Multi-class Classification of Bird Species Using Birdsongs

SC1015 Mini Project | FCMA Team 3

Introduction - Data Set

Data Set: British Birdsong Dataset (Kaggle)
Data Base: Xeno-Canto
Author: Rachael Tatman

^“xc”[0-9]+$.flac

- 264 samples
- 88 unique species
- 3 samples per species

metadata.xls

- Label video ID with species
- Contributor of Audio
- Country of Origin

Icons created by Pixel perfect, from flaticon.com

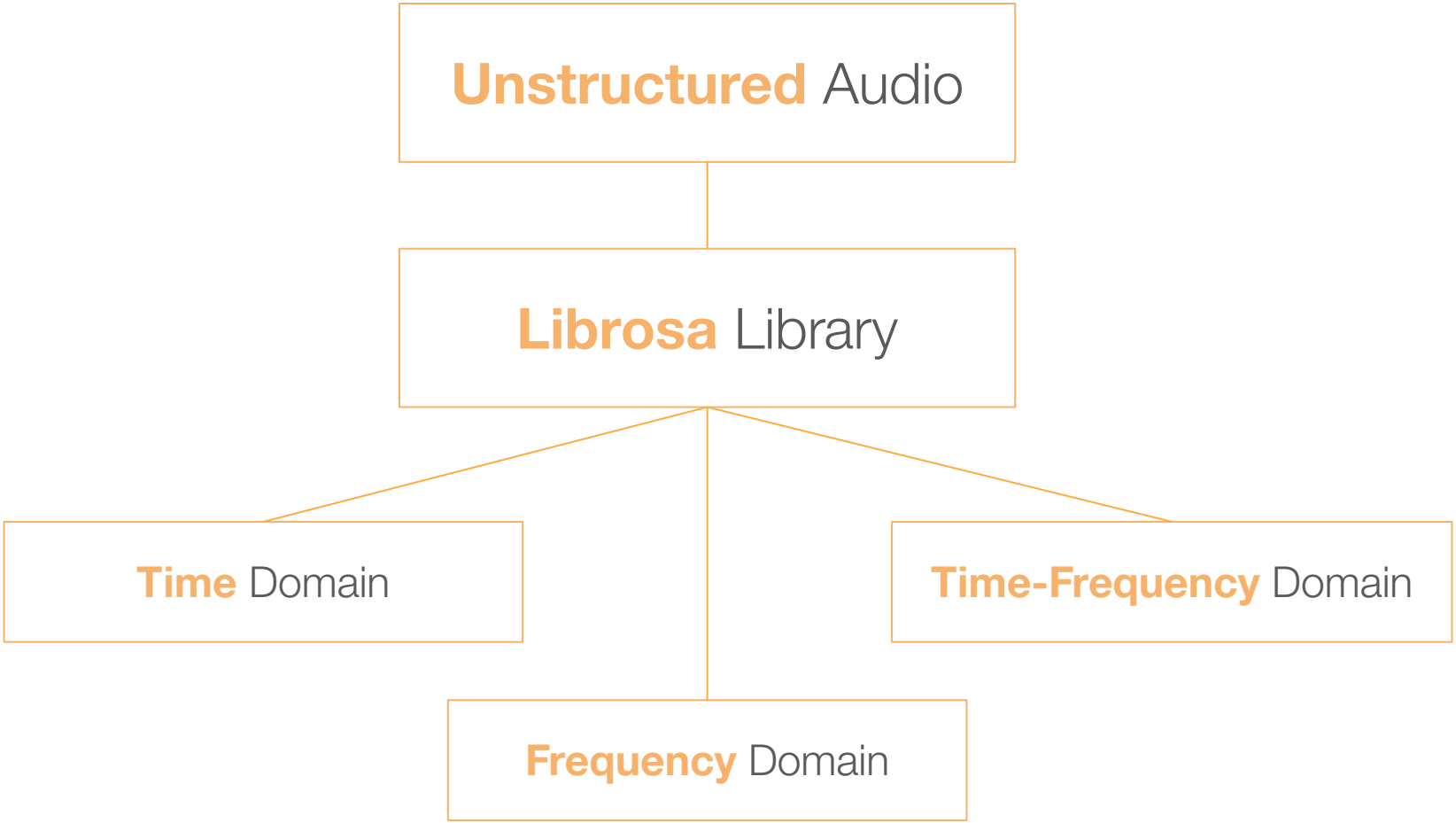
To **identify** bird species based on their **birdsong**

Birdsong: birdsong as a unique identity of bird species

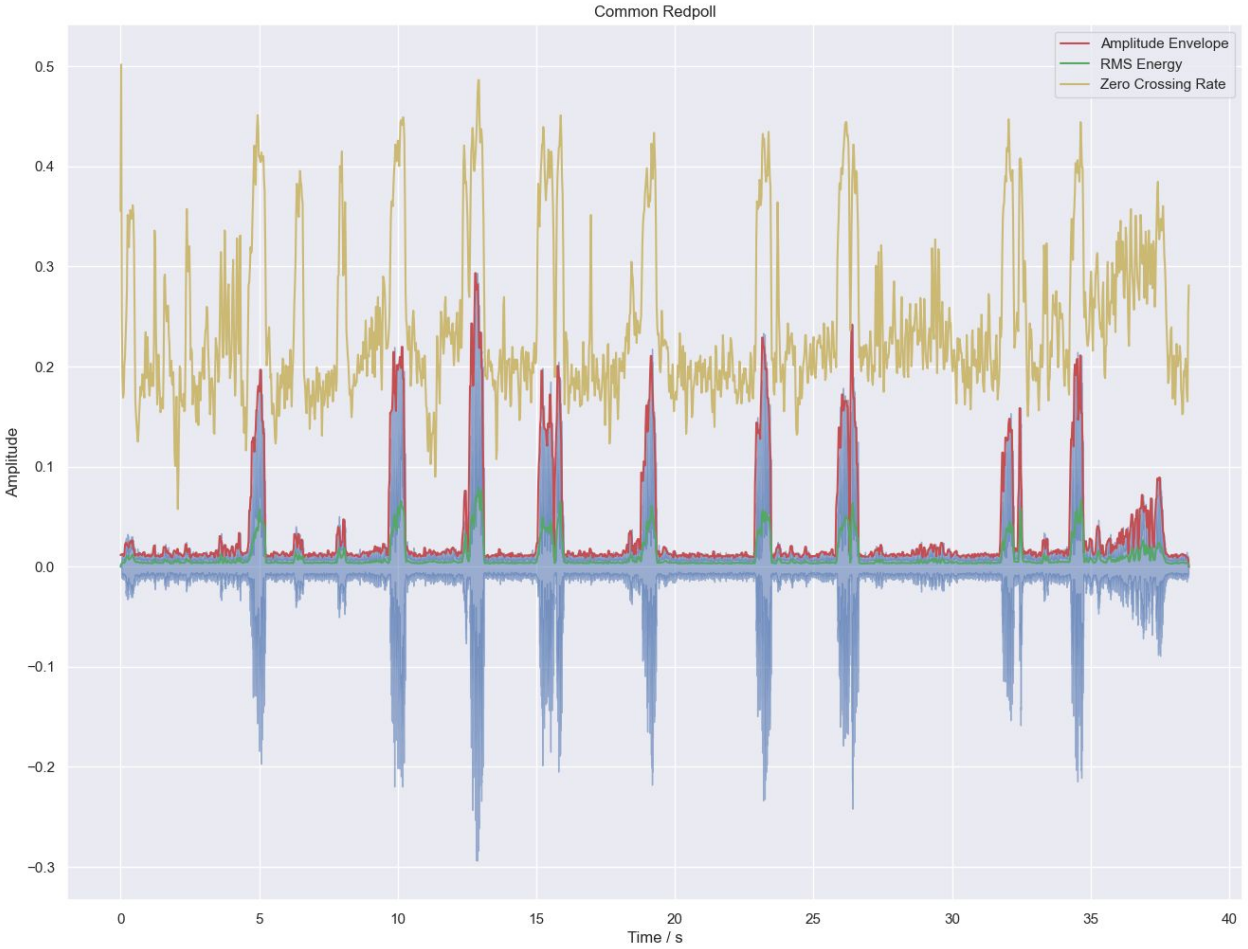
Conservation: identify endangered species

Appreciation: knowing the presence of the species

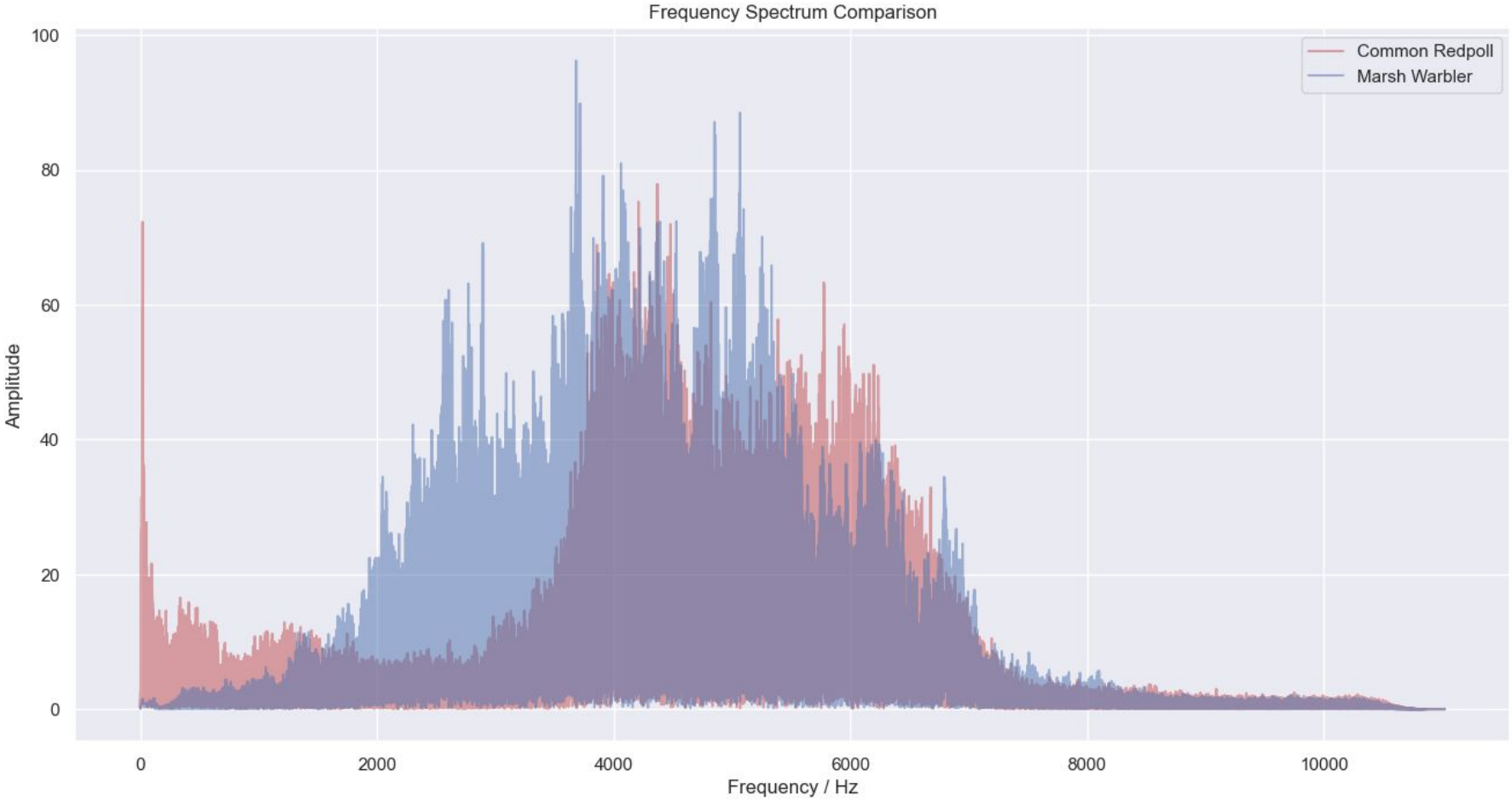
Exploratory Data Analysis


Unstructured Audio
Librosa Library
Time Domain
Time-Frequency Domain
Frequency Domain

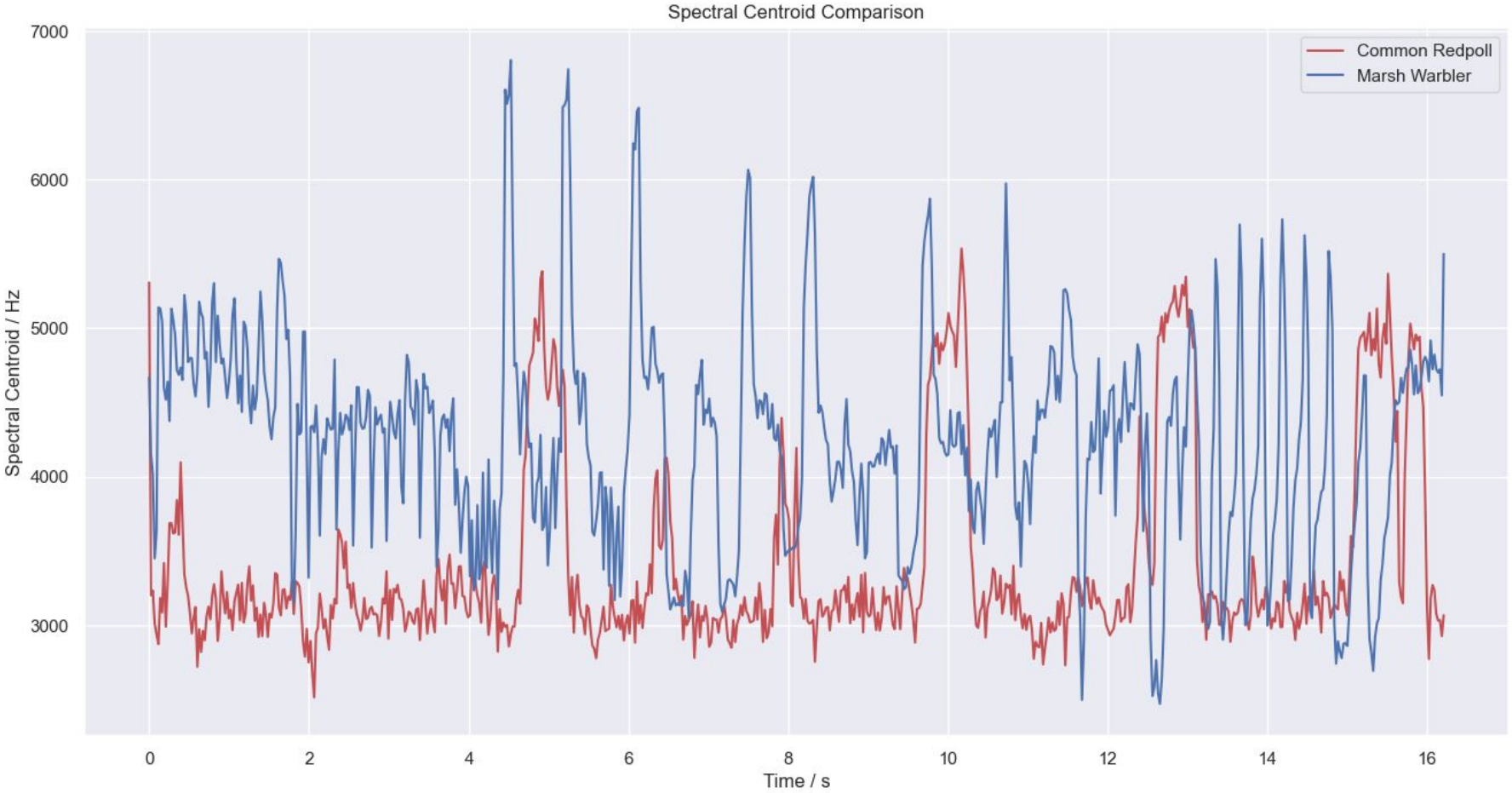
Exploratory Data Analysis - Time Domain



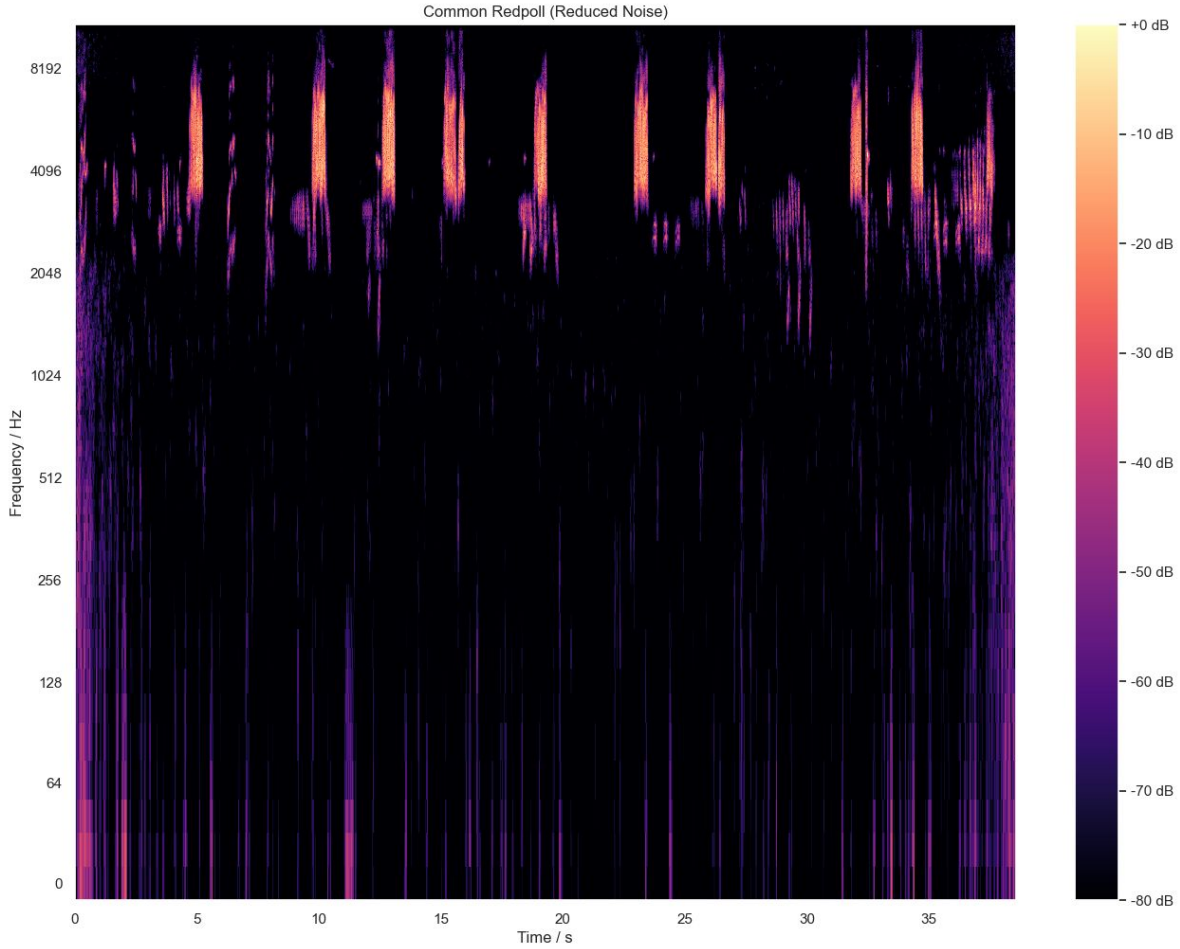
Exploratory Data Analysis - Frequency Domain



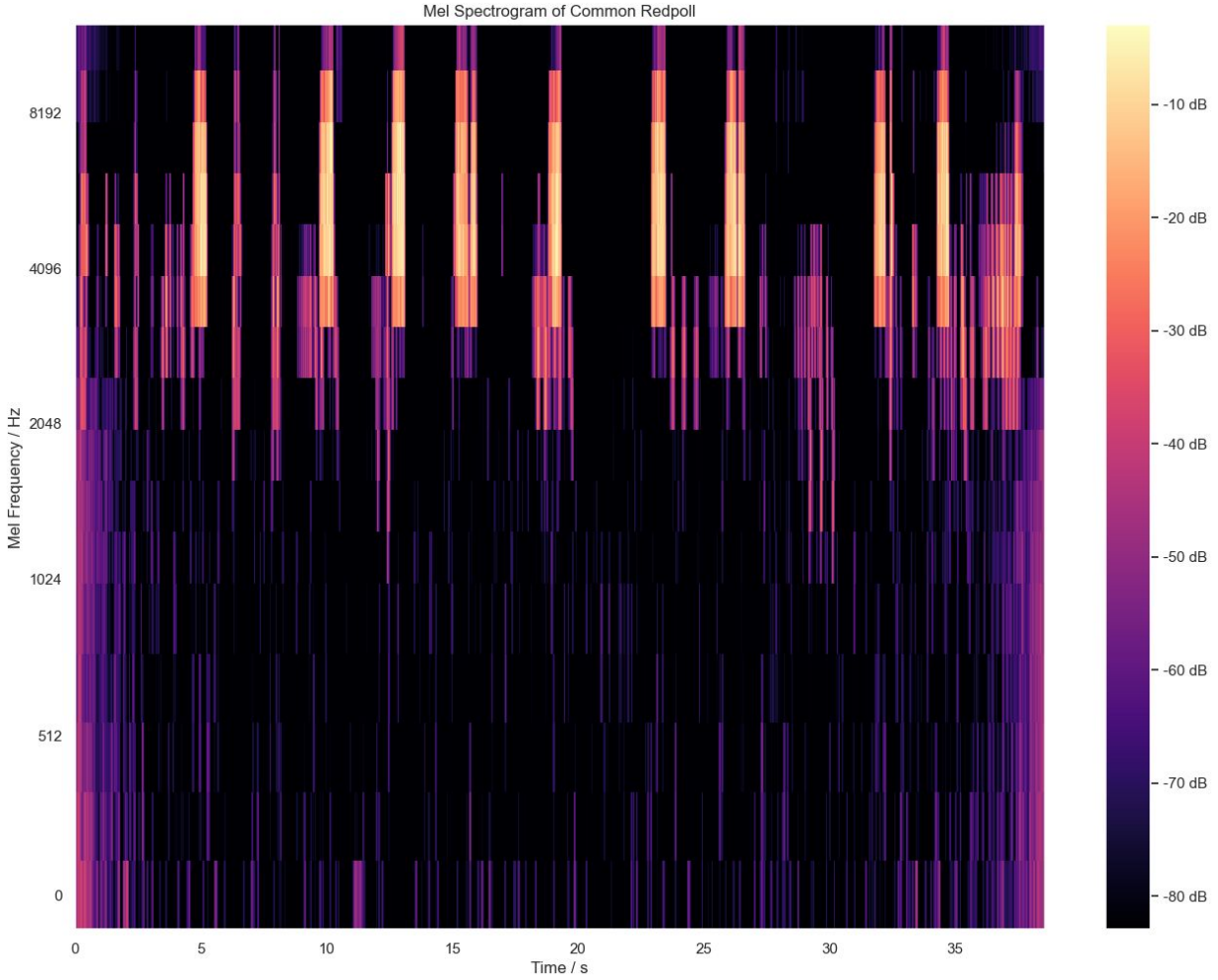
Exploratory Data Analysis - Frequency Domain



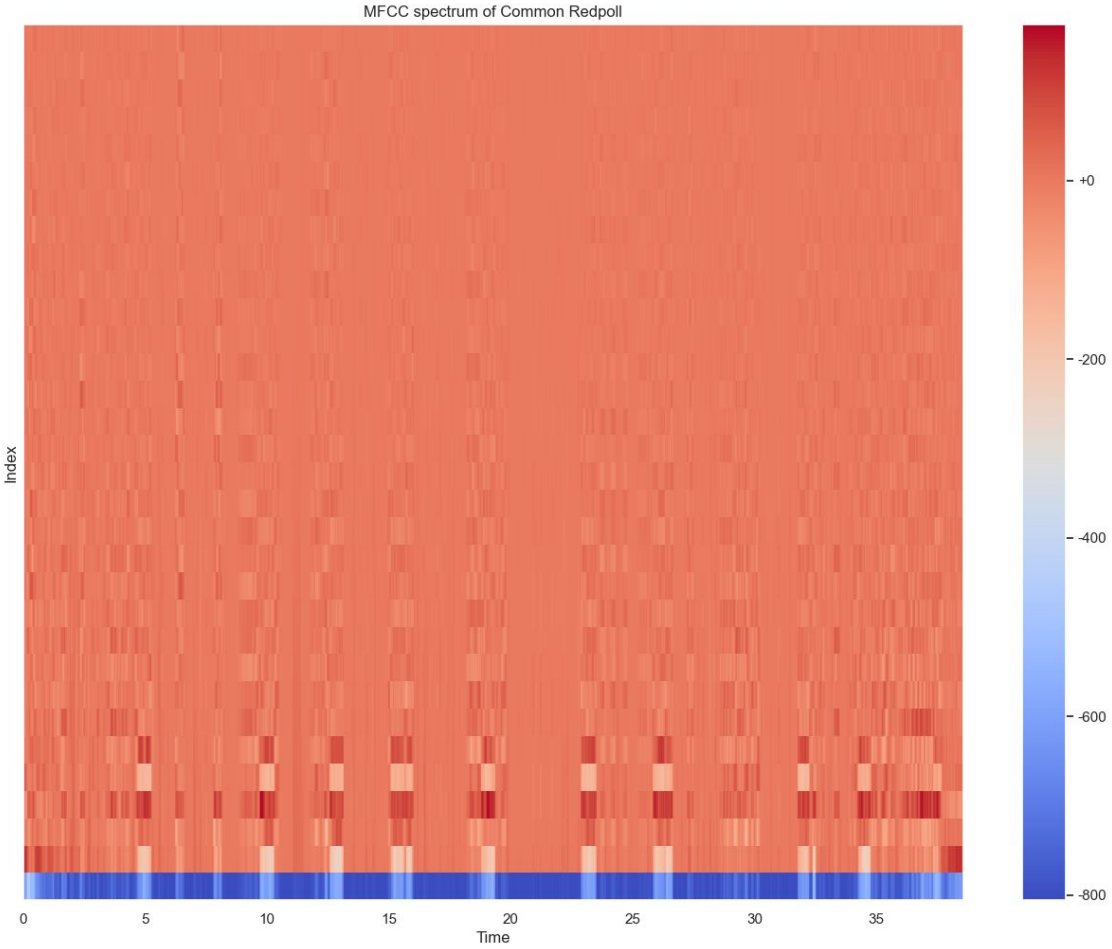
Exploratory Data Analysis - Time-Frequency Domain



Exploratory Data Analysis - Time-Frequency Domain

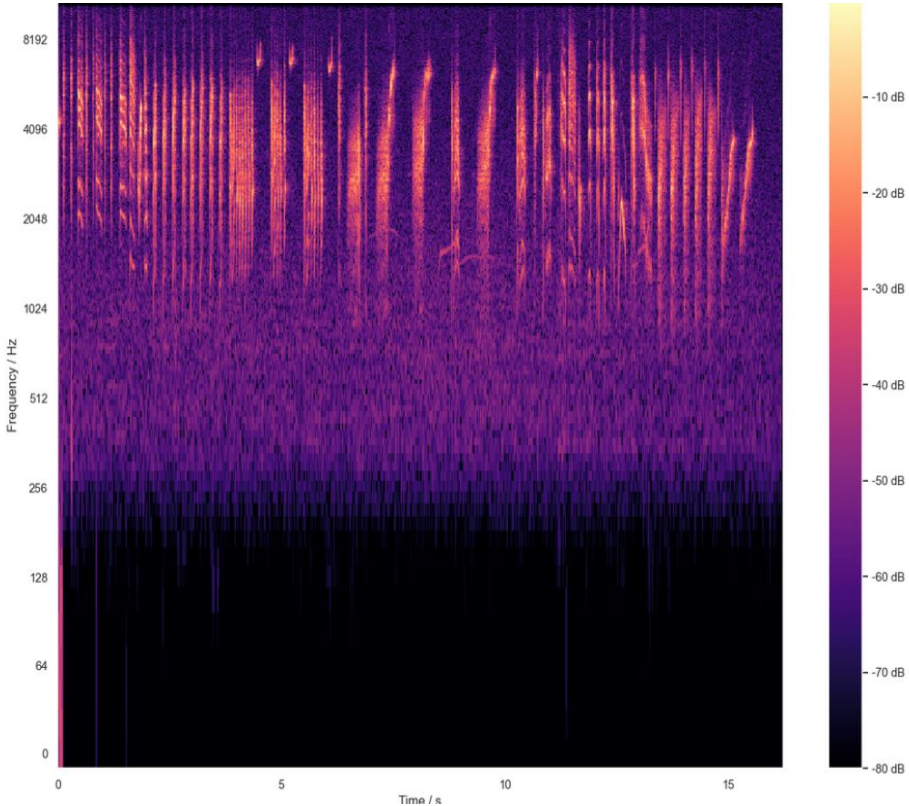


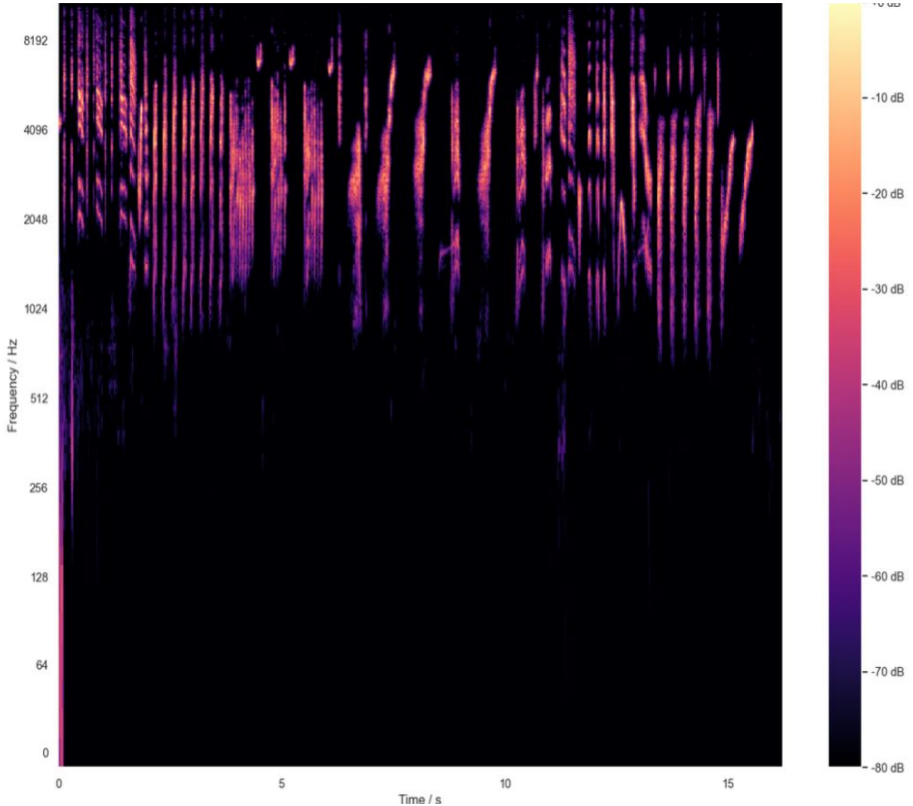
Exploratory Data Analysis - Time-Frequency Domain



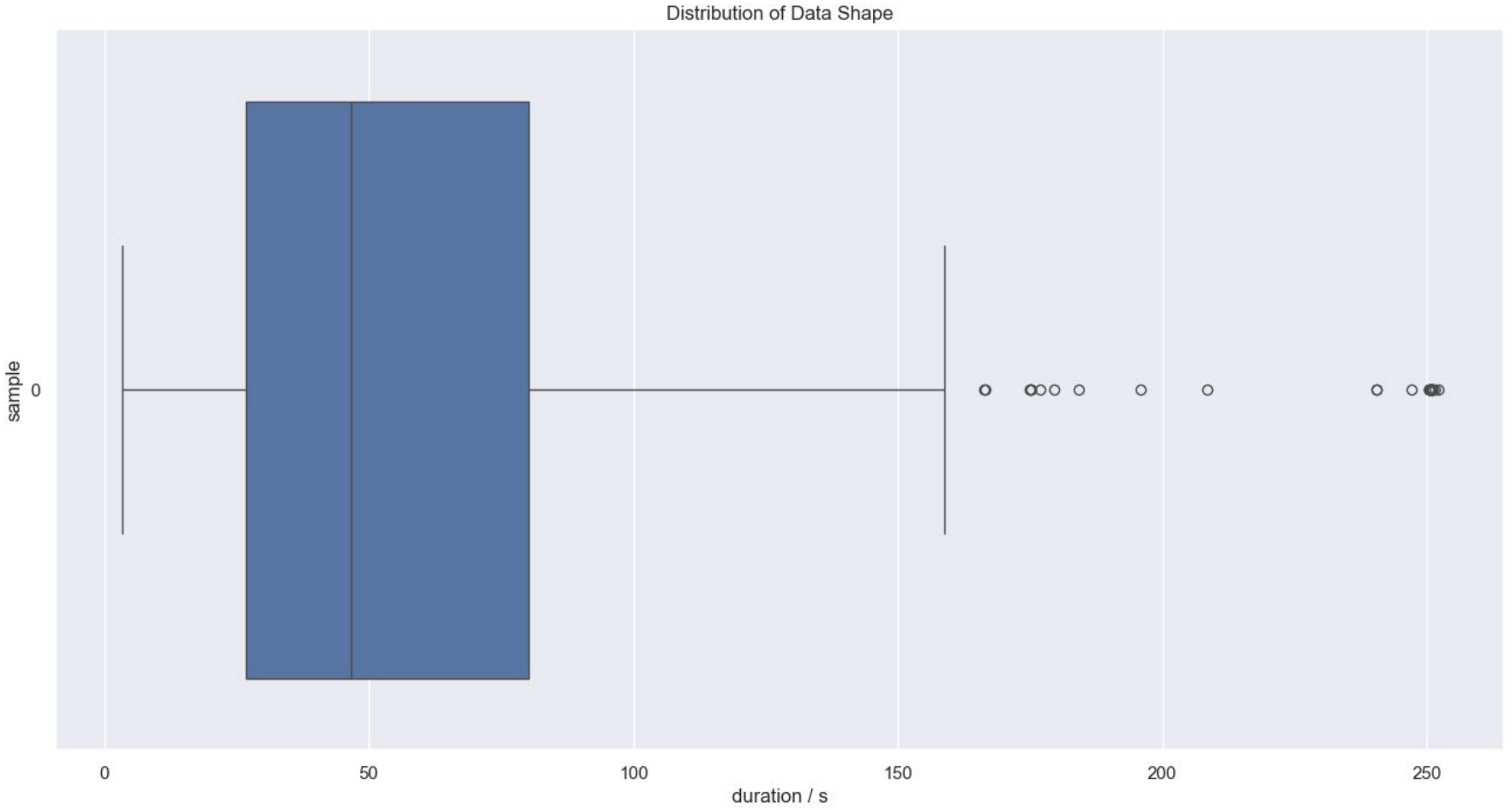
Data Processing

Data Processing - Noise Reduction





Data Processing - Standardising Duration



feature_12_128_40

feature_12_12_12

feature_0_32_13

feature_0_32_0

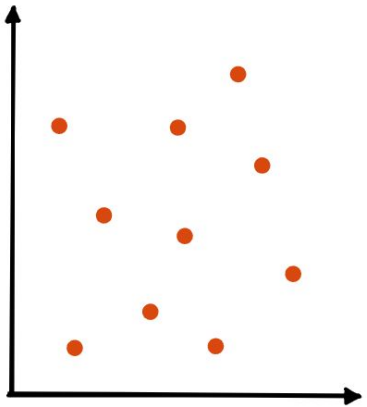
feature_0_16_13

Number of sample to feature ratio affects performance of model

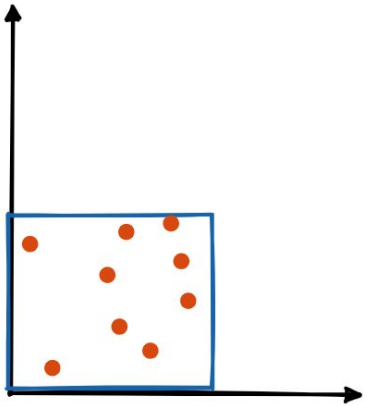
Txt file icons created by Smashicons, taken from flaticon.com

$$z = \frac{x - \mu}{\sigma}$$

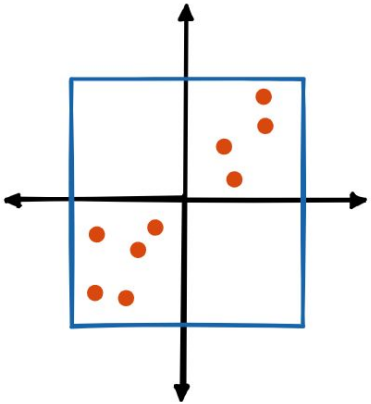
We are interested in ensuring **mean = 0**, and **standard deviation = 1**



Actual Data



Normalised Data



Standardised Data

Library used: sklearn

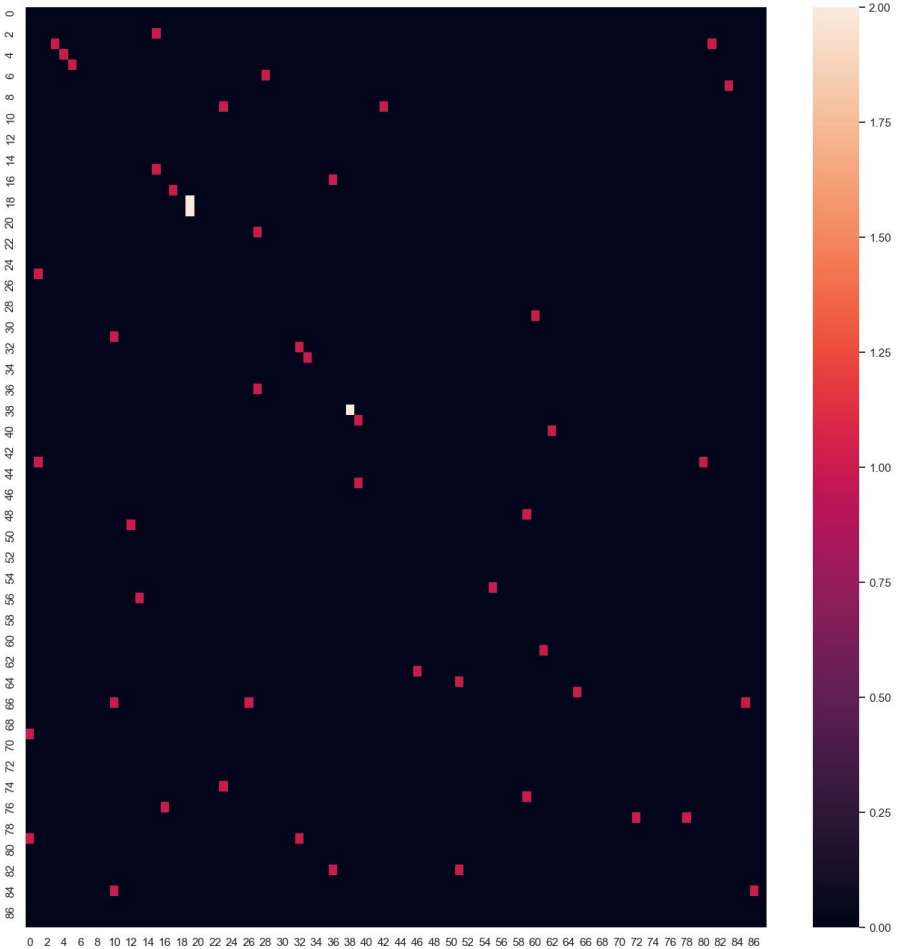
Model Testing

$$p(guess) = \frac{1}{88} \cdot 100$$
$$= 1.14\%$$

- **Optimal K:** 2
- **Train Accuracy:** 62.56%
- **Test Accuracy:** 20.75%
- **CV Score:** 21.97%

Not worth investigating KNN

Model Testing - Random Forest



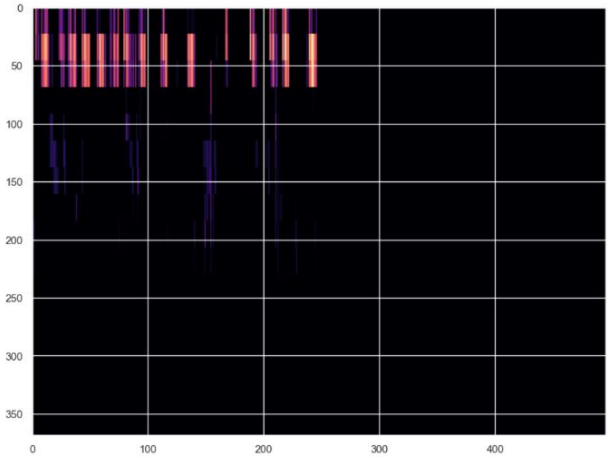
- **Train accuracy:** 100.0%
- **Test accuracy:** 22.64%
- **CV score:** 28.03%

CV score always higher than test suggests limitation in data set

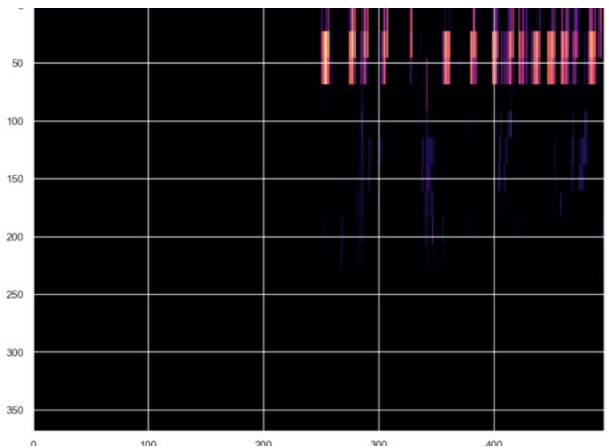
Data Processing II

Transformation: horizontal flip, noise on each sample
Initial Sample Size: 264
Final Sample Size: 792

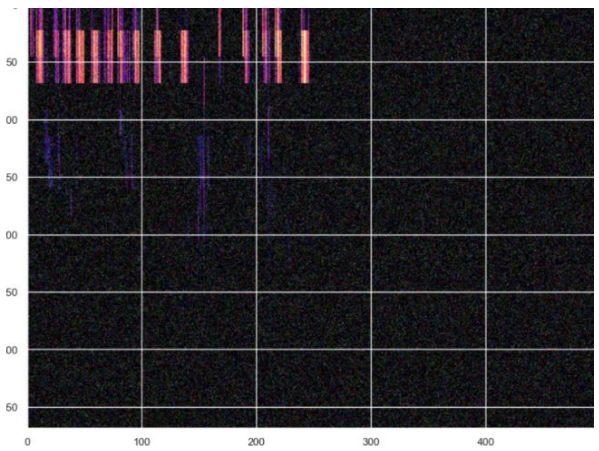
Original



Horizontal Flip



Noise



Library used: skimage

Discrete Integer Representation

Class	Label
A	0
B	1
C	2

One-hot Encoded Label

Class	Label [A, B, C]
A	[1, 0, 0]
B	[0, 1, 0]
C	[0, 0, 1]

API used: Keras

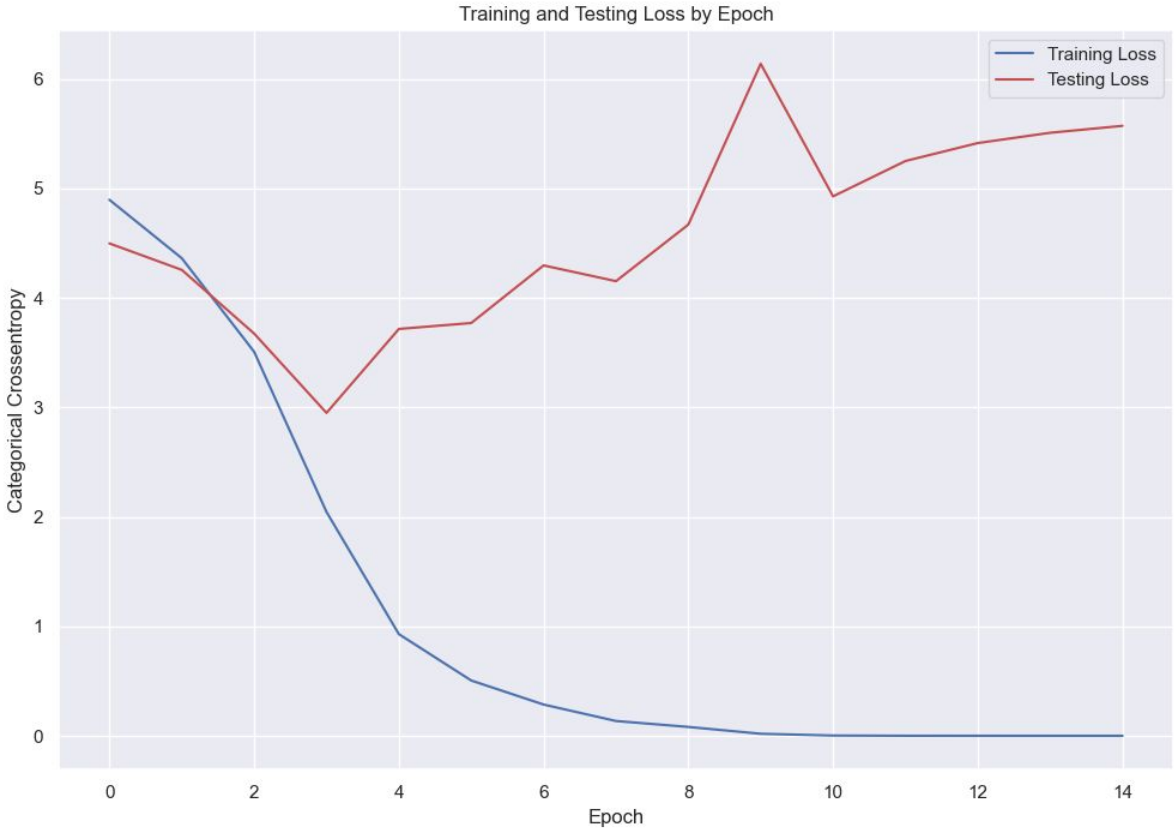
CNN Parameters

CNN Parameters - Compilation

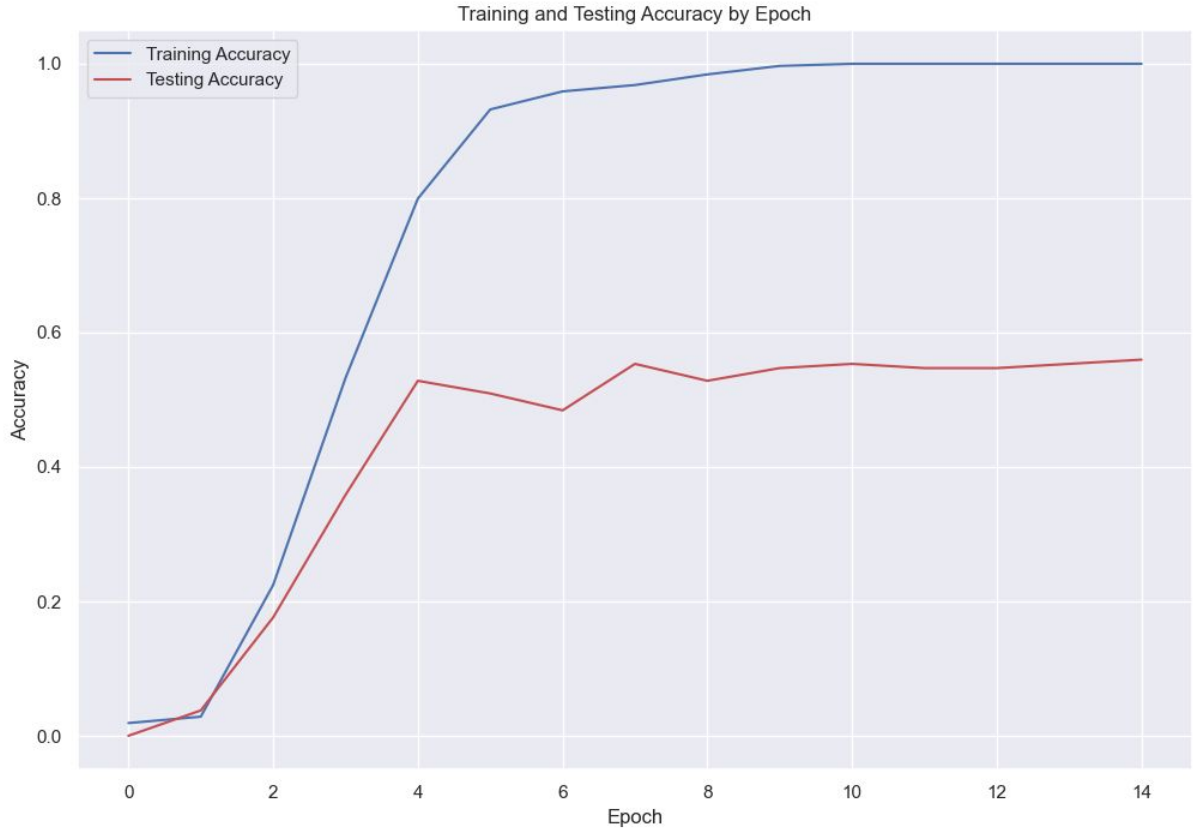
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 494, 367, 32)	1,184
max_pooling2d (MaxPooling2D)	(None, 247, 183, 32)	0
conv2d_1 (Conv2D)	(None, 245, 181, 128)	36,992
max_pooling2d_1 (MaxPooling2D)	(None, 122, 90, 128)	0
conv2d_2 (Conv2D)	(None, 120, 88, 128)	147,584
max_pooling2d_2 (MaxPooling2D)	(None, 60, 44, 128)	0
conv2d_3 (Conv2D)	(None, 58, 42, 128)	147,584
max_pooling2d_3 (MaxPooling2D)	(None, 29, 21, 128)	0
flatten (Flatten)	(None, 77952)	0
dense (Dense)	(None, 1024)	79,823,872
dense_1 (Dense)	(None, 88)	90,200

CNN Performance

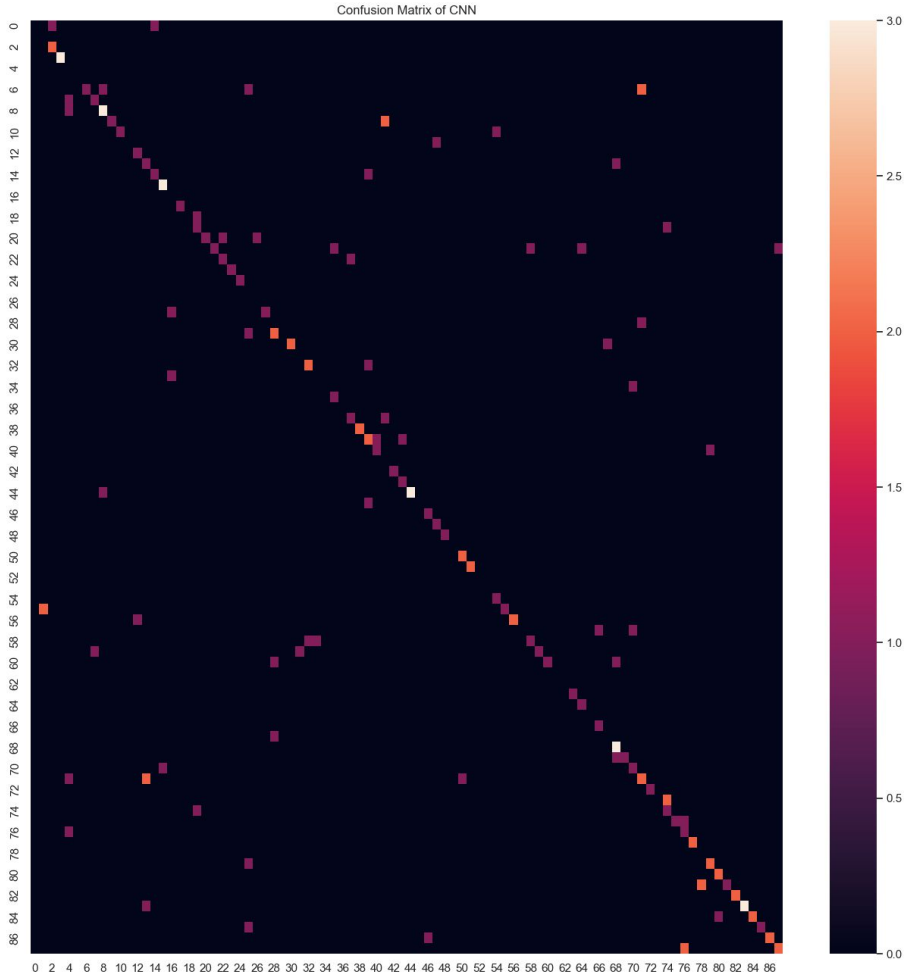
CNN - Loss Curve



CNN - Accuracy Curve



CNN - Confusion Matrix and Score Metrics



- **Test Accuracy score:** 55.97%
- **Test Precision score:** 57.58%
- **Test Recall score:** 53.52%
- **Test Set F-score score:** 51.56%

Learning Outcomes

Data Set: require much larger data set to avoid overfitting

Hyperparameter: further tuning is required

Number of Epochs: implement EarlyStop callback

Features used: experiment different domains of features