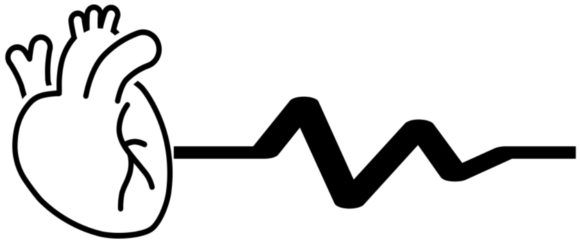
The Iby and Aladar Fleischman
Faculty of Engineering
Tel Aviv University



Classifying Abnormal Heart Rate

1. The problem and our motivation
2. Data, EDA, Audio preprocessing
3. Fourier, STFT, Wavelet, Spectrogram
4. Classes, Augmentations and Model
5. Performance analysis and possible improvements

Introduction:

The goal of this project is to build a deep learning tool that can classify the rhythm of a heartbeat as normal or abnormal (murmur or extrhs).

Tools:

The project will combine signal processing, image processing, neural networks, convolutions, and transfer learning.

Motivation:

Creating a model to classify abnormal heart rhythms has the potential to greatly impact healthcare. Accurate diagnosis of arrhythmias is crucial for effective treatment.

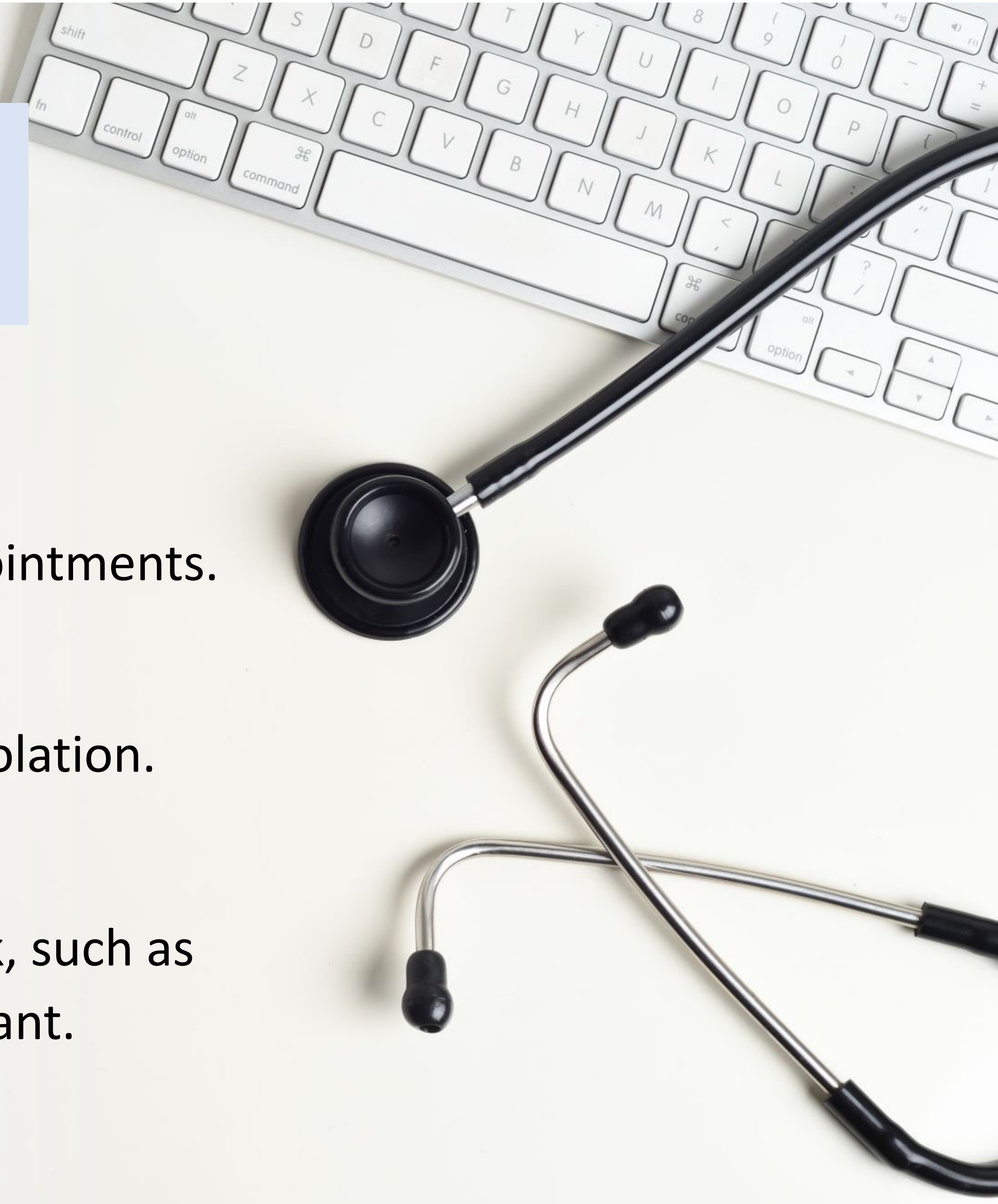
Value:

Our model aims to enhance diagnostics, aiding medical professionals in making life-saving decisions.

Why do we need this method

The benefits of such an app include:

- Elderly persons who are unable to attend appointments.
- Reducing the strain on the health-care system.
- Pandemics - offering distant heart health consolation.
- Following up on heart patients.
- Giving consultations to patients who are at risk, such as those who are immunocompromised or pregnant.

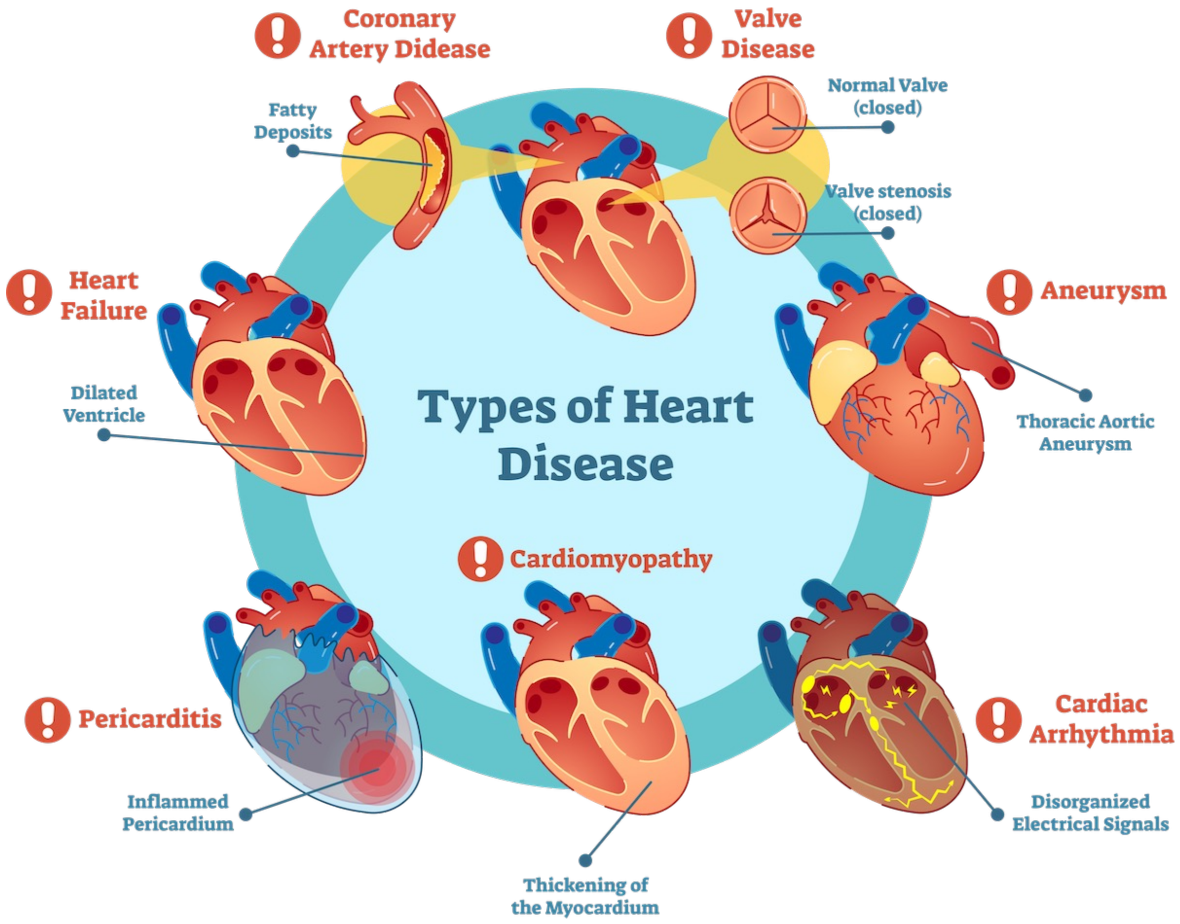


Data set from Pascal challenge 2012

Source: iStethoscope Pro iPhone app

Size: 124 + 52 for test

Classes: Normal, Murmur, Extra Heart Sound, Artifact



Heart Conditions

Electrical | Vascular | congenital | valvular

- Murmur can be normal but also can identify a heart diesis.
- Extra sound is more common in valve prolapse.
- Most heart conditions can show both symptoms.

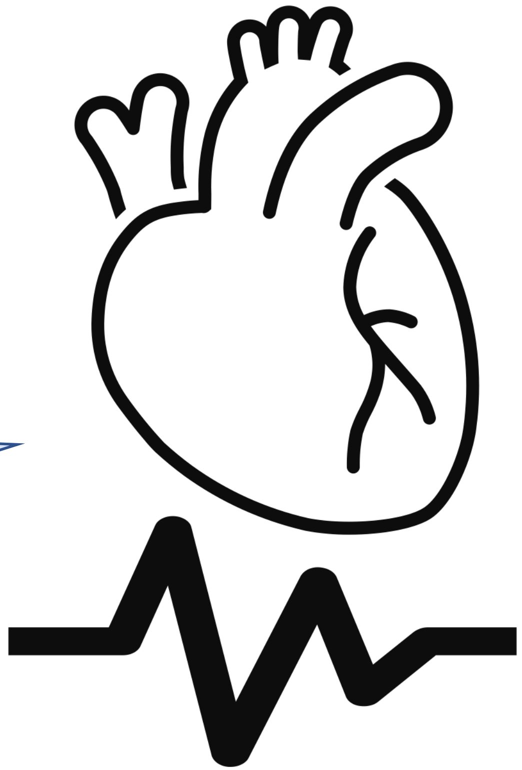
Heartbeat Categories

Normal

Healthy heart has a distinct lub dub auditory pattern with a good rhythm and the heartbeats are fairly strong

Extra heart sound

Extra heart sounds are identified by an additional beat. The heartbeats sound as though there is a “galloping noise”



Murmurs

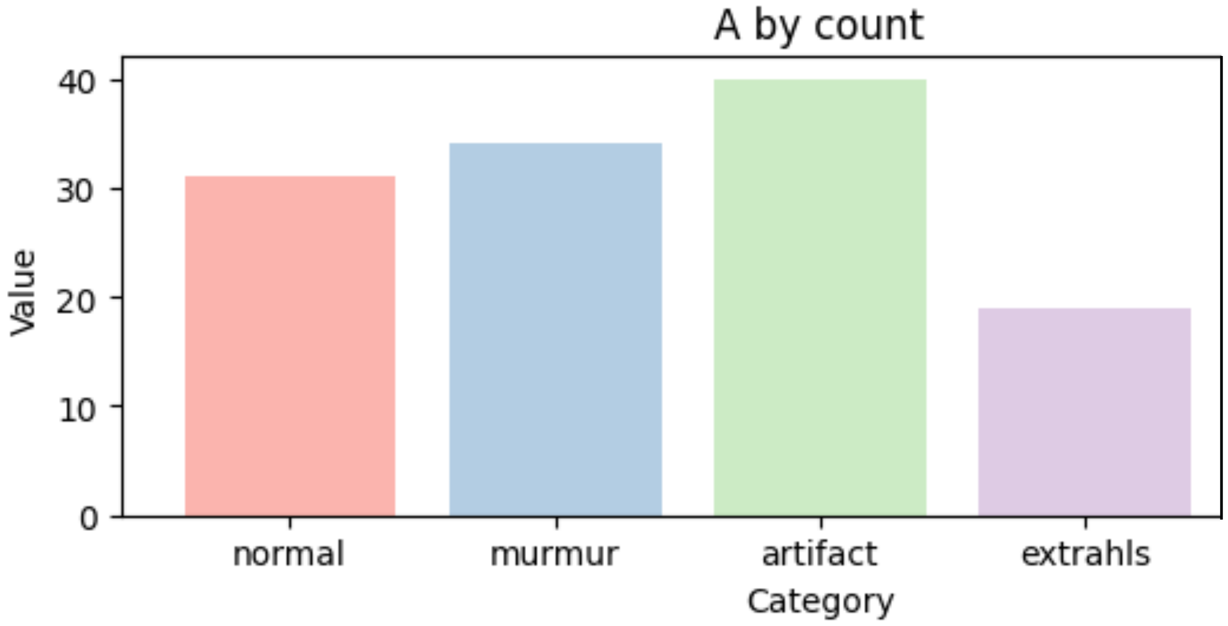
This sounds as if there is a “whooshing, roaring, rumbling or turbulent fluid” in one of the temporal locations S1 or S2

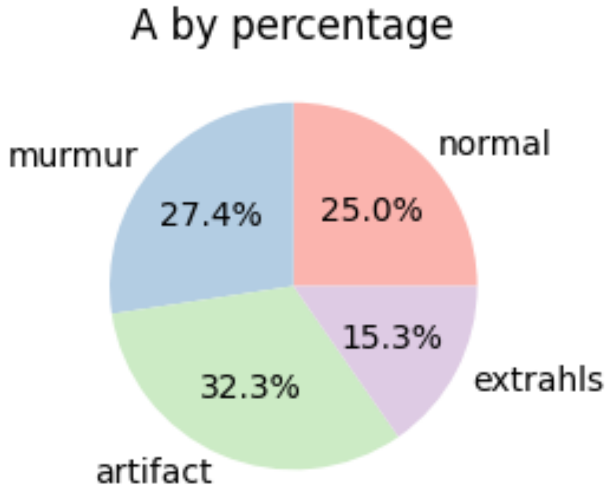
Artifacts

Characterized by wide range of different sounds (feedback squeals and echoes, speech, music and noise). Basically, not a heartbeat

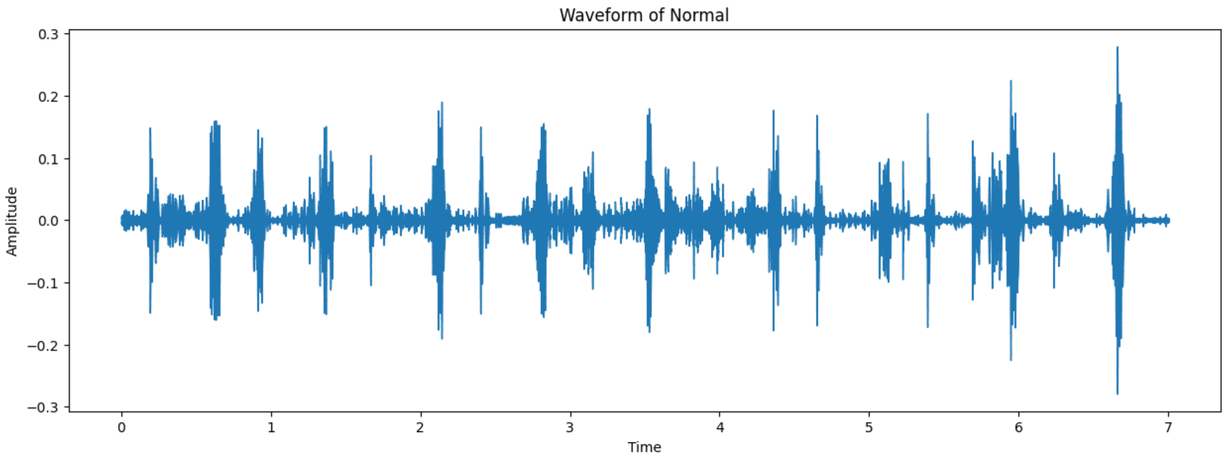
Exploratory Data Analysis – Category Distributions

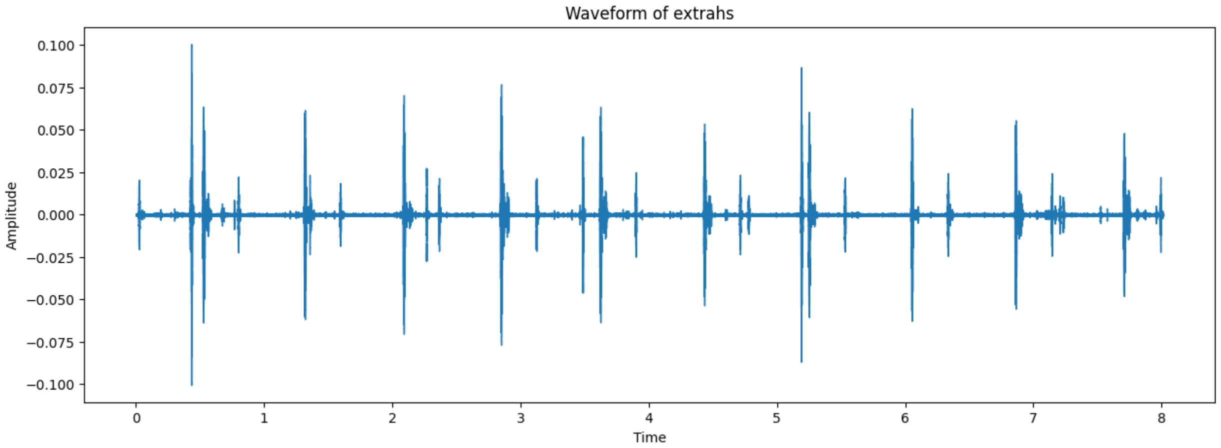
Train Date - A

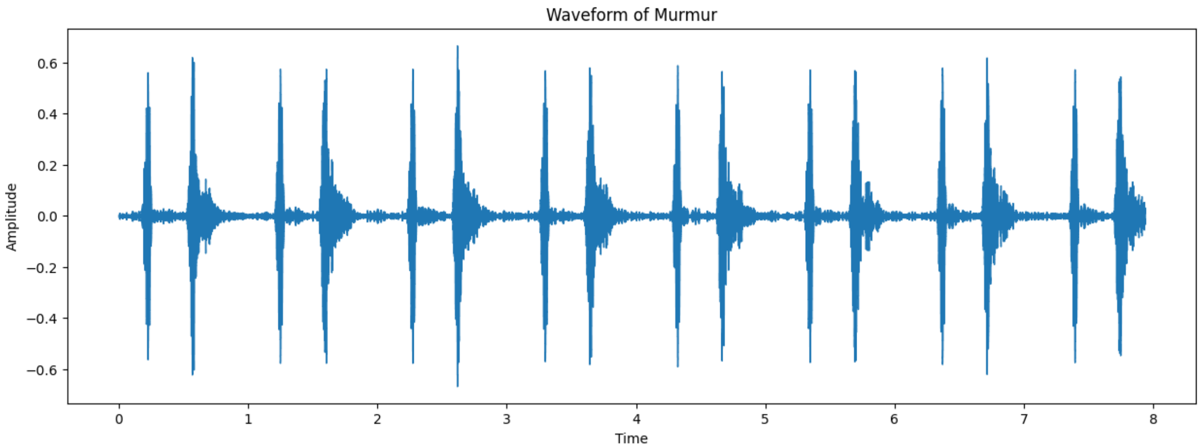


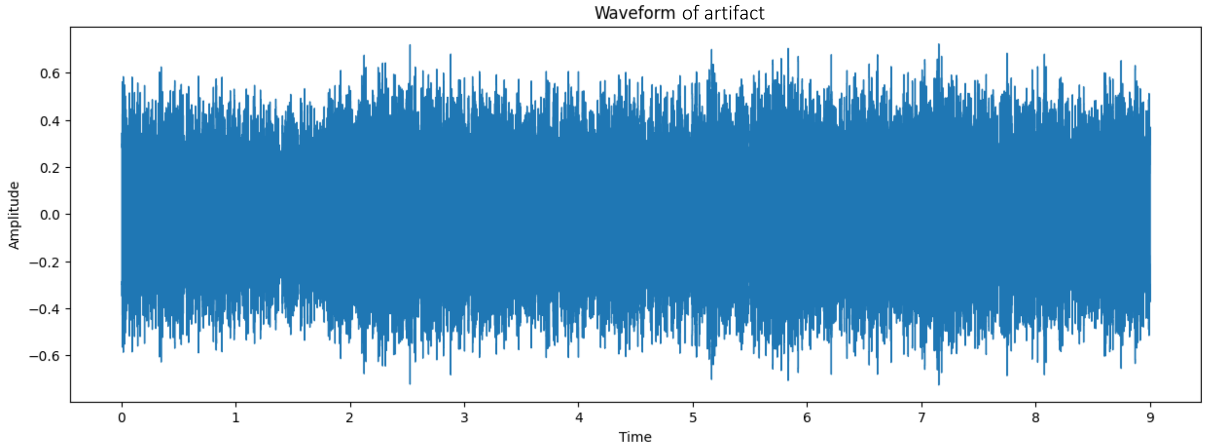


Exploratory Data Analysis – Heartbeat Visualization

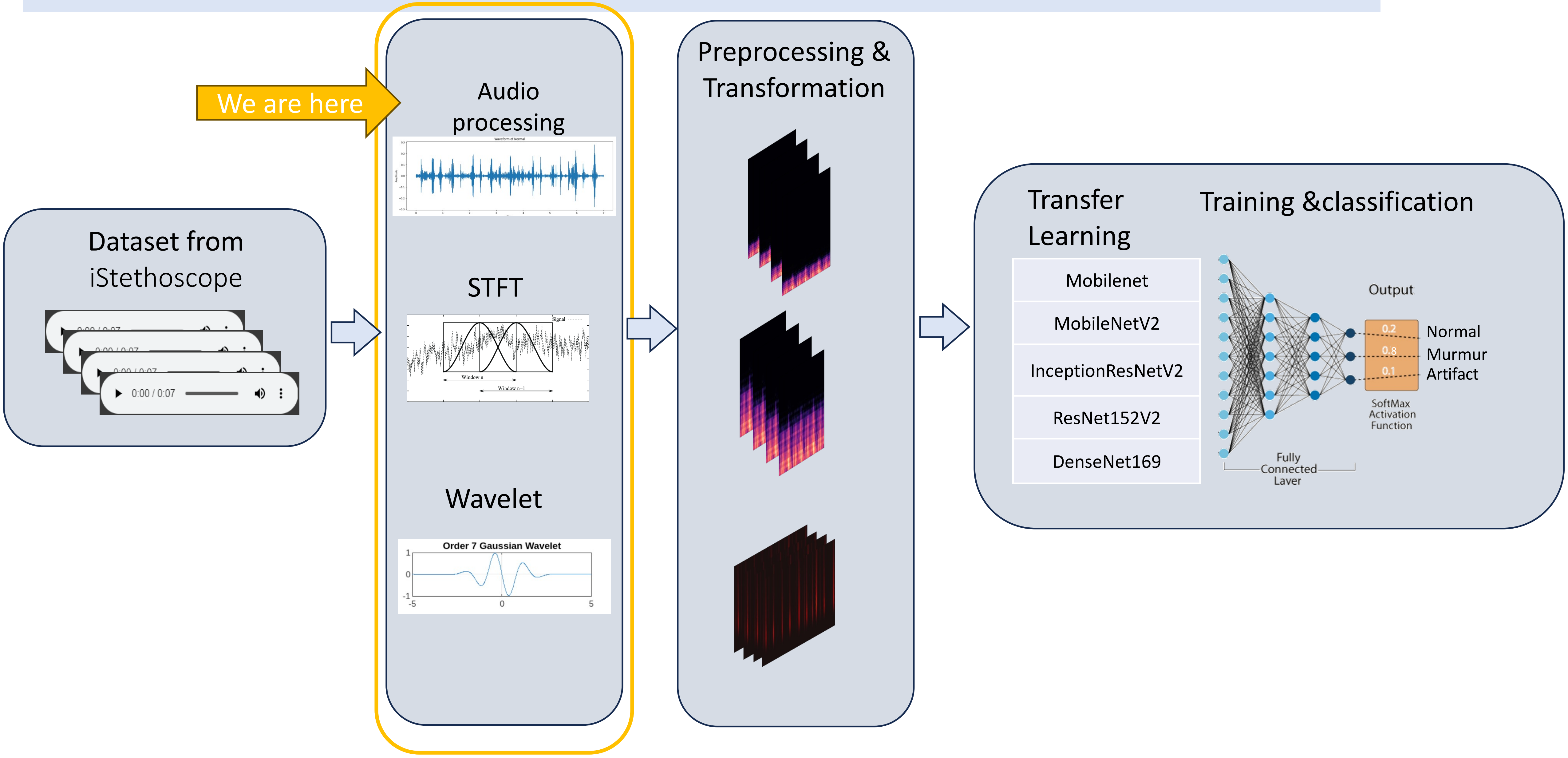








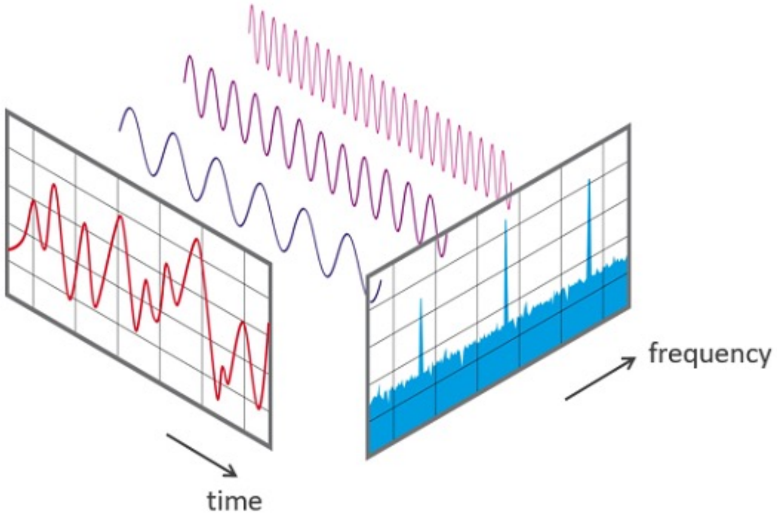
Our Process:



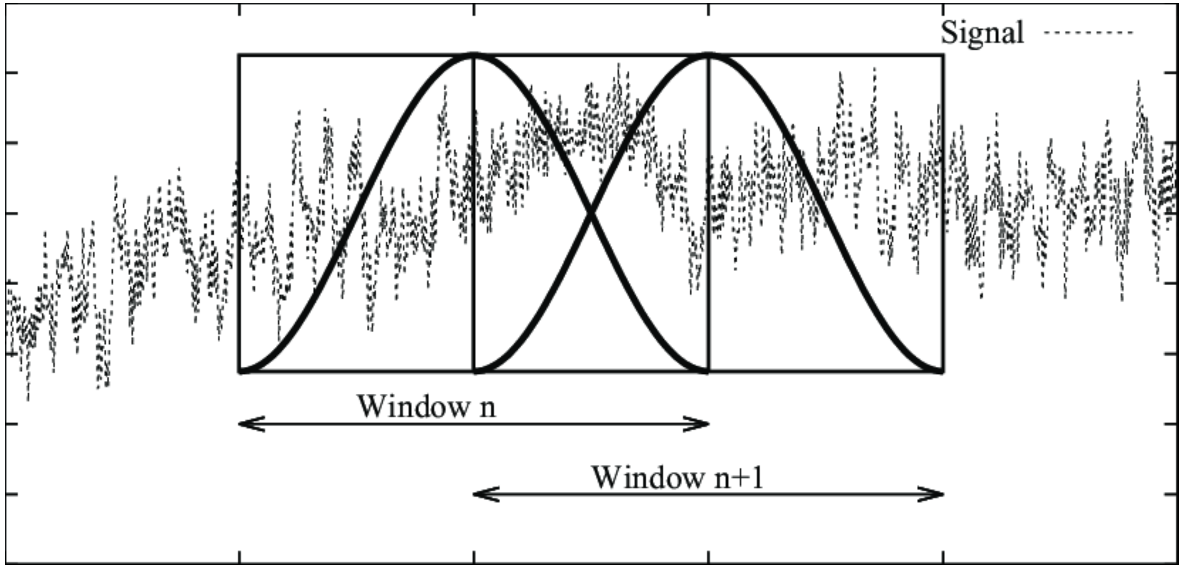
We are here
Dataset from iStethoscope
0:00 / 0:07
Audio processing
STFT
Window n
Window n+1
Signal
Wavelet
Order 7 Gaussian Wavelet
Preprocessing & Transformation
Transfer Learning
Mobilenet
MobileNetV2
InceptionResNetV2
ResNet152V2
DenseNet169
Training &classification
Output
0.2
0.8
0.1
Normal
Murmur
Artifact
SoftMax Activation Function
Fully Connected Layer

Signal Processing

- Furrier Transform

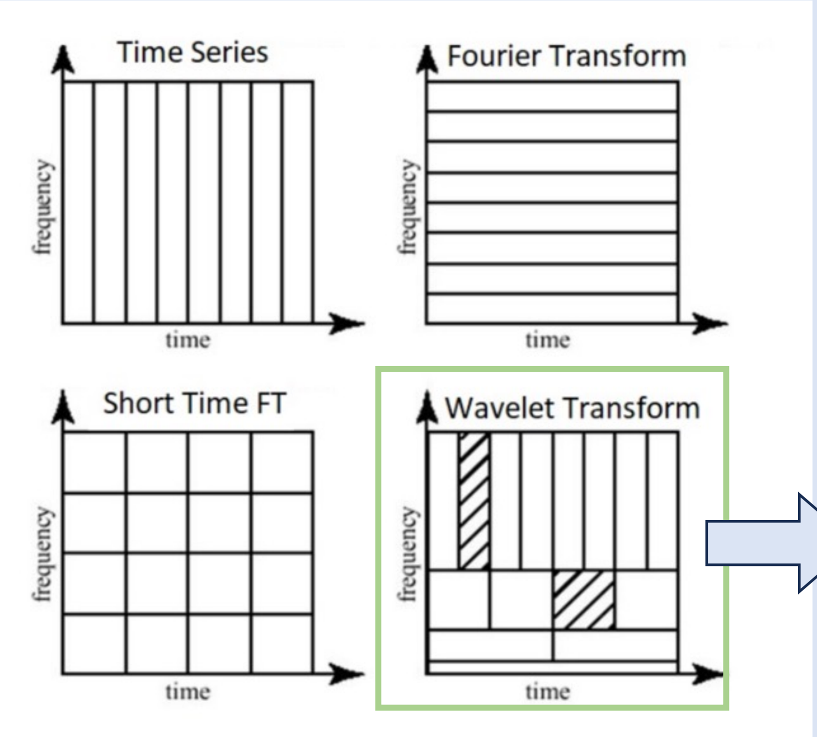


- Short Time Furrier

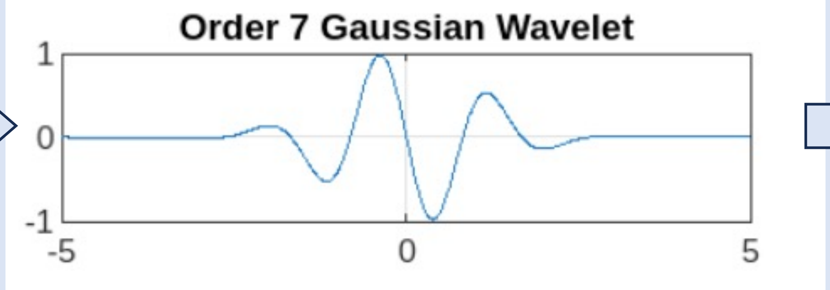


We chose STFT
because time-localized frequency information

Wavelets



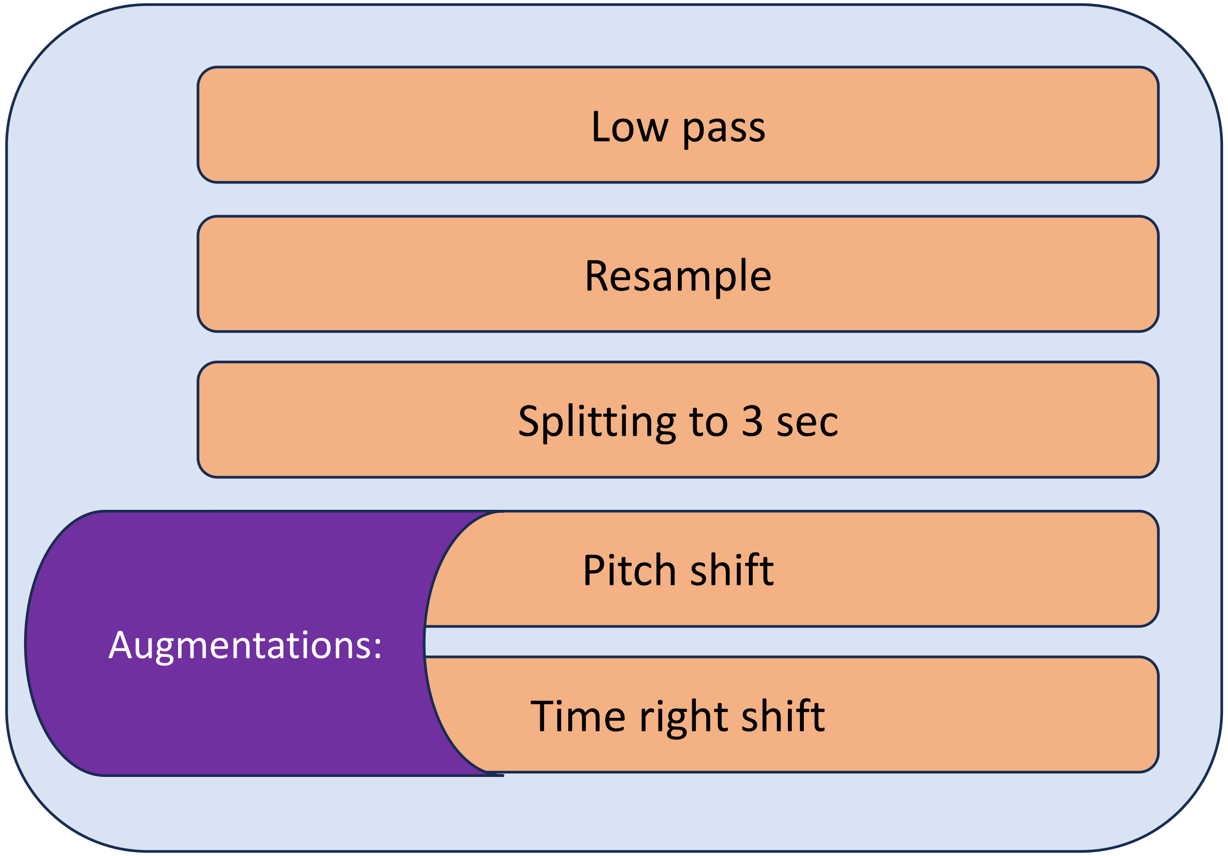
Wavelet vs STFT



Pre-Processing

Dataset A
Size: **124**

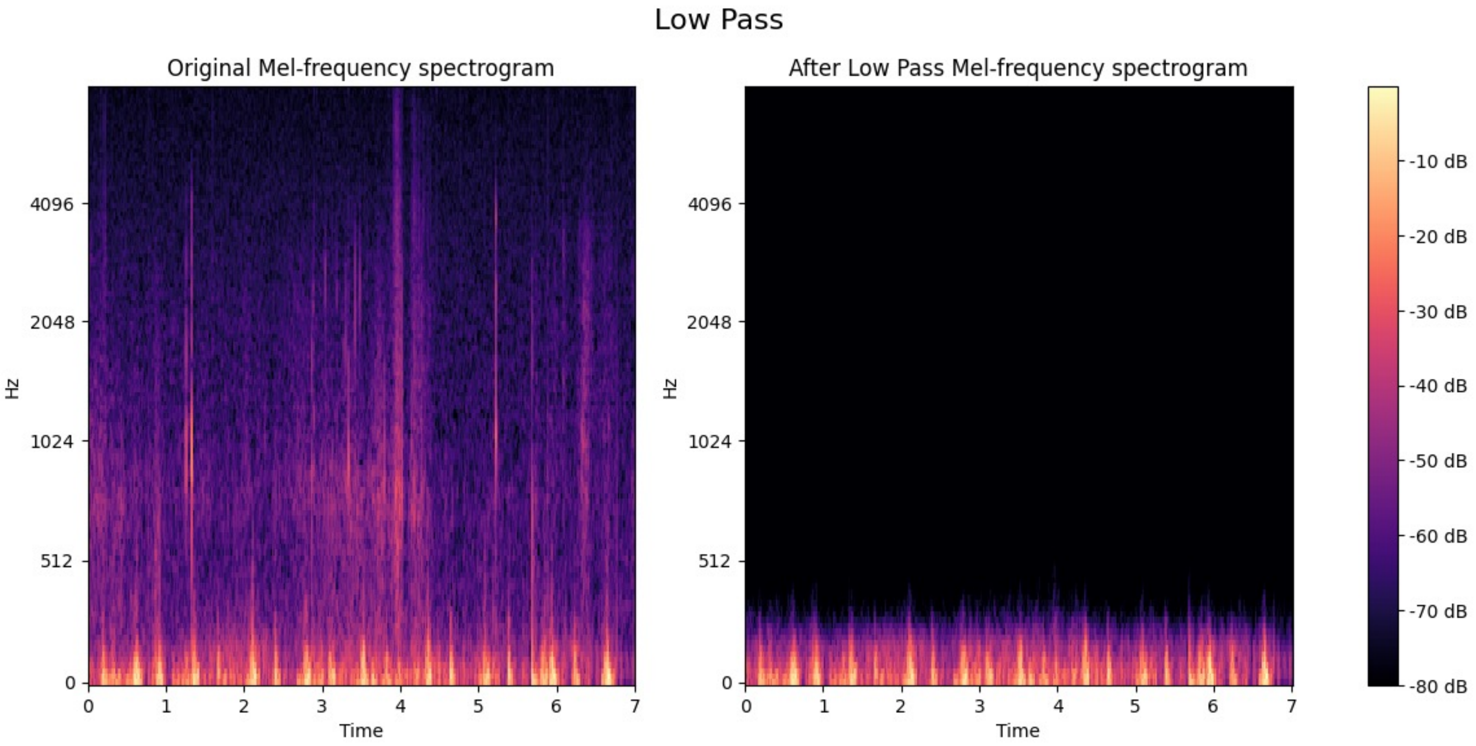
labels	value
normal	31
murmur	34
artifact	40
extrahls	19



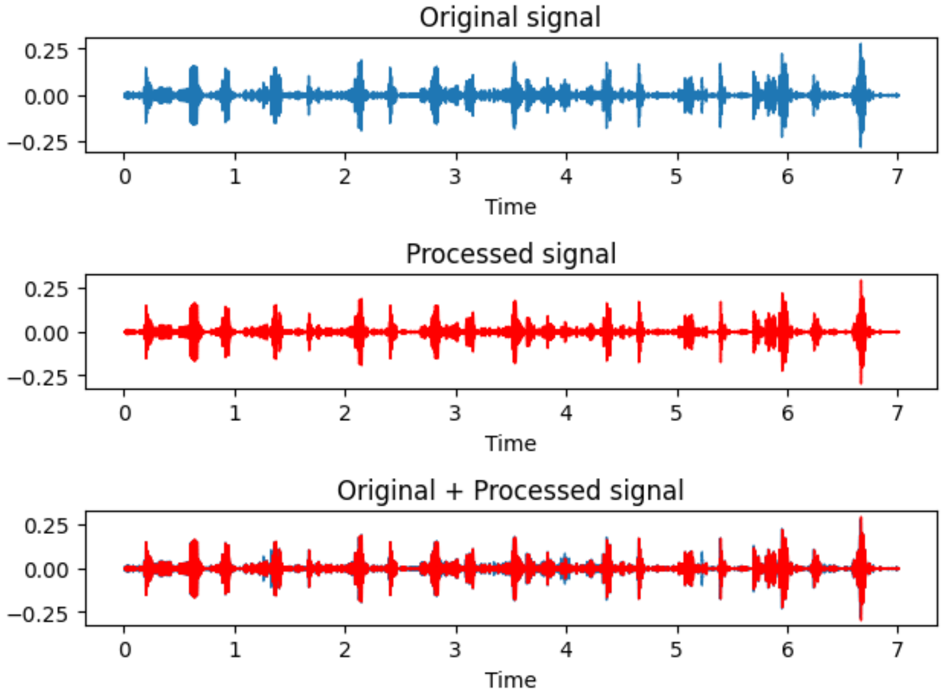
Dataset A
After Pre-processing
Size: **873**

	labels	A	A_pitch_shiftted	A_time_shiftted	sum	%
0	normal	68	68	68	204	0.23
1	murmur	66	66	66	198	0.23
2	artifact	120	120	120	360	0.41
3	extrahls	37	37	37	111	0.13

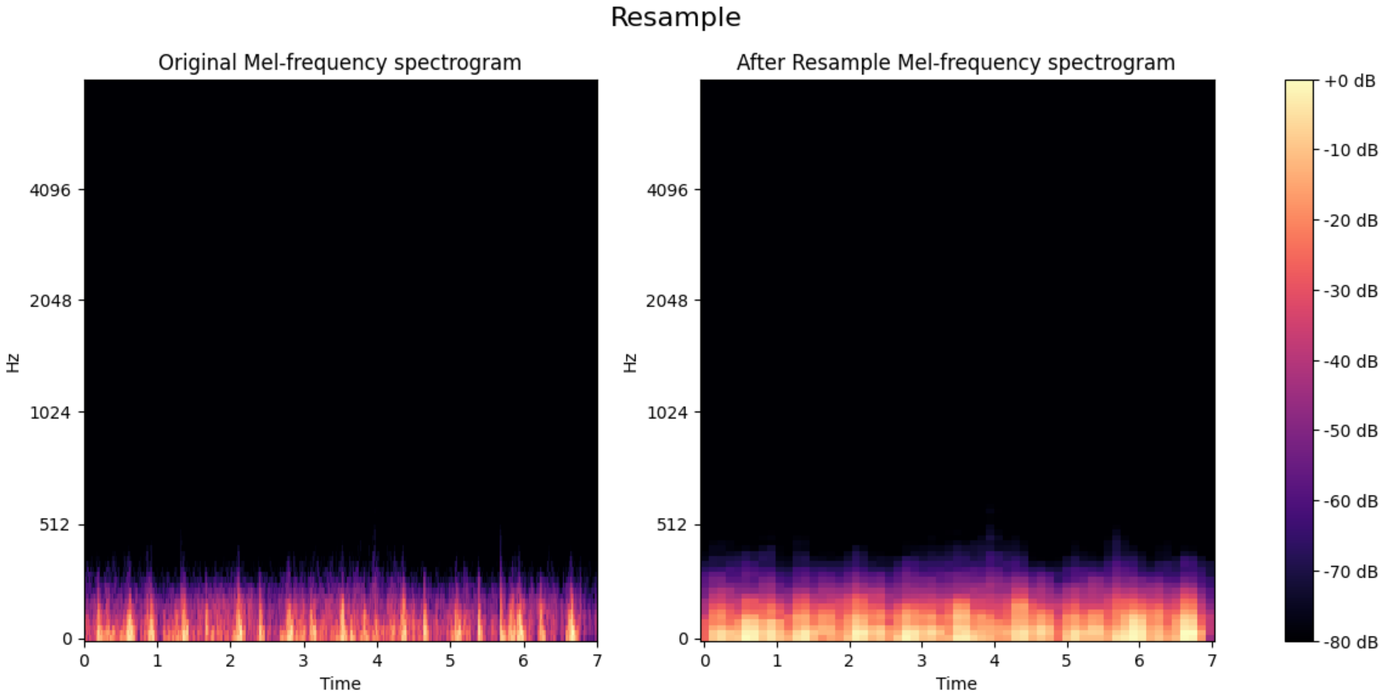
Low pass

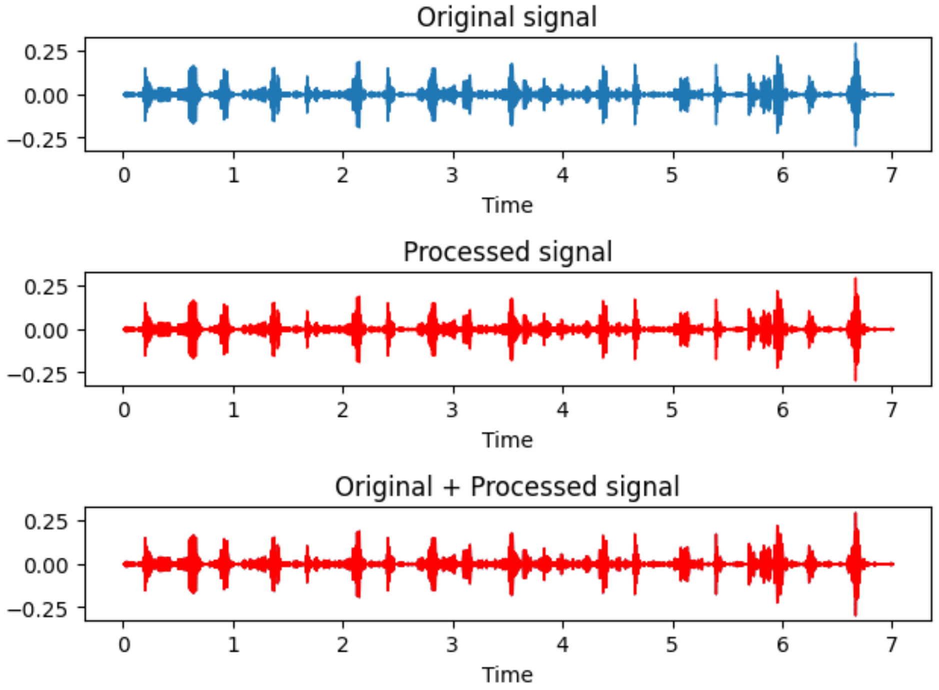


The target frequency is 200 Hz, mainly because it is the frequency of a beating heart



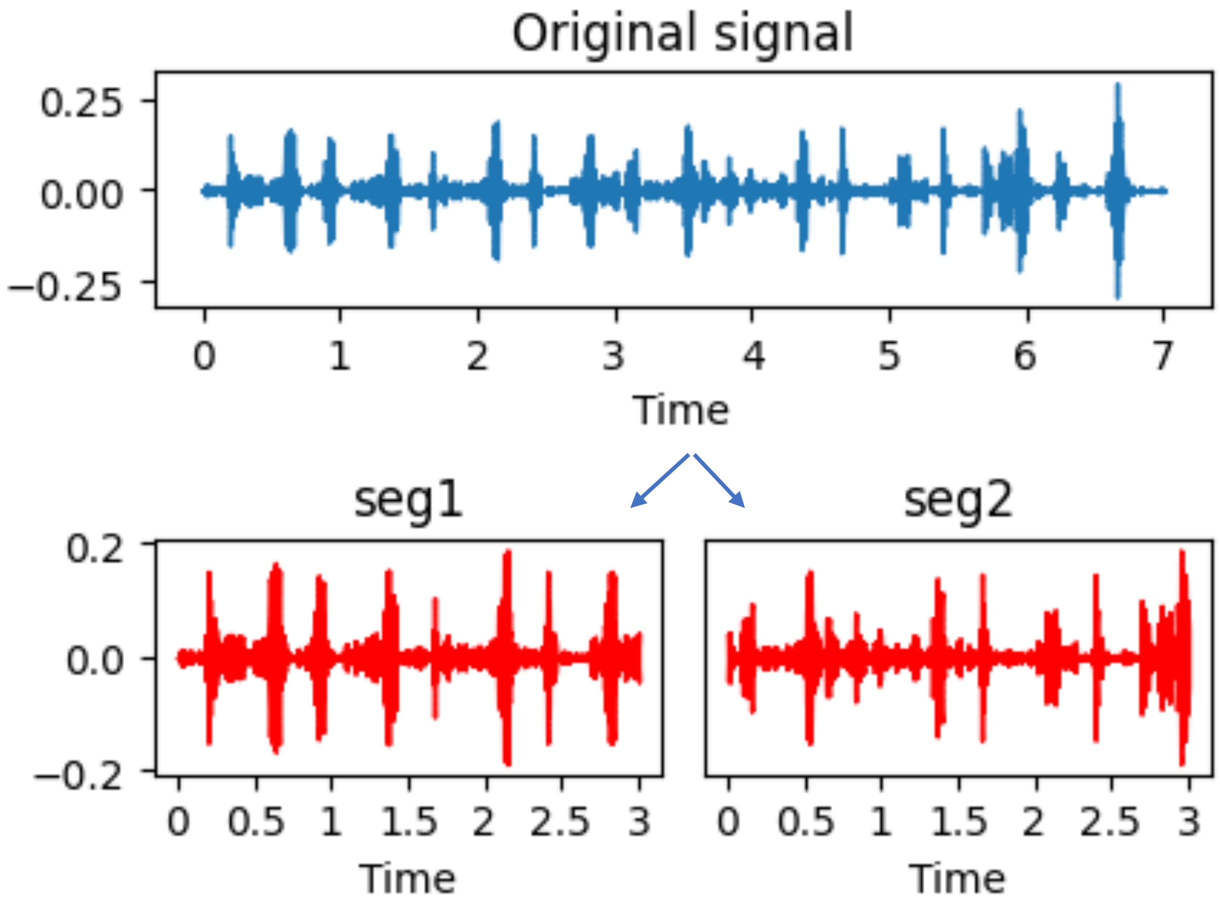
resample





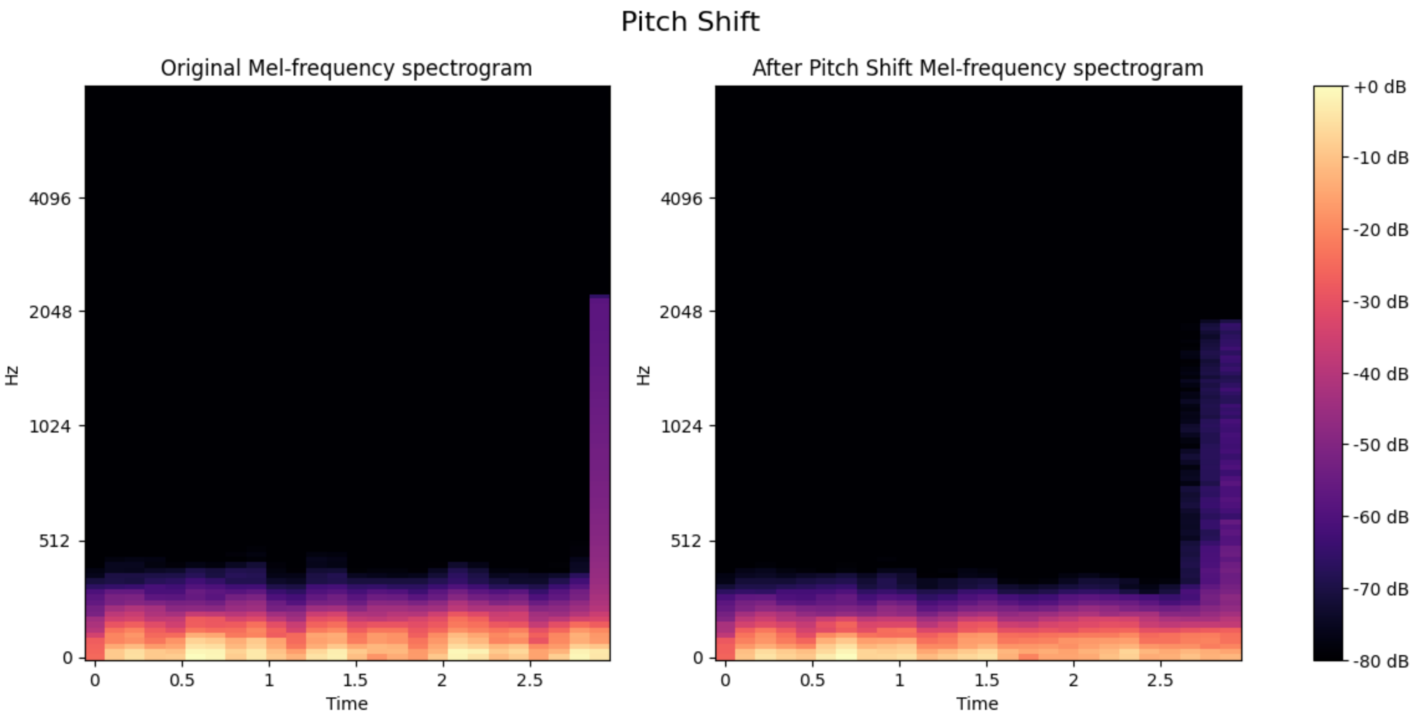
44100 Hz → 4410 Hz

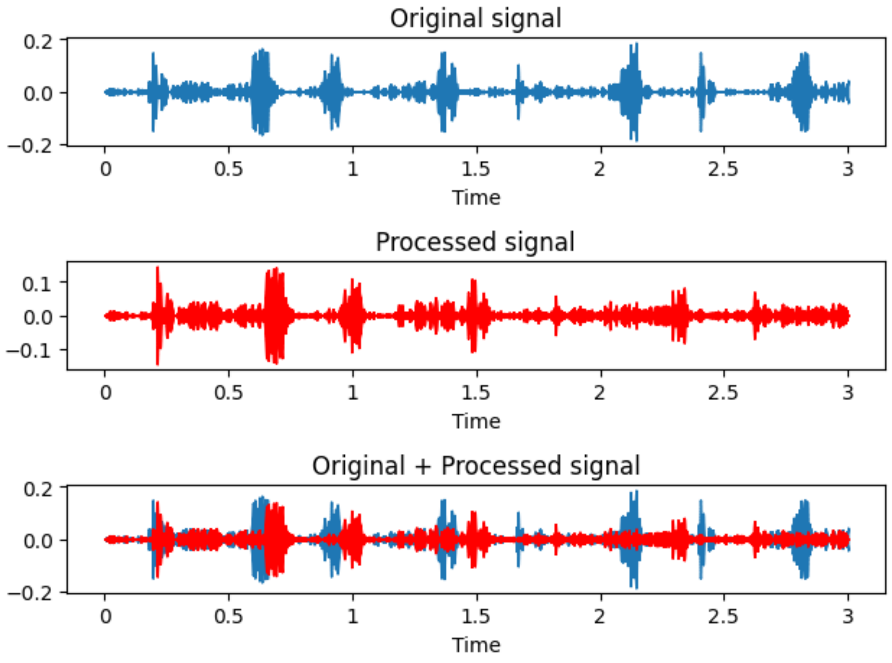
Splitting to 3 sec



	labels	A
0	normal	68
1	murmur	66
2	artifact	120
3	extrahls	37

Augmentations: Pitch shift

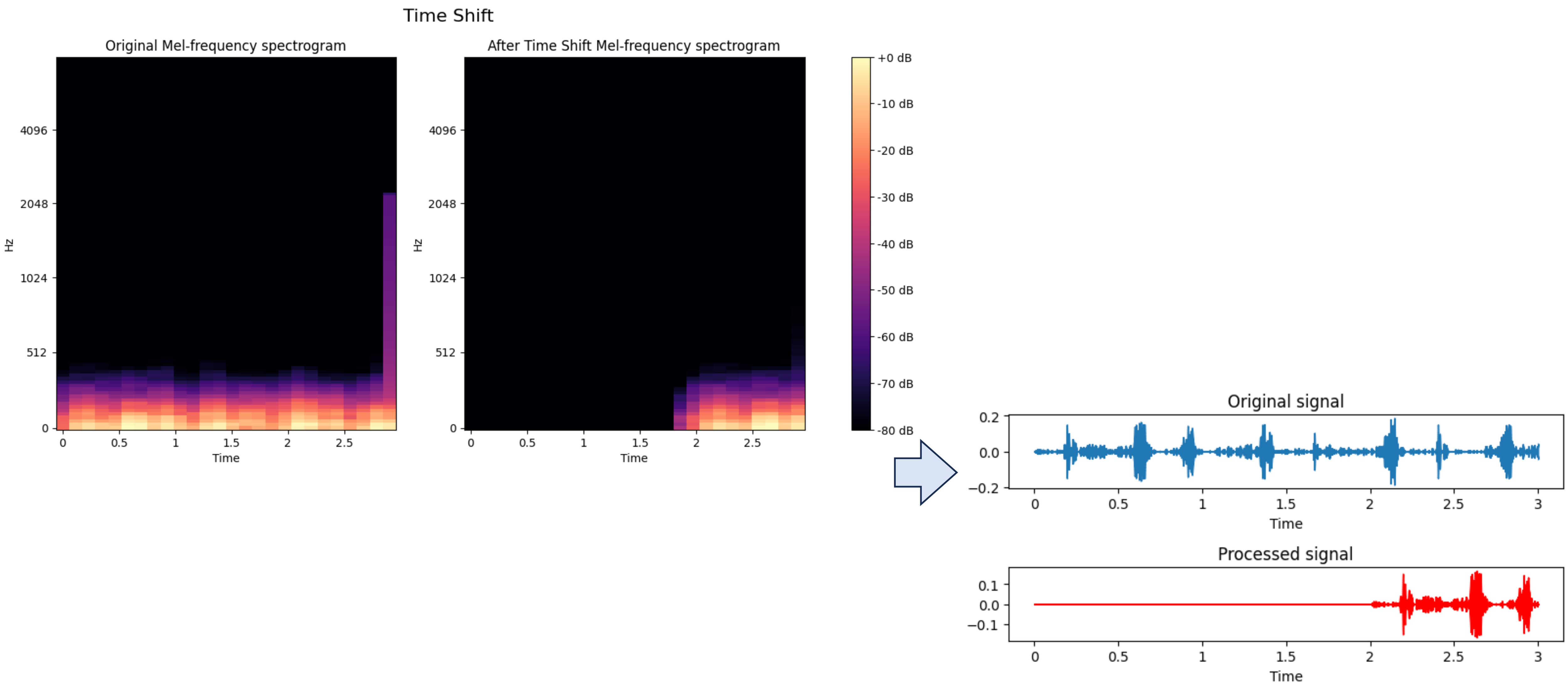




	labels	A	A_pitch_shiftted
0	normal	68	68
1	murmur	66	66
2	artifact	120	120
3	extrahls	37	37

Augmentations:

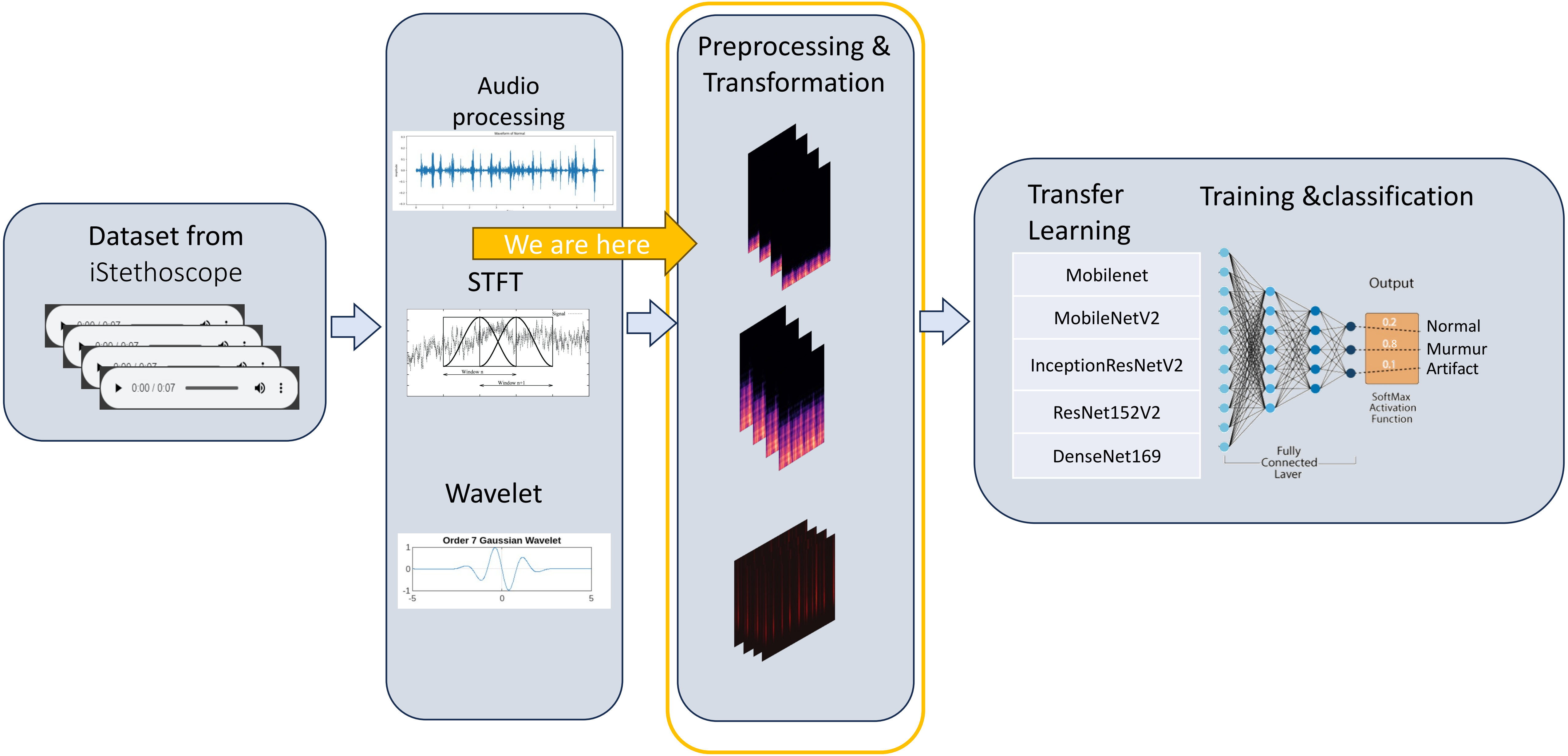
right shift



Summary – Number of samples after augmentations

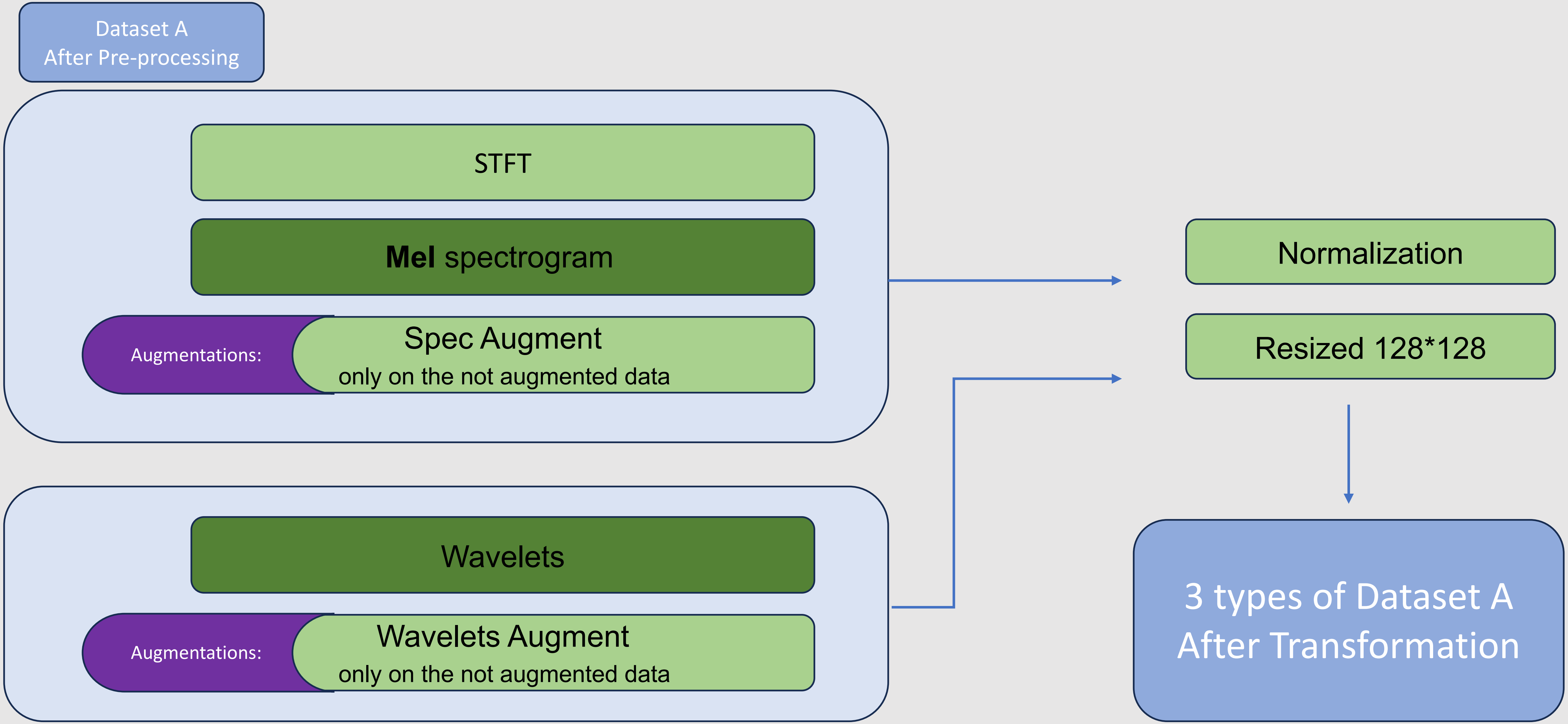
	labels	A	A_pitch_shiftted	A_time_shiftted	sum	%
0	normal	68	68	68	204	0.23
1	murmur	66	66	66	198	0.23
2	artifact	120	120	120	360	0.41
3	extrahls	37	37	37	111	0.13

Our Process:



Dataset from iStethoscope
0:00 / 0:07
Audio processing
STFT
Wavelet
Order 7 Gaussian Wavelet
We are here
Preprocessing & Transformation
Transfer Learning
Training &classification
Mobilenet
MobileNetV2
InceptionResNetV2
ResNet152V2
DenseNet169
Output
0.2
0.8
0.1
Normal
Murmur
Artifact
SoftMax Activation Function
Fully Connected Layer

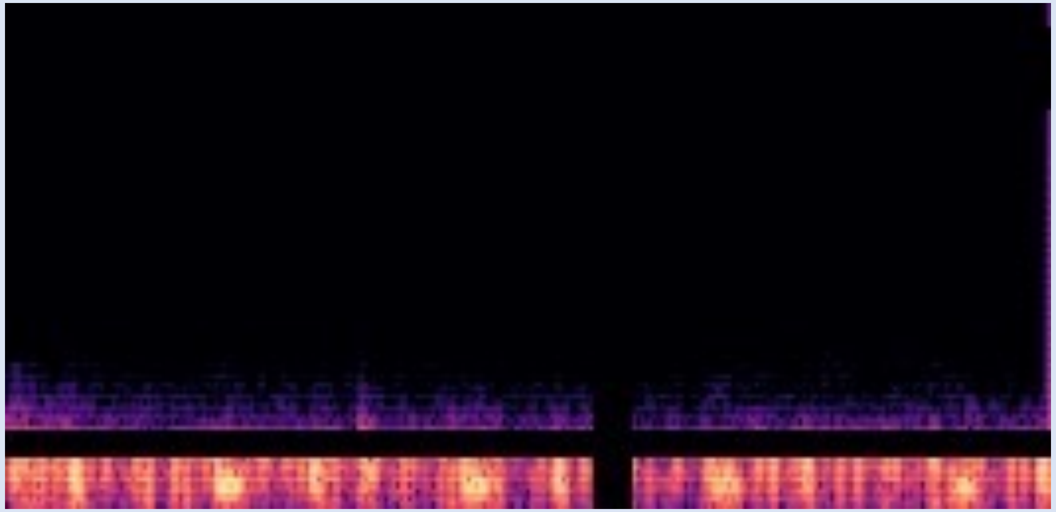
Transformation

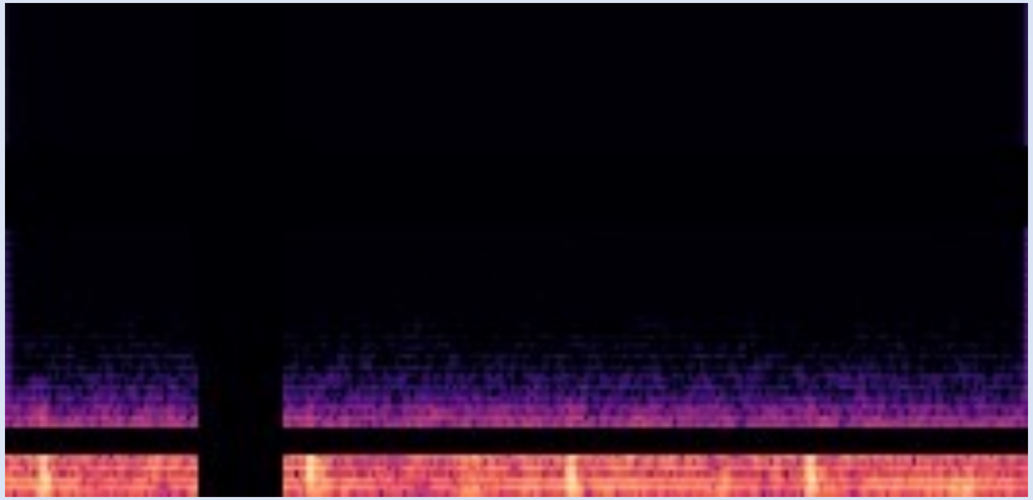


Augmentations:

Spec Augment

only on the not augmented data

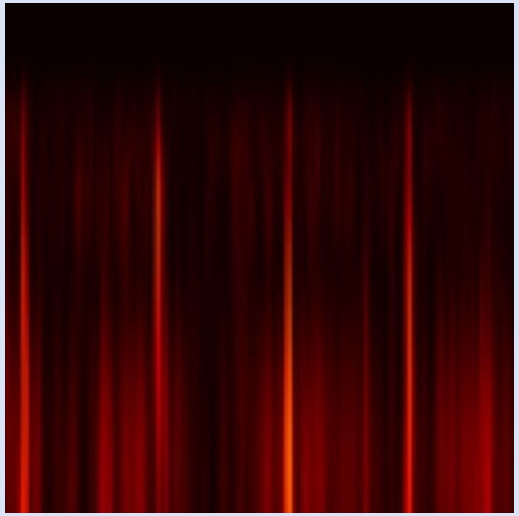




Augmentations:

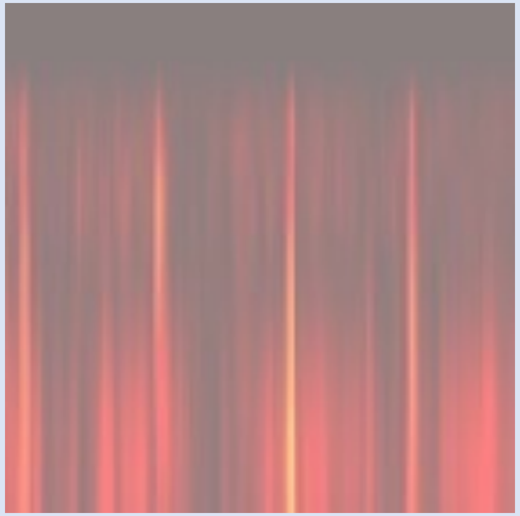
Wavelets Augment

only on the not augmented data



Original

noise



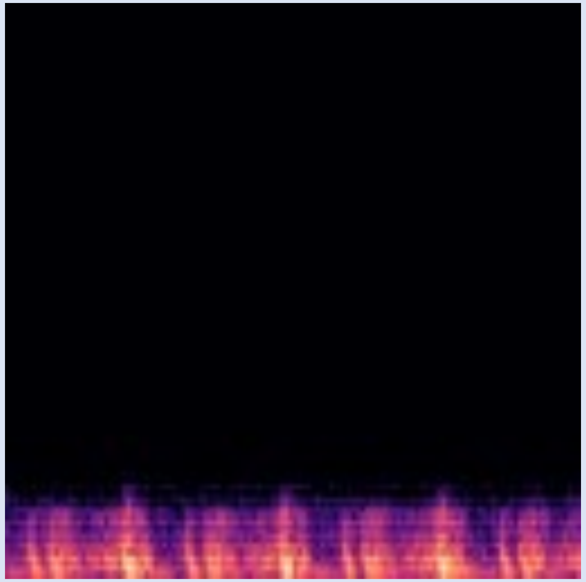
bright

Salt paper

3 types of Dataset A
After Transformation

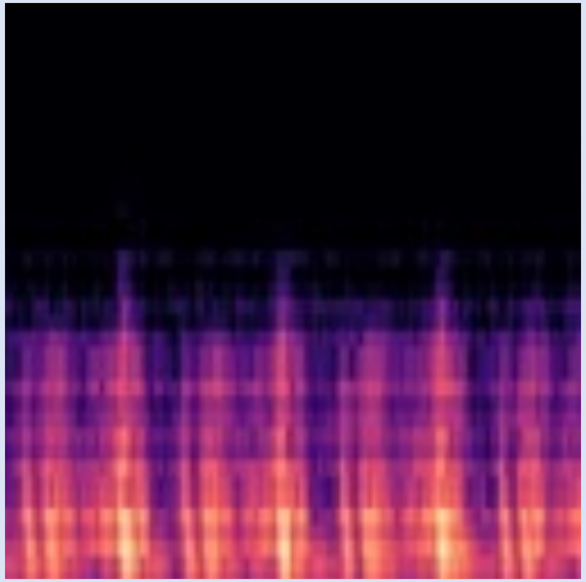
4400 Hz

Original

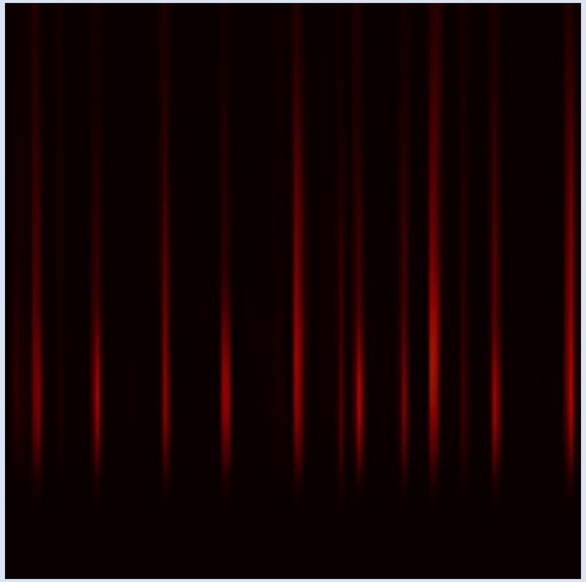


500 Hz

Zoom in



Wavelets



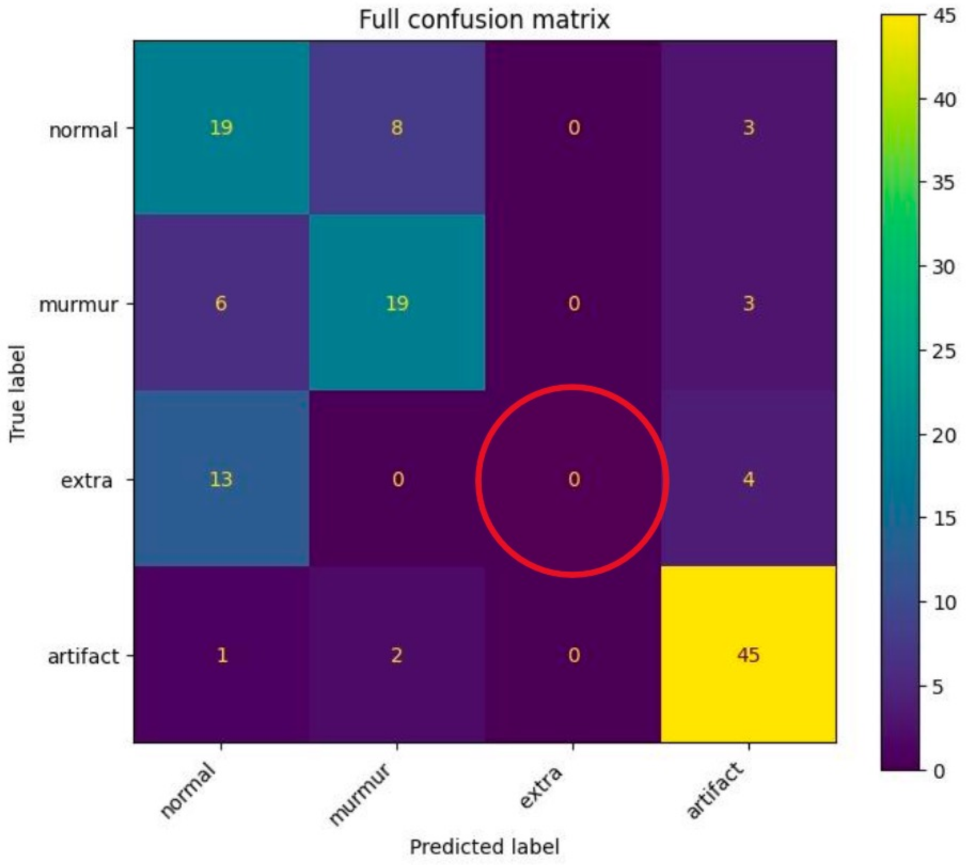
3 **Classes:** Normal, Murmur, Artifact

Why did we drop Extrahls class?

Our model was having difficulty categorizing.

Extra noise and murmurs are common in young children, pregnant women.

The difference was not significant enough for the model to detect.
Without additional data we decided to remove it.



Training & evaluation:

3 types of Dataset A
After Transformation

Split: Training 80% /validation 20%

Transfer Learning

Mobilenet
MobileNetV2
InceptionResNetV2
ResNet152V2
DenseNet169

Why?
- Pre trained
- Familiarity with those kinds of images
- Training rum time

Evaluate Training

Improvement-Weights

Model Selection

The Transfer Model:

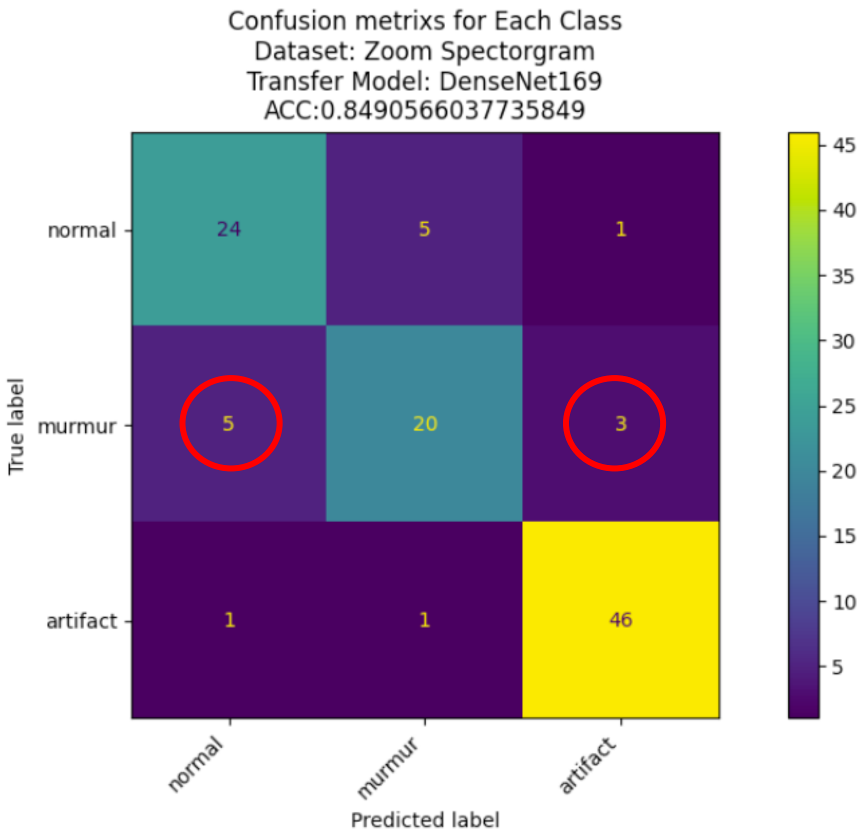
```
model_transfer = Sequential()
transfer.trainable = False
model_transfer.add(transfer)
model_transfer.add(Flatten())
model_transfer.add(BatchNormalization())
model_transfer.add(Dense(64, kernel_regularizer=l2(0.01)))
model_transfer.add(Dropout(0.3))
model_transfer.add(Dense(num_of_class, activation='softmax'))
lr = 0.001  # Learning rate
optimizer = Adam(learning_rate=lr)
model_transfer.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
```

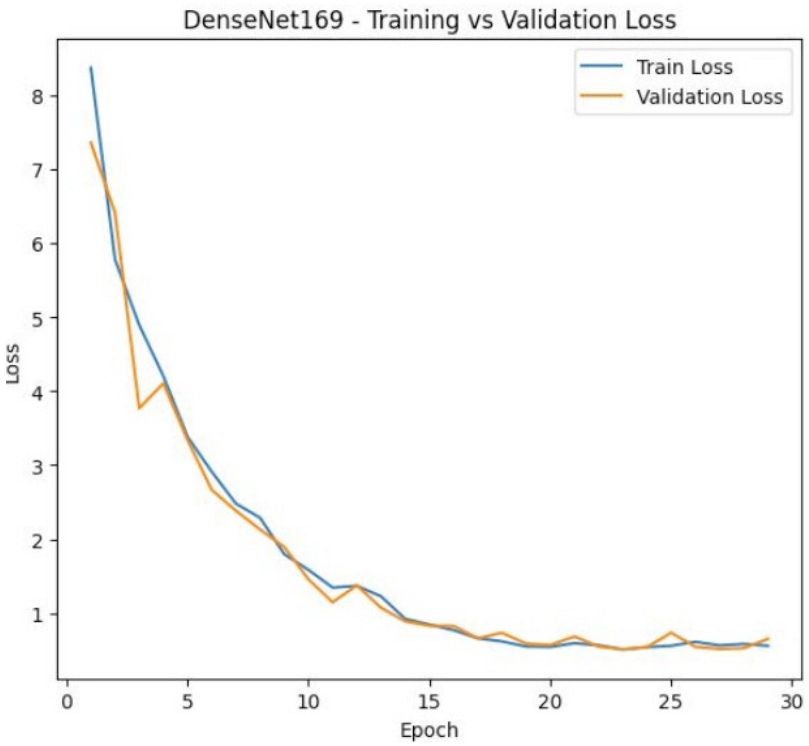
We implemented image generation augmentations, early stopping and saved the weights of the models.

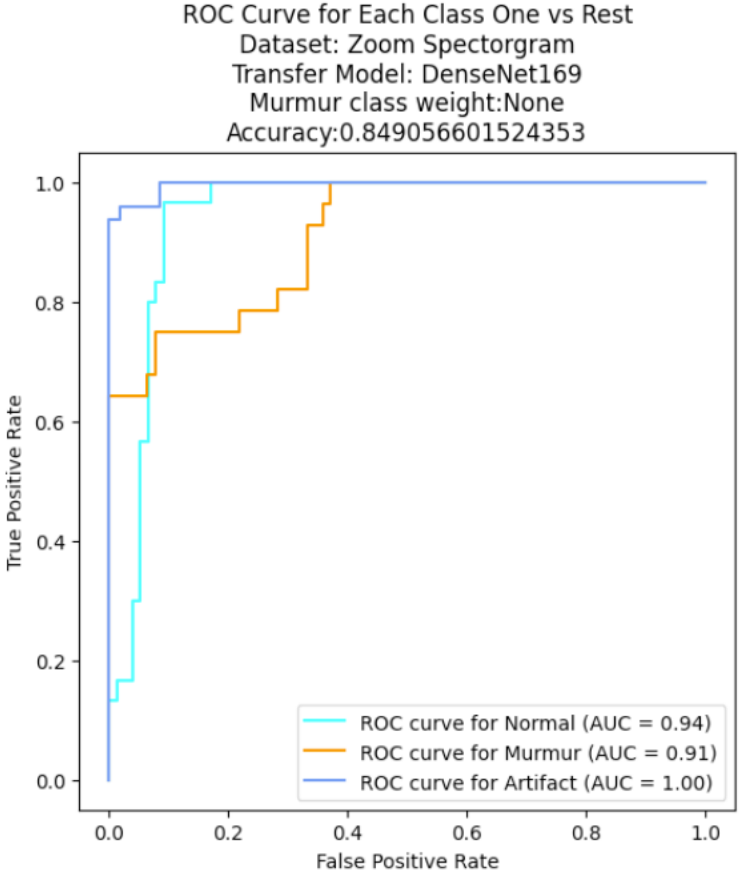
Performance – No class weights best result

df	name	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss	Best Epoch	False negative Murmur
Original Spectorgram	Mobilenet	0.845	0.673	0.857	0.857	0.821	0.797	27	8
	MobileNetV2	0.624	7.709	0.727	0.727	0.745	0.880	1	8
	InceptionResNetV2	0.545	6.601	0.675	0.675	0.745	0.716	1	10
	ResNet152V2	0.572	3.184	0.558	0.558	0.774	0.671	1	9
	DenseNet169	0.641	8.891	0.883	0.883	0.830	0.615	1	10
Zoom Spectorgram	Mobilenet	0.879	0.608	0.844	0.844	0.802	0.813	27	9
	MobileNetV2	0.868	0.651	0.831	0.831	0.811	0.852	26	9
	InceptionResNetV2	0.660	0.880	0.662	0.662	0.717	0.791	29	10
	ResNet152V2	0.646	2.891	0.675	0.675	0.708	0.944	1	10
	DenseNet169	0.868	0.559	0.779	0.779	0.849	0.596	29	8
Wavelet	Mobilenet	0.791	0.704	0.857	0.857	0.792	0.684	18	10
	MobileNetV2	0.817	0.663	0.935	0.935	0.755	0.807	26	7
	InceptionResNetV2	0.545	8.603	0.649	0.649	0.660	0.838	1	7
	ResNet152V2	0.656	2.889	0.857	0.857	0.764	0.916	1	8
	DenseNet169	0.811	0.632	0.935	0.935	0.755	0.752	25	8

Performance – No class weights best result







Why is our focus on murmur false negatives ?

- In medical data analysis, we want to be more accurate in our predictions, so the quantity of false negatives is critical; we don't want to miss diagnose potential patients.
- On the other hand, we don't want to cause medical anxiety to our patients by performing unnecessary testing and stress the medical system.

We decided to iterate the data sets that performed effectively and use the models that performed well for the next performance

Thus, we selected to compare:

Zoom and original spectrogram

MobileNet and DenseNet169

Main reason – runtime in iterations over different weights

Class weights **zoom** spectrogram

df	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Loss	Best Epoch	Test Accuracy score	Weights	False negative Murmur
DenseNet169	0.838	0.512	0.857	0.857	0.594	24	0.849	{0: 1, 1: 0.5, 2: 1}	8
	0.846	0.661	0.818	0.818	0.672	15	0.802	{0: 1, 1: 1, 2: 1}	9
	0.837	0.745	0.792	0.792	0.731	15	0.802	{0: 1, 1: 1.25, 2: 1}	8
	0.812	0.872	0.831	0.831	0.824	10	0.764	{0: 1, 1: 1.5, 2: 1}	7
	0.826	0.873	0.831	0.831	0.778	19	0.783	{0: 1, 1: 1.75, 2: 1}	7
	0.850	0.767	0.883	0.883	0.761	16	0.792	{0: 1, 1: 2, 2: 1}	8
	0.855	0.756	0.883	0.883	0.715	13	0.811	{0: 1, 1: 2.5, 2: 1}	7
	0.836	1.042	0.844	0.844	0.767	8	0.774	{0: 1, 1: 3, 2: 1}	6
	0.823	1.185	0.805	0.805	0.903	17	0.755	{0: 1, 1: 4, 2: 1}	7
	0.849	0.859	0.818	0.818	0.789	12	0.792	{0: 1, 1: 5, 2: 1}	6

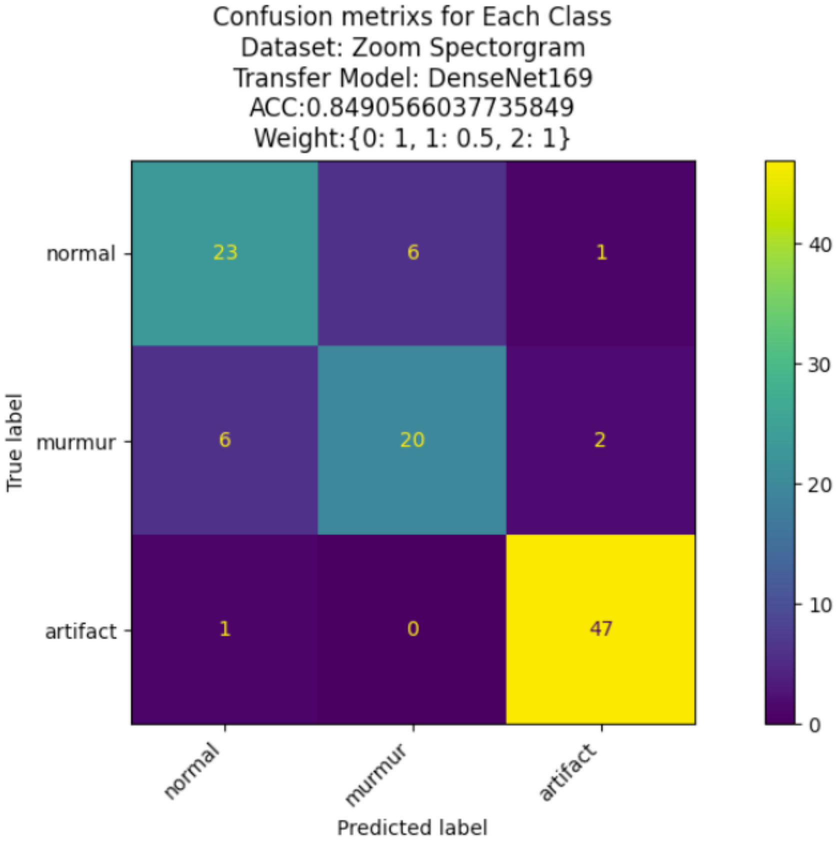
df	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Loss	Best Epoch	Test Accuracy score	Weights	False negative Murmur
MobileNet	0.845	0.579	0.818	0.818	0.662	28	0.840	{0: 1, 1: 0.5, 2: 1}	8
	0.880	0.653	0.844	0.844	0.726	14	0.821	{0: 1, 1: 1, 2: 1}	8
	0.877	0.693	0.831	0.831	0.808	9	0.764	{0: 1, 1: 1.25, 2: 1}	10
	0.871	0.811	0.805	0.805	0.844	17	0.811	{0: 1, 1: 1.5, 2: 1}	8
	0.849	0.956	0.870	0.870	0.826	17	0.811	{0: 1, 1: 1.75, 2: 1}	10
	0.873	0.903	0.857	0.857	0.976	7	0.764	{0: 1, 1: 2, 2: 1}	9
	0.901	0.723	0.857	0.857	0.802	22	0.811	{0: 1, 1: 2.5, 2: 1}	9
	0.879	0.868	0.805	0.805	0.899	13	0.811	{0: 1, 1: 3, 2: 1}	8
	0.888	0.851	0.883	0.883	0.743	14	0.840	{0: 1, 1: 4, 2: 1}	9
	0.876	1.011	0.857	0.857	1.027	15	0.774	{0: 1, 1: 5, 2: 1}	6

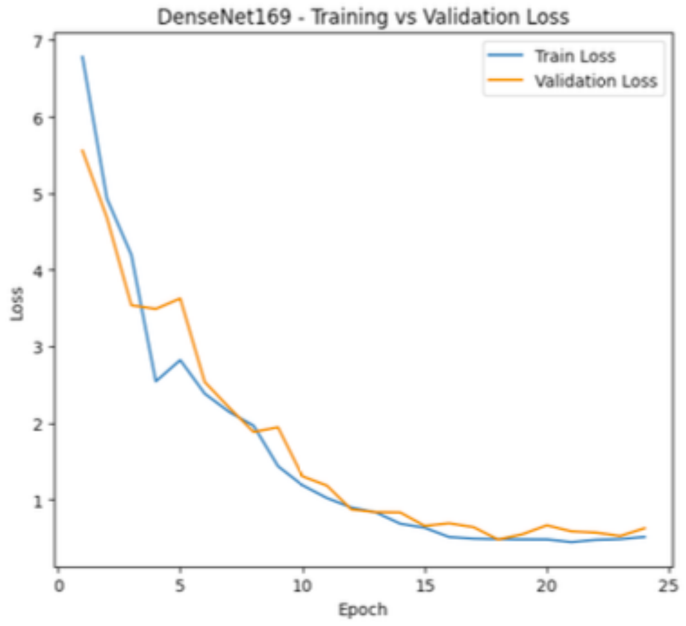
Class weights **original** spectrogram

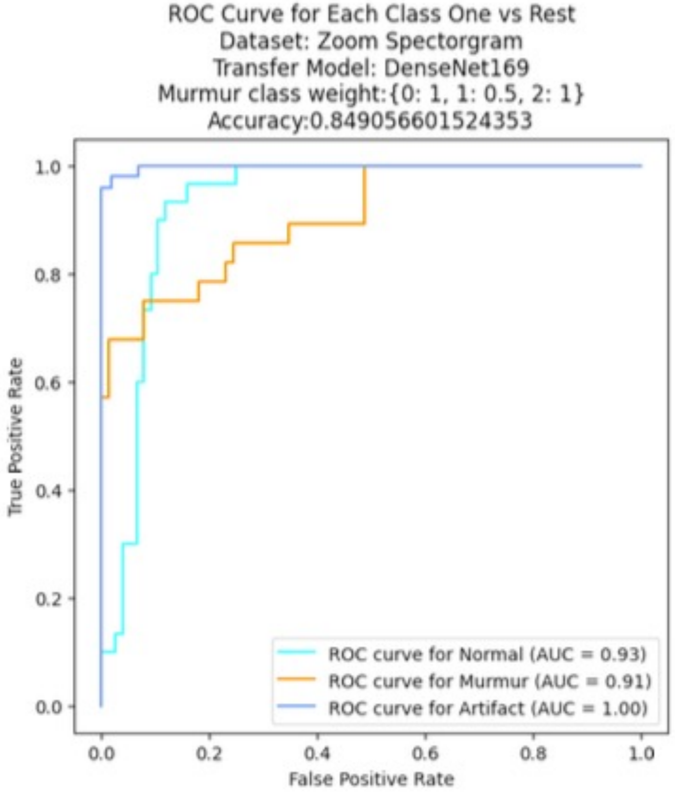
name	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Loss	Best Epoch	Test Accuracy score	Weights	False negative Murmur
DenseNet169	0.626	9.087	0.818	0.818	0.628	1	0.8585	{0: 1, 1: 0.5, 2: 1}	12
	0.774	0.768	0.883	0.883	0.654	11	0.7642	{0: 1, 1: 1, 2: 1}	10
	0.792	0.822	0.870	0.870	0.668	7	0.7642	{0: 1, 1: 1.25, 2: 1}	9
	0.785	0.856	0.857	0.857	0.673	14	0.8396	{0: 1, 1: 1.5, 2: 1}	9
	0.750	0.950	0.896	0.896	0.800	7	0.8113	{0: 1, 1: 1.75, 2: 1}	8
	0.770	1.050	0.883	0.883	0.703	15	0.8302	{0: 1, 1: 2, 2: 1}	10
	0.767	1.108	0.896	0.896	0.806	12	0.7358	{0: 1, 1: 2.5, 2: 1}	7
	0.770	1.156	0.831	0.831	0.765	17	0.7736	{0: 1, 1: 3, 2: 1}	9
	0.770	1.138	0.844	0.844	0.760	13	0.7736	{0: 1, 1: 4, 2: 1}	8
	0.785	1.194	0.857	0.857	0.797	15	0.7736	{0: 1, 1: 5, 2: 1}	6

name	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Loss	Best Epoch	Test Accuracy score	Weights	False negative Murmur
MobileNet	0.637	5.408	0.805	0.805	0.684	1	0.8302	{0: 1, 1: 0.5, 2: 1}	9
	0.806	0.756	0.844	0.844	0.808	17	0.8396	{0: 1, 1: 1, 2: 1}	8
	0.829	0.795	0.883	0.883	0.812	17	0.7642	{0: 1, 1: 1.25, 2: 1}	11
	0.839	0.825	0.844	0.844	0.897	9	0.8113	{0: 1, 1: 1.5, 2: 1}	9
	0.814	0.894	0.883	0.883	0.916	17	0.7453	{0: 1, 1: 1.75, 2: 1}	7
	0.822	0.862	0.831	0.831	0.932	9	0.8113	{0: 1, 1: 2, 2: 1}	10
	0.816	0.960	0.805	0.805	1.000	17	0.7358	{0: 1, 1: 2.5, 2: 1}	4
	0.832	1.021	0.805	0.805	1.103	7	0.7358	{0: 1, 1: 3, 2: 1}	5
	0.832	0.986	0.779	0.779	1.000	21	0.7453	{0: 1, 1: 4, 2: 1}	5
	0.798	1.228	0.831	0.831	1.131	1	0.7170	{0: 1, 1: 5, 2: 1}	5

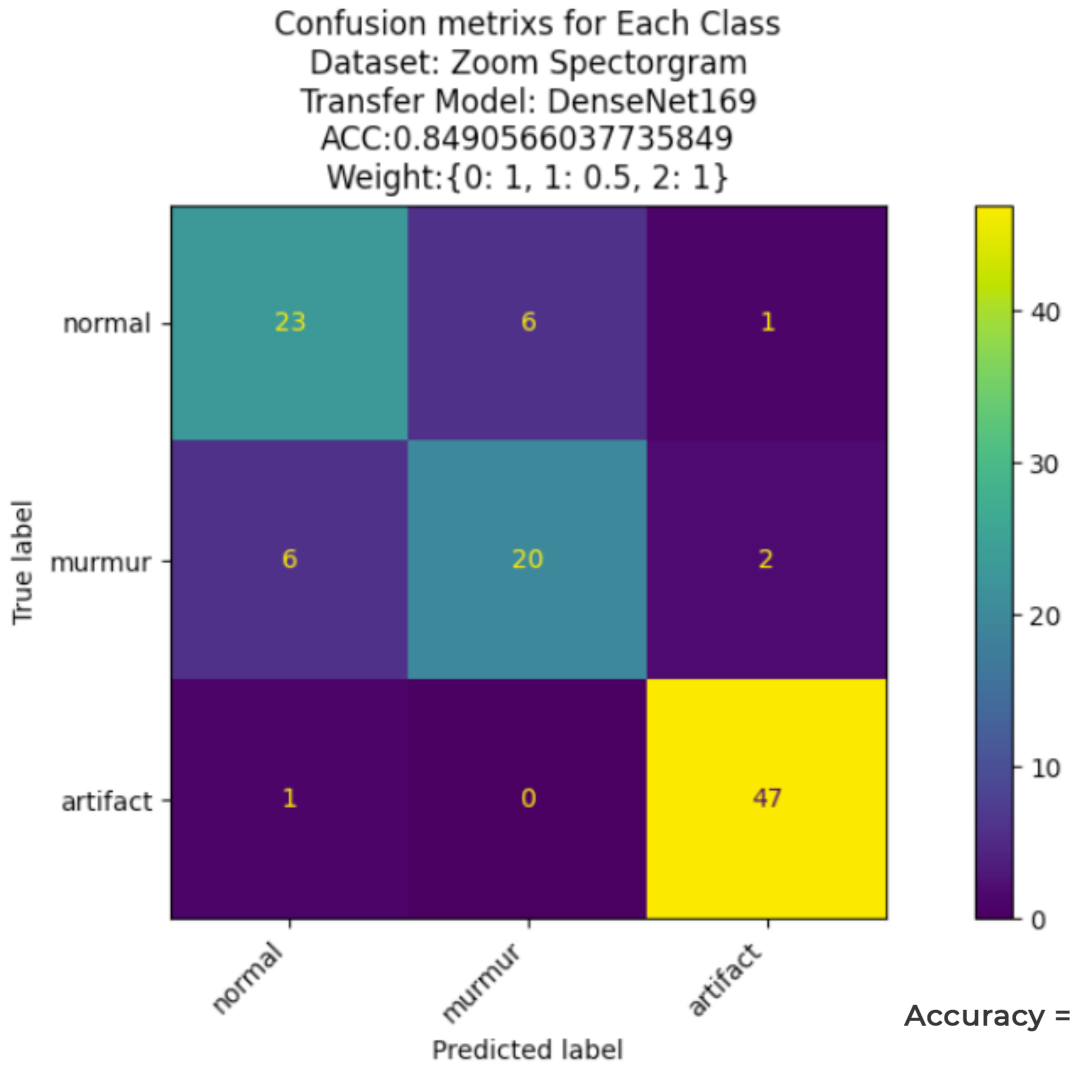
Best Performance With Weights
Zoom and DenseNet169

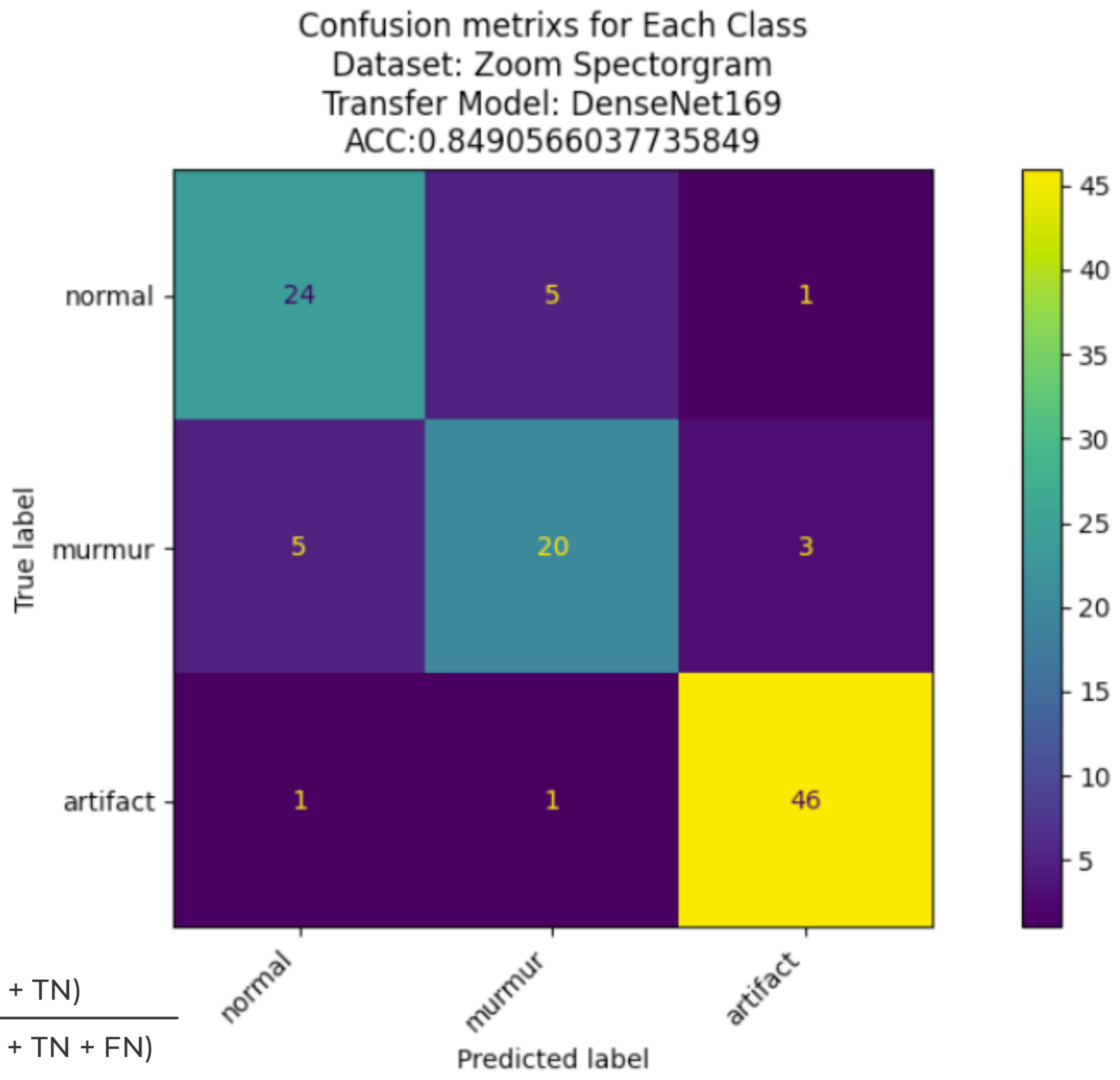






Training Weights VS No Weights

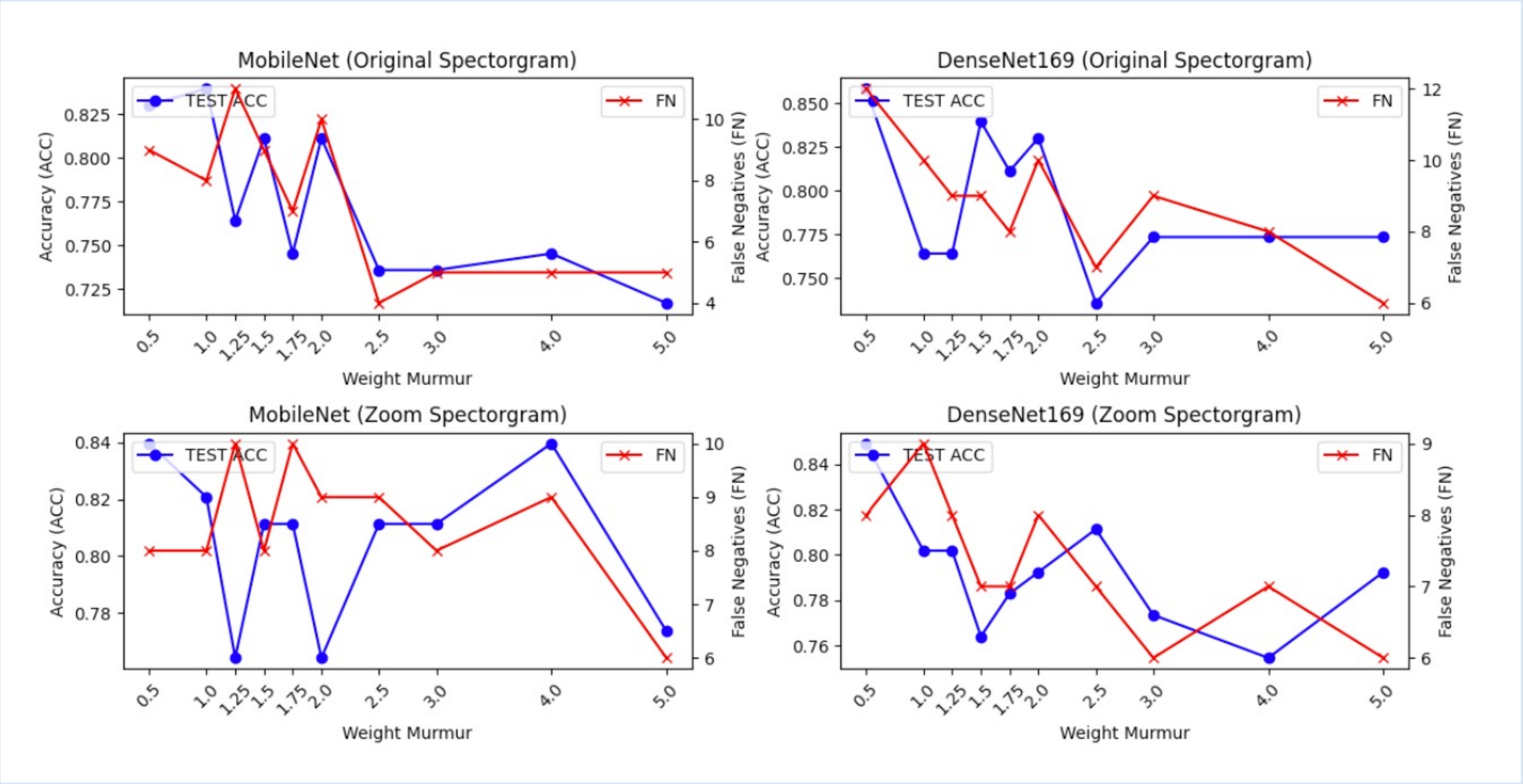




$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

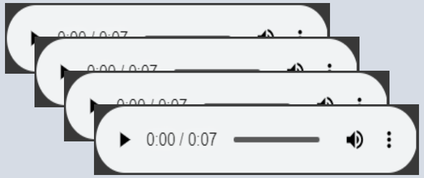
The accuracy is the same because the sum of false positive and false negatives is the same hence the equation of accuracy

Different Training Weights By Accuracy For The Parameters Above

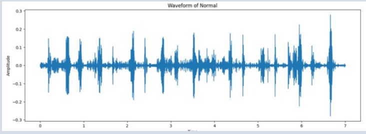


Recap Of Our Process

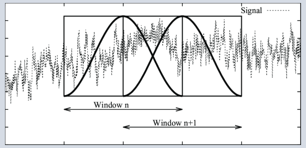
Dataset from iStethoscope



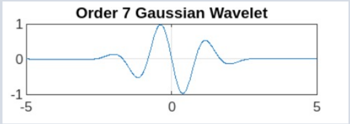
Audio processing



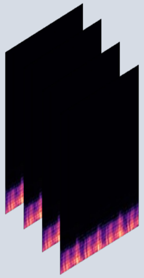
STFT

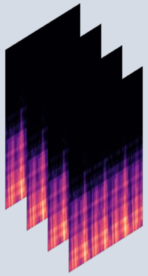


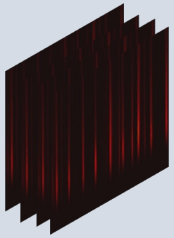
Wavelet



Preprocessing & Transformation



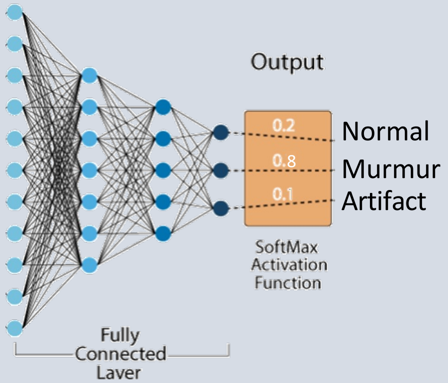




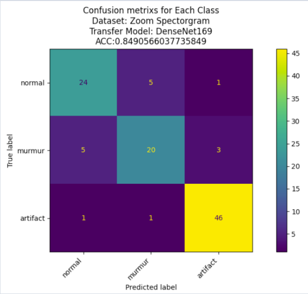
Transfer Learning

- Mobilenet
- MobileNetV2
- InceptionResNetV2
- ResNet152V2
- DenseNet169

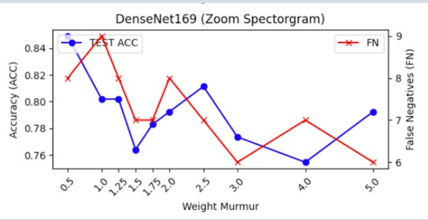
Training &classification



Model evaluation



Weighted classes



Conclusions

- Zoom and DenseNet169 provide the best performance.
- Given the limited data, we did see progress and good results.
- There are no major differences between the original and the zoom.
- Weights didn’t show significant difference.
- We dropped Wavelet and Extrahls class due to low performance and runtime issues.
- Stochasticity – performance could not be duplicated.

Future work and limitation

Limitations

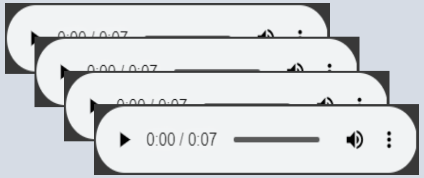
- More detailed data – murmur is too broad.
- Threshold for 3 classes
- Runtime and commutating power.
- Efficient execution.

Future work

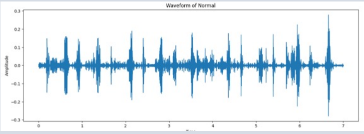
- Model embedded in an app.
- Specify specific heart condition.
- Active learning.

Recap Of Our Process

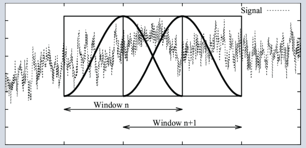
Dataset from iStethoscope



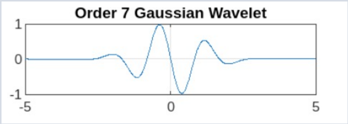
Audio processing



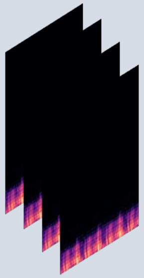
STFT

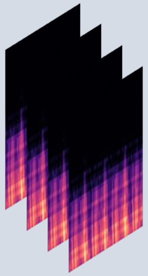


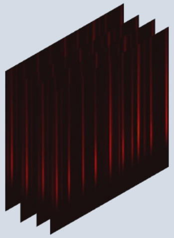
Wavelet



Preprocessing & Transformation



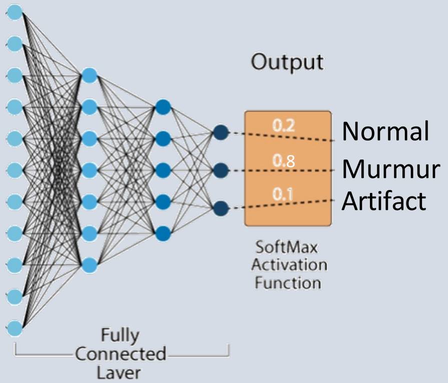




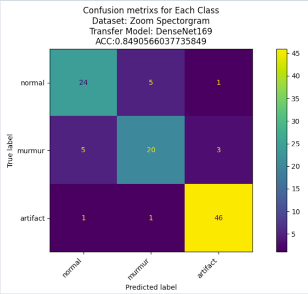
Transfer Learning

Mobilenet
MobileNetV2
InceptionResNetV2
ResNet152V2
DenseNet169

Training &classification



Model evaluation



Weighted classes