## Introduction

- Heart disease is the leading cause of death globally, causing more than seventeen million deaths each year.
- Lack of regular heart screening due to limited healthcare availability, unhealthy lifestyles, or other factors can lead to an increased risk of cardiovascular diseases (CVDs). Early diagnosis of CVDs is crucial in reducing the risk of heart failure.
- Doctors perform a process called auscultation to diagnose potential heart problems. This involves using a stethoscope to diagnose heart sounds as normal or abnormal.
- However, this method can only by performed by a doctor in a clinical setting, as most people do not have the skills or equipment to perform this themselves. This is where machine learning comes in.
- Using a machine learning model, people will be able to perform auscultation from the comfort of their homes through a smartphone app. By learning patterns and trends in heart sound data, the model will be able to accurately diagnose the user's heart sounds. This can aid in early detection and prevention of heart problems.

## Background

Heart sounds are caused by the opening and closing of certain valves in the heart and can provide valuable information about the health of the heart. Normal heart sounds include the S1 and S2 sounds (often called "lub" and "dub", respectively). These sounds are typically regular and evenly spaced. Abnormal heart sounds, which are typically murmurs, indicate a potential problem with the heart. They can be caused by valve disorders like mitral regurgitation, heart defects like ventricular septal defect, and other conditions, such as hypertension.

Closure of these valves cause the S2 or "dub" sound

Closure of these valves cause the S1 or "lub" sound

Figure 1. Diagram of the human heart From eastcoastheartvet.ca

# **Engineering Goal**

- The goals of this project are to:
  - Create a robust machine learning model trained on a publicly available heart sounds dataset
  - Create an accompanying **smartphone app** that can accurately classify normal and abnormal heart sounds from user-recorded or uploaded audio.



# **Training Data**

• I used the publicly available PhysioNet heart sounds dataset at https://physionet.org/content/challenge-2016/1.0.0/, which contains thousands of labeled recordings of heart sounds from both healthy individuals and patients with various heart conditions taken in clinical and nonclinical settings.

### Procedure

#### **1. Data preprocessing**

I first split the audio files into five second segments, then used the Python Librosa library to transform each segment into Mel Frequency Cepstral Coefficients, or MFCC. MFCCs are a set of features extracted from an audio file. Figure 2, below, shows the Mel Spectrograms and MFCCs for normal and abnormal audio files.







Figure 2. Visualizations of abnormal (top) v. normal (bottom) heart sounds

#### 2. Creating the model

• I used a Convolutional Neural Network (CNN) for this project because it excels at image processing. By treating the MFCC features as an image, I can leverage this for higher accuracy. The full structure of my CNN can be seen in Figure 3 below.

	onv2d_8_input		mpu	.:	[(None, 40, 98, 1)]						
L	InputLayer		Julipt		[(140ffe, 40, 96, 1)]						
<b>↓</b>											
	conv2d_8	in	input:		(None, 40, 98, 1)						
	Conv2D		output:		(None, 39, 97, 16)						
1	dropout 8	ir	V V	(None 39 97 16)							
	Dropout	Dropout ou		(None, 39, 97, 16)							
I	Diopout	00	loput.	(110116, 33, 37, 10)							
	conv2d_9	v2d_9 input:		(None, 39, 97, 16)							
	Conv2D	output:		(None, 38, 96, 32)							
ma	max pooling2d 3 in			ıt:	(None, 38, 96, 32)						
N	/laxPooling21	 D	) outp		(None, 19, 48, 32)						
▼											
	dropout_9 input:		(None, 19, 48, 32)								
	Dropout o		itput:	(None, 19, 48, 32)							
[	conv2d_10	v2d_10   input:		(None, 19, 48, 32)							
	Conv2D	output		(None, 18, 47, 64)							
⊢	Dress and another the sector of the sector o		nput:	(None, 18, 47, 64)							
Dropout output: (None, 18, 47, 64)											
<b>↓</b>											
[	conv2d_11 input:			(None, 18, 47, 64)							
	Conv2D	output:		(None, 17, 46, 64)							
Г	dropout 11	i	nput:		None, 17, 46, 64)						
ŀ	Dropout		output:		(None, 17, 46, 64)						
L	Dropour	10		1.1							
				·							
	conv2d_12	v2d_12 input:		(None, 17, 46, 64)							
	Conv2D	01	itput:	(N	Ione, 16, 45, 64)						

		1	3								
max pooling2d	4	inpu	t:	(Non	e, 16, 4	15, 64)					
MaxPooling2I	5	outpu		(Nor	ne, 8, 2	2, 64)					
[]	_	+		NT	0 22 4	642					
dropout_12	1	input:		(None, 8, 22, 64)							
Dropout		Julput:	1	(None, 8, 22, 64)							
		↓	ġ.								
conv2d_13	i	input:		(None, 8, 22, 64)							
Conv2D	0	output:		(None, 7, 21, 64)							
		1									
dropout 13		input:		(None, 7, 21, 64)							
Dropout	6	output:		None,	64)						
	<b>.</b>	<del>\</del>	10		7 21 6						
conv2d_14	1	input:		None,	7, 21, 6	<u>(4)</u>					
Conv2D	0	output:		None,	6, 20, 6	94)					
		↓									
dropout_14	i	input:		(None, 6, 20, 64)							
Dropout	0	output:		(None, 6, 20, 64)							
conv2d_15	ir	input:		(None, 6, 20, 64)							
Conv2D	ou	output:		(None, 5, 19, 128)							
man naalin a2d	=	<b>*</b>		(NIer	- 5 10	120					
MaxPooling2U		outru		t: (None 2 9 128		128)					
Maxi bolilig21		Jourbe			10, 2, 3	, 120)					
8 <u></u>			ā								
dropout_15	1	(	(None, 2, 9, 128)								
Dropout output: (None, 2, 9, 128)											
		1									
global average pool	ling	2d 1	ir	put:	(None	, 2, 9, 128)					
GlobalAveragePool	ling	ou	output: (None, 128)								
donce	in rute		(None	128)							
Dense_1	-	output		(None 2)							
Dense output. (Rone, 2)											

Figure 3. Model architecture

# HeartNN: **A High Accuracy Neural Network** for Cardiac Auscultation

## **Training the Model**

- Before training the model, I split the PhysioNet dataset into 80% training data and 20% testing data.
- Figure 4, below, shows the progression of the model over 500 epochs. The four lines depicted in the graph are:
  - **Training loss:** This line shows how the model's loss (or error) decreases over time as it is trained on the training set.
  - **Training accuracy**: This line shows how the model's accuracy increases over time as it is trained on the training set.
  - Validation loss: This line shows how the model's loss on the validation set changes over time and is used to evaluate how well the model is generalizing to new data.
  - Validation accuracy: This line shows how the model's accuracy on the validation set changes over time and is also used to evaluate how well the model is generalizing to new data.



All images and graphics were created by the researcher unless otherwise noted.

## Results

• Figure 5, below, shows the confusion matrix that resulted from the evaluation of the model. A confusion matrix is used to show the number of testing samples the model predicted correctly and incorrectly, as well as which classes they were. \*I consider a normal label to be positive in this work.



Figure 5. Confusion matrix

• Using the values in the confusion matrix, we can calculate several statistics for the model:

- Accuracy: 97.59%
- **Precision**: 99.48%
- Recall or Sensitivity: 97.37%
- **F1 Score**: 98.41%

• Challenge score of the model exceeds that of the winning entry of the 2016 PhysioNet challenge, whose dataset I used to train the model.

# **Smartphone App**

• After training the model, I hosted it on a cloud server. • I then created a smartphone app in Android Studio that uses on the cloud server to diagnose the user's recordings. To use the app, the user can load a recording of their heart sound onto the app and view its diagnosis. Figure 6. shows this process.



Figure 6 : Integration of ML model on the cloud server with smartphone app

## **Conclusion and Future**

• Overall, I used the PhysioNet dataset to train a highaccuracy neural network, and then deployed it on a smartphone app for anyone to use.

• In the future, I plan to add the recording feature within the app, reminders for regular heart sound testing, and the ability to share results with healthcare professionals.