of accuracy. Based off these results alone, the hardware technology in phones is still not advanced enough to reliably diagnose with machine learning.

Introduction

Chest pain can be from a multitude of different causes, and it is difficult to pinpoint precisely what the issue is as a person is going through the pain. The only way to tell if something is wrong is through the diagnosis process. Unfortunately, this process can be very time-consuming and expensive. Sometimes the help needed is not even available during a medical emergency. Having the instruments to diagnose without expertise, available via tech accessible to many people, would be an optimal solution. A piece of tech that fits the previous requirements would be smartphones. This paper focuses on how diagnosis using a mobile smartphone would stack up against a digital stethoscope using the same artificial intelligence model. Diagnosis is inherently a classification problem, as the action of diagnosis is classifying one's symptoms into a certain disease or no disease. The data that is given would be heart sounds (these sounds are very similar to what can be heard through a traditional stethoscope), the nature of which is audio data but will be transformed into visual data (as a spectrogram). Once through the model, the output would be represented in binary labels: irregular and regular.

Background

The 2018 Kang et al published paper focuses on the feasibility of cardiac auscultation using a smartphone with no additional attachments for audio collection. They created a smartphone app called CPstethoscope, which used a built-in microphone to collect heart sounds by pressing the bottom of a smartphone against the skin of the patient. This research was insightful in learning the most optimal way to collect heart audio data and if it is feasible to do so with only the built-in microphone. The conclusion that they drew was that cardiac auscultation diagnosis was feasible. This meant that the study on the margin of accuracy between a digital stethoscope and a smartphone was also attainable.

The Deep Learning Methods for Heart Sounds Classification: A systematic Review researched the methods of classification of heart sounds. The idea of using a convolutional neural network (CNN) for our method of classification was originally from this article. The most prominent methods of feature extraction mentioned in the article are MFSC, MFCC, and spectrograms. We decided to use the spectrogram because of this article.

Dataset

The dataset that was used in the project was sourced from Kaggle from a machine learning challenge data set called Heartbeat Sounds. Two folders contain heart sound audio files: set_a and set_b, in the form of WAV files. Set_a audio was recorded on an iPhone using the iStethescope pro app. This data was gathered from the general public. Set_b audio was recorded using the digital stethoscope DigiScope from a clinical trial. There are also three CSV files that go along with the folders: set_a.csv, set_b.csv, and set_a_timing.csv. For set_a, the labels include artifact, extrahls, murmur, and normal. The distribution of the labels in set_a can be seen in figure 1. For set_b, the labels include extrasystole, murmur, and normal. The distribution of the labels in set_b can be seen in figure 2.

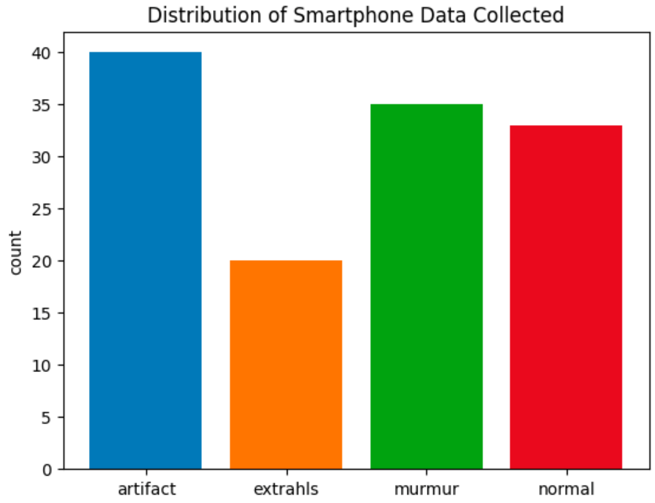


Figure 1. This graph shows the distribution of the types of heart diagnoses of the audio collected by the smartphone.

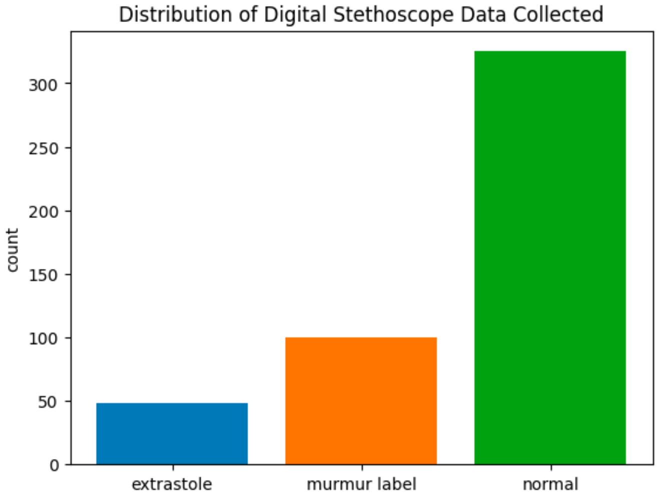


Figure 2. Graph shows the distribution of the types of heart diagnoses of the audio collected by digital stethoscope.

The categorical labels were then changed into numerical labels. These labels were then transformed into 1 for abnormal rhythm and 0 for normal rhythm. The number of audio files that were collected by a digital stethoscope greatly outnumbered those collected by a smartphone, so reducing the size of set_b was necessary to avoid bias in our model with a more even distribution of input data.

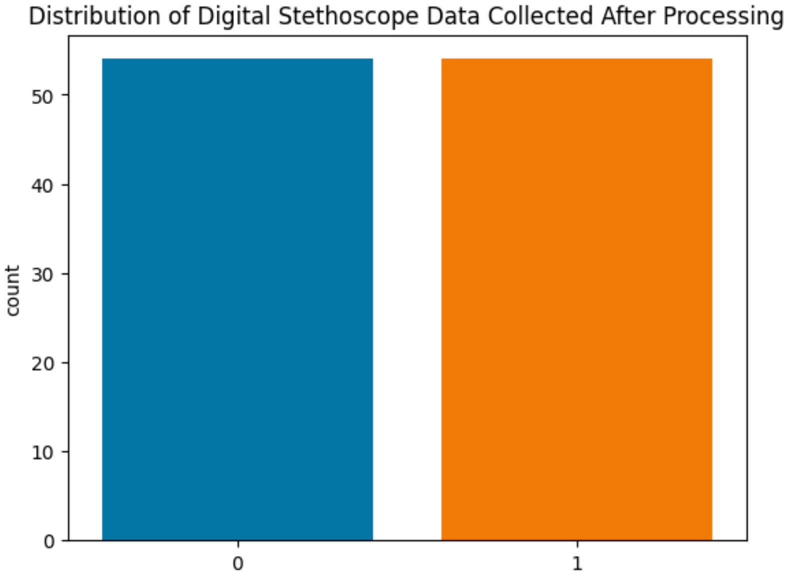


Figure 3. This graph shows the distribution of the normal audio (0) and the abnormal audio (1) from the digital stethoscope data after it was pared and bucketized.

The data was then split into train and test sets. The train set (for training the machine learning model) and the test set (for gathering metrics on the performance in the model). 40% of the data was used in the test set, while the remaining 60% was used in the train sets.

For displaying the audio WAV files and to be used as the input feature in our AI model, the method chosen was a spectrogram, such as the one shown in figure 4.

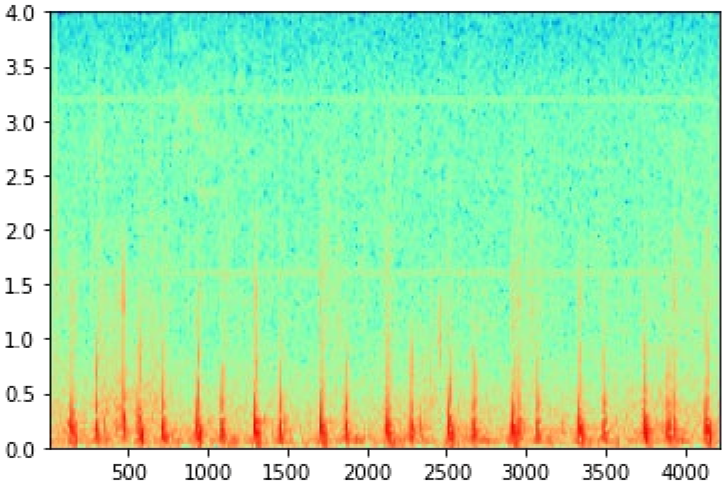
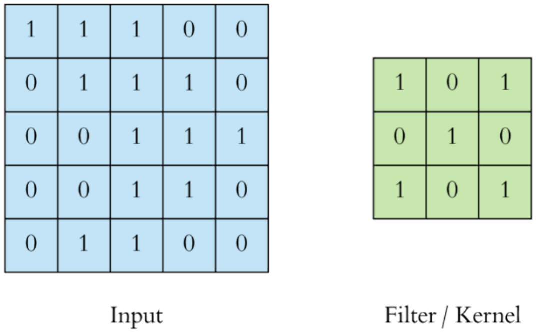


Figure 4. An example of a spectrogram taken from the smartphone audio data.

A spectrogram displays audio data using the x-axis, y-axis, and color. The x-axis represents time, the y-axis represents frequency, and the brightness of the color indicates the decibels of the frequency at that point in time.

Methodology/Models

The model used is a CNN (convolutional neural network) from the python library sklearn and keras. CNNs are particularly talented at reading and interpreting visual data. This makes it perfect for reading our data, as the input spectrograms are visual data. CNNs work by applying different filters over an image for feature extraction, and then using the output of all the filters for the input of a dense layer for classification. The input for a CNN consists of an array containing pixel data in the form of 3 numbers 0-255 (if the image is colored) representing the brightness of the color of that pixel. The way a filter is applied for feature extraction is by using a kernel as shown in figure 5.



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

Figure 5. An example of a kernel used in a convolutional neural network.

This kernel will slide across the input and perform an elementwise multiplication operation on each entry in the input. After the kernel reaches the end, the output will have the filter applied. CNNs are not limited to one layer of feature extraction. There can be many layers and many kernels. Between the convolutional layers, there can be pooling layers that reduce the size of each image. This is done to make classification more efficient and prevent overfitting to the training data. There are two types of pooling: max pooling and average pooling. Max pooling returns the maximum value of an area, while average pooling returns the minimum. This number is then plotted in the output. After all the pooling layers and convolution layers, the final image is then used as the input for a small dense network used for classifying. This dense network uses values called weights and biases to determine an output based on an input.

The CNN model has settings, called hyperparameters, that can be tuned to better fit the data. These settings are not learned automatically through training. The three parameters that were tuned on the model were learning rate(lr), epochs, and learning rate decay (or just decay). The learning rate assigns an error to each weight. This affects how fast the model "learns" the inputs. Learning rate decay slowly reduces the learning rate until it reaches the most optimal value. Learning rate and decay are implemented to avoid overfitting to the training data. This means that the model has "memorized" the training data and answers. This would be similar to when a student memorizes a practice test but fails the real test because they only memorized the answers and not the method to find those answers. Epochs are the number of rounds the model will work through the dataset. The way that these hyperparameters are tuned and chosen is by trial and error. This can be seen in table 1. By testing different values and evaluating the accuracy results, the tuned values we arrived at that achieved the highest results are lr=0.001, epochs=10, and decay=1e-6.

Table 1. Table shows the result of testing different values of different hyperparameters and comparing the accuracy.

Hyperparameter	Value 1	Result 1	Value 2	Result 2	Value 3	Result 3
Epochs	Epochs = 5	67%	Epochs = 10	82%	Epochs = 15	82%
LR	LR = 0.001	82%	LR = 0.005	71%	LR = 0.0001	61%
Decay	D = 1e-6	82%	D = 1e-7	76%	D = 1e-5	80%

The CNN model used consisted of a single convolutional layer with a kernel size of 32x32 with a stride of 3x3, taking in an input shape of 288x432x4 (the size of the spectrograms created in preprocessing). After the convolutional layer, there is a max pooling layer with a pool size of 2x2, then a dropout layer with a rate of 0.15. Finally, there are two dense layers used for classifying. The first contains 512 nodes, then the last contains two nodes, which are used as an output. One node signifies an abnormal heartbeat, while the other represents a normal heartbeat. The final configuration for the CNN was realized through trial and error. Any other tested configuration would drop the accuracy score by 15-20%. These tested configurations include adding more convolution layers, increasing and decreasing the shapes of the kernels and the strides, and changing the number of dense layers.

Results

The metrics of the model trained on heart sounds collected from a digital stethoscope performed significantly higher than the model trained on heart sounds collected from a smartphone. We used four different metrics to assess the performance of each model: accuracy, recall, precision, and f1 score. These metrics are obtained through different calculations using the tp, tn, fp, and fn of a model's predictions. Tp or true positive is the number of positive labels the model correctly predicted. In the case of the heart data, true positive represents how many abnormal heartbeats the model correctly predicted. Tn or true negative is the number of non-positive labels the model predicted correctly. With the heart data, tn represents how many normal heartbeats the model guessed correctly. Fp and fn stand for false positive and false negative. They are similar to tp, and tn except the model guessed them incorrectly. These values can be plotted using a confusion matrix. A confusion matrix created out of tp, tn, fp, and fn labels from the smartphone model are shown in figure 6.

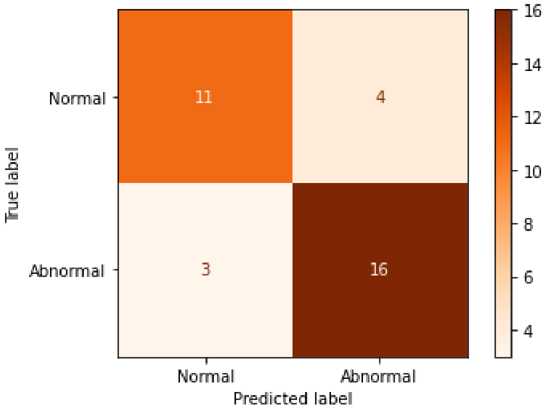


Figure 6. An example of a confusion matrix that uses tp, tn, fp, and fn.

With these values we calculated the metrics needed for evaluating the model. Accuracy is calculated using this formula: $\frac{tp+tn}{tp+tn+fp+fn}$. Accuracy is used to measure the percentage of correct predictions out of the total number of predictions. Precision is calculated with this formula: $\frac{tp}{tp+fp}$. In this case precision is used to measure how many abnormal heart beats were classified correctly out of all the predictions classified as abnormal heart beats. Recall is calculated utilizing this formula: $\frac{tp}{tp + fn}$. Recall is useful as it evaluates how many abnormal heartbeats were classified correctly out of all the actual abnormal heartbeats. The last metric used, f1, is calculated using this formula: $F1 = 2 \times \frac{precision \times recall}{precision + recall}$. F1 allows us to quickly judge the recall and precision using only 1 number. We gathered metrics for each model over five trials. Each trial included randomly shuffling the data from the test set and the train set. The final metrics for the smartphone-trained model are shown in table 2, and the digital stethoscope-trained model is shown in table 3.

Table 2. The metrics of the convolutional neural network trained on the smartphone data over five trials.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	AVG
Accuracy	79%	79%	65%	82%	82%	77.4%
Recall	80%	88%	64%	76%	76%	76.8%
Precision	84%	84%	84%	100%	100%	90.4%
F1	82%	86%	72%	86%	86%	82.4%

Table 3. The metrics of the convolutional neural network trained on the digital stethoscope data over five trials.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	AVG
Accuracy	95%	95%	95%	86%	91%	92.4%
Recall	100%	100%	100%	100%	82%	96.4%
Precision	91%	86%	90%	63%	100%	86.0%
F1	95%	92%	95%	77%	90%	89.8%

Overall, the model trained on the digital stethoscope performed better than the model trained on smartphone data. On average, accuracy was 15% higher, recall was 19.6% higher, and f1 was 7.4% higher. The digital stethoscope model performed slightly worse in terms of precision compared to the smartphone model with a precision 4.4% less than the smartphone model.

Discussion

From these metrics, we can establish that the smartphone model was able to distinguish abnormal heartbeats fairly accurately. Where the model ran into issues was with discerning normal heartbeats. This was still apparent in the digital stethoscope model, but the gap between the accuracy of classifying abnormal and normal sounds was smaller than the gap in accuracy for the smartphone model. There are a couple of reasons why this could be occurring. The smartphone could be picking up surrounding noise due to its less specific use case. Smartphone microphones are meant to pick up a variety of sounds, while stethoscopes are manufactured for one purpose. The smartphone could pick up noises from the environment, and the model could interpret that as an abnormality of the heart. The second possibility is the distribution of abnormal heartbeats and normal heartbeats. As shown in figure 1 and figure 2, the stethoscope data has many more normal sounds compared to the smartphone data. The paring of the data, which was explained in the preprocessing section, was thought to have fixed this, but the way the data was pared was through random selection. This is a problem as each time the program was run, there is a possibility that the digital stethoscope model got more normal heart data than the smartphone model, which increased the variability of the metrics in the digital stethoscope model.

Conclusion

Having a smartphone that is able to record heart audio data and classify that data reliably and accurately would be able to aid many people. The technology would make the diagnosis process less stressful, more efficient, widely available, and less expensive.

The purpose of this research was not to create this model that would be put to use but to evaluate how well a certain configuration of a machine learning model using data acquired from a smartphone compares to another model of the same configuration using data collected using a digital stethoscope. The stethoscope model outperformed the smartphone model in terms of accuracy, recall, and f1 but interestingly the smartphone model outperformed the digital stethoscope model in terms of precision. This means that the smartphone model is particularly skilled at classifying abnormal heartbeats. Future research endeavors could investigate a different approach for feature extraction, such as mfcc's instead of spectrograms, a way of filtering out environmental sound from the smartphone data, and a way of regularizing the data that evens the ratio of normal heart audio and abnormal heart audio the same between the smartphone data and the stethoscope data.

Limitations

The dataset used in the research does not mention where on the body the smartphone and the digital stethoscope were placed when collecting the heart audio. This is important as some heart disorders can only be heard from certain locations. This is why when doctors are diagnosis using a stethoscope, they will place the stethoscope on various locations of the chest as seen in figure 7.

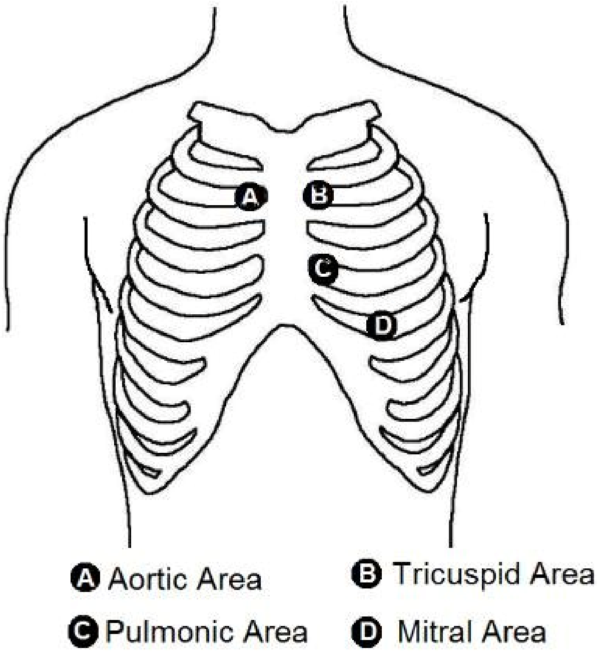


Figure 7. The different locations that a stethoscope is placed during cardiac auscultation.

This process gives the most accurate diagnosis possible. The heart audio used in the study only gives audio taken from one unnamed point on the chest. I predict that the metrics of both models used in the study would greatly increase if the dataset included audio taken from multiple points on the chest.

The type of phone and microphone used to take the audio data is not presented in the dataset. Different types of phones have different qualities of microphones. Higher quality microphones may pick up more minute details than lower quality microphones. These details may denote an abnormality which only a higher quality microphone would be able to detect. Gathering metrics across a variety of smartphones would give metrics that are closer to the results that would be found in everyday smartphones.

Smartphone microphones are unable to separate background noise from the subject of the audio. This results in audio that has increased levels of background noise. This is a problem as the algorithm could mistake the background noise as very noisy blood flow. Very noisy blood flow could denote defects inside the heart which means that the algorithm would misinterpret the audio as abnormal. Using a denoising algorithm as a part of pre-processing the dataset would remove a majority of the background noise. This would solve the problem of the misinterpretation of the background noise, but it would also present another problem; the denoising algorithm could remove the sound of the blood flow of the heart. Listening to blood flow is a very important part of cardiac auscultation. Without it, many diagnoses would be missed.

Acknowledgments

I would like to thank Inspirit AI for the resources available to me and Sophia Barton for supervising the entire research process.

References

[1] *Achieving better voice quality: Why smartphones need 3 microphones*. (2013, September 9). Embedded. Retrieved November 30, 2022, from https://www.embedded.com/achieving-better-voice-quality-why-smartphones-need-3-microphones/

[2] *Audio recording with a smartphone*. (2021, September 20). Wild Mountain Echoes. Retrieved November 21, 2022, from https://www.wildmountainechoes.com/equipment/audio-recording-with-a-smartphone/

[3] Chen, W., Sun, Q., Chen, X., Xie, G., Wu, H., & Xu, C. (2021). Deep learning methods for heart sounds classification: A systematic review. *Entropy*, *23*(6). https://doi.org/10.3390/e23060667

[4] Derek, T. (n.d.). *How to use a stethescope*. Caregiverology. Retrieved November 22, 2022, from https://www.caregiverology.com/stethoscope.html#gallery[pageGallery]/0/

[5] Güven, M., Hardalaç, F., Özışık, K., & Tuna, F. (2021). Heart diseases diagnose via mobile application. *Applied Sciences*, *11*(5). https://doi.org/10.3390/app11052430

[6] Saha, S. (2018, December 15). *Web page*. Towards Data Science. Retrieved October 18, 2022, from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[7] Shervegar, M. V., & Bhat, G. V. (2018). Heart sound classification using gaussian mixture model. *Porto Biomedical Journal*, *3*(1). https://doi.org/10.1016/j.pbj.0000000000000004

[8] Si-Hyuck, K., Byunggill, J., Yeonyee, Y., Goo-Yeong, C., Insik, S., & Jung-Won, S. (2018). Cardiac auscultation using smartphones: Pilot study. *JMIR Publications*, *6*(2). https://doi.org/10.2196/mhealth.8946