DS 203
E-7

Sound to Structure

Audio Genre Classification

Decoding the beats

Presentated by
23B2287 Kanika Sharma
23B2231 Mahima Sahu
23B2201 Dnyaneshwari Kate
23B2236 Hima Varsha

Executive Overview

Problem Statement

- What is given?
  MFCC (Mel-Frequency Cepstral Coefficients) for 116 audio files
- What is to be done?
  Classify these 116 files broadly into 6 categories
  - National Anthem
  - Bhavgeet
  - Lavni
  - Songs of Asha Bhosale
  - Songs of Kishor Kumar
  - Songs of Michael Jackson

But, But, But what exactly is this fancy word 'MFCC'?

The raw audio data of someone speaking is complex and contains a lot of information.

MFCCs help **simplify** this data

- by focusing on the features that are most important for understanding the speech
- by converting the signal into a set of numbers

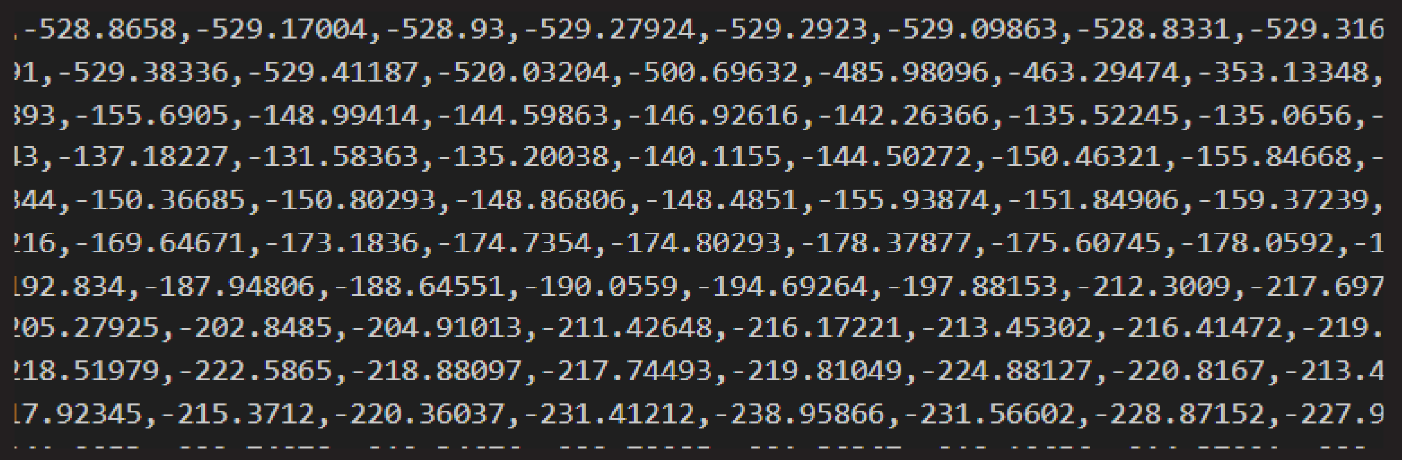
What do we want?

Numbers

(Some Magic)

Singers and Genres

The files provided look something like this



,-528.8658,-529.17004,-528.93,-529.27924,-529.2923,-529.09863,-528.8331,-529.316
91,-529.38336,-529.41187,-520.03204,-500.69632,-485.98096,-463.29474,-353.13348,
393,-155.6905,-148.99414,-144.59863,-146.92616,-142.26366,-135.52245,-135.0656,-
13,-137.18227,-131.58363,-135.20038,-140.1155,-144.50272,-150.46321,-155.84668,-
344,-150.36685,-150.80293,-148.86806,-148.4851,-155.93874,-151.84906,-159.37239,
216,-169.64671,-173.1836,-174.7354,-174.80293,-178.37877,-175.60745,-178.0592,-1
192.834,-187.94806,-188.64551,-190.0559,-194.69264,-197.88153,-212.3009,-217.697
205.27925,-202.8485,-204.91013,-211.42648,-216.17221,-213.45302,-216.41472,-219.
218.51979,-222.5865,-218.88097,-217.74493,-219.81049,-224.88127,-220.8167,-213.4
17.92345,-215.3712,-220.36037,-231.41212,-238.95866,-231.56602,-228.87152,-227.9

(pretty intimidating)

Executive Overview

How did we tackle this problem?

- 1. Performed **EDA** to understand what our data set actually entails
- 2. Extracted Complex Features from the given files using **CNN** (as just passing the raw data gives poor results)
- 3. Generated a **training data** set of 750 files
- 4. Implemented **hyper-parameter** tunning to tune the neural network as per our data-set
- 5. Implemented techniques to **reduce overfitting**
- 6. Classified the 116 files into 6 categories using our model

Data Preprocessing

Training Set Generation

Convolutional Neural Network

Predictions

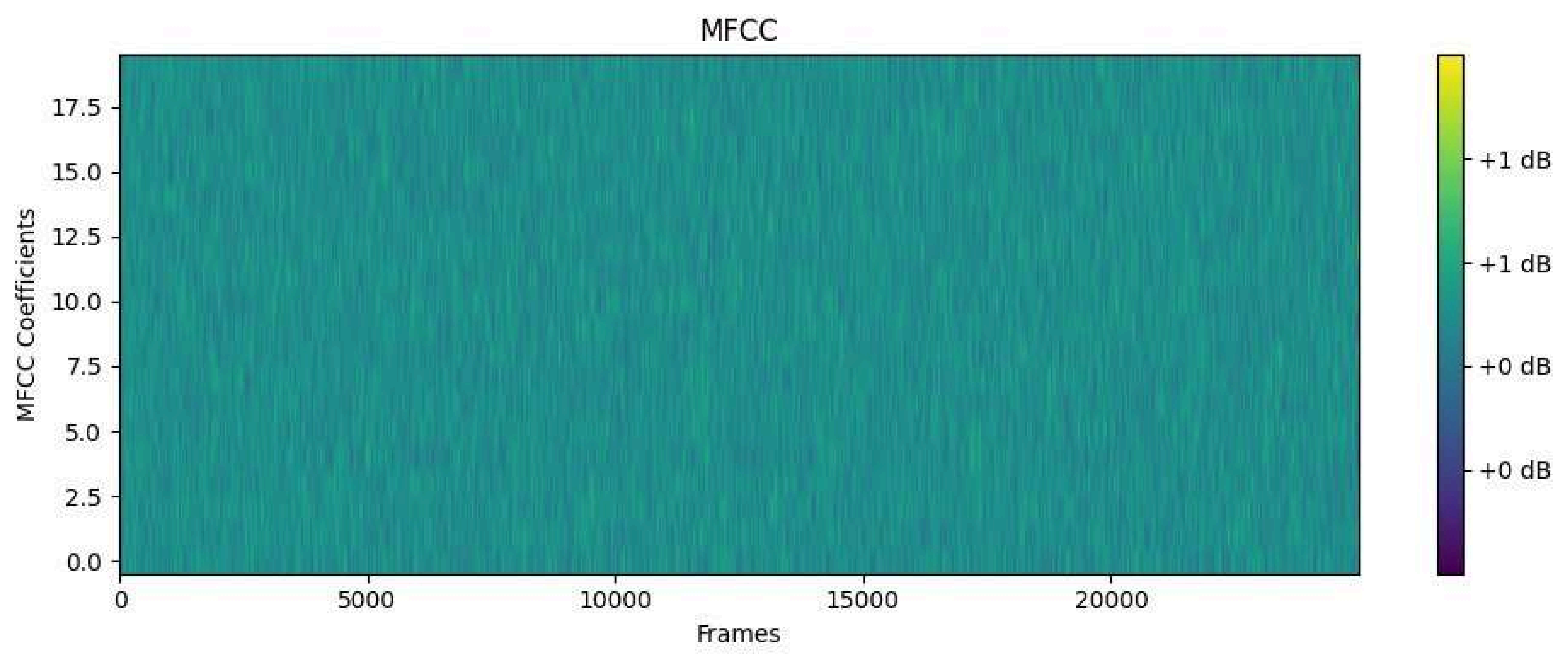
Why choose CNN?

Isn't it used for images only? nope!

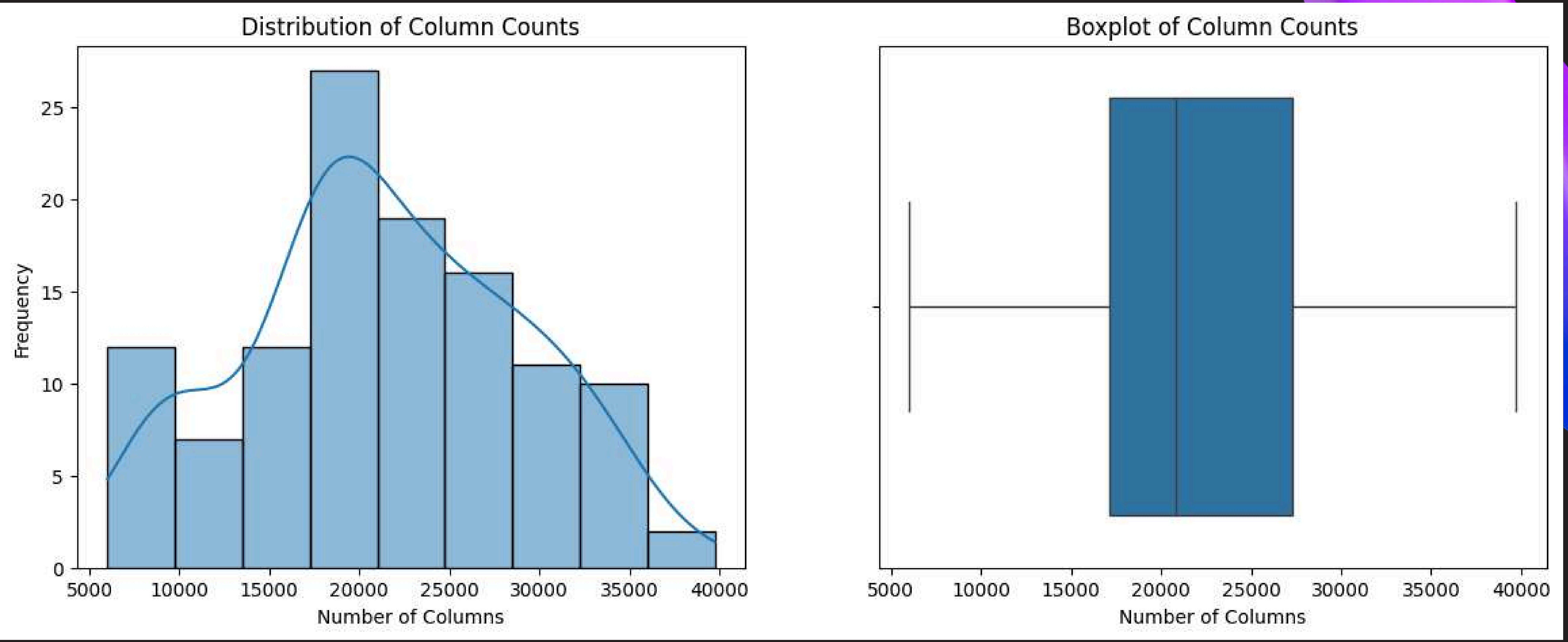
- CNNs learn **hierarchical** representations of data, which can be applied to our data set as to capture,
    - **low-level features** (e.g., pitch, timbre) and
    - **high-level features** (e.g., phonemes, words).
- CNNs recognize patterns regardless of their position in the input.
  Which is especially handy for our classification where we need to essentially identify the voices of 3 different singers.
- This is a **supervised learning** approach, unsupervised learning models won't perform well on the given data as it
    - lacks clear structure and
    - has high dimensionality.

Exploratory Data Analysis

- **EDA analysis** on test files was done to check Missing values.
  We didnt find any missing values in the our dataset of mfcc csv files.
- This is a **spectrogram**, with time on the x-axis and frequency on the y-axis. The color of each point in the spectrogram represents the **intensity of the signal** at that point



Column Distribution




```
Column Count Statistics:
25th Percentile: 17155.25
Median (50th Percentile): 20844.5
75th Percentile: 27268.5
90th Percentile: 31972.0
Interquartile Range (IQR): 10113.25
```

We considered the **Median** and **75th Percentile** value and choose **25000** as fixed column size for our all mfcc files to do mean max pooling.

Data Pre-Processing

File sizes of all the test files were analysed. There were a lot of **variations**.

Mean-Max Pooling

- By considering the mean, median and 75th percentile , we chose **20*25000** as the fixed size for Mean-max pooling
- Max pooling emphasizes the **most important features** by retaining the maximum values
- Mean pooling helps retain **global** contextual information by **averaging**

Model Architecture

Main Model Layers

- 01 1D Convolutional Layer w/ 32 filters
- 02 1D Convolutional Layer w/ 64 filters
- 03 1D Convolutional Layer w/ 128 filters
- 04 Dense Layer with 128 neurons
- 05 Dense Layer with 6 neurons

Layer (type)	Output Shape	Param #
conv1d_3 (Conv1D)	(None, 20, 32)	6,400,032
batch_normalization_4 (BatchNormalization)	(None, 20, 32)	128
max_pooling1d_3 (MaxPooling1D)	(None, 10, 32)	0
dropout_4 (Dropout)	(None, 10, 32)	0
conv1d_4 (Conv1D)	(None, 10, 64)	10,304
batch_normalization_5 (BatchNormalization)	(None, 10, 64)	256
max_pooling1d_4 (MaxPooling1D)	(None, 5, 64)	0
dropout_5 (Dropout)	(None, 5, 64)	0
conv1d_5 (Conv1D)	(None, 5, 128)	41,088
batch_normalization_6 (BatchNormalization)	(None, 5, 128)	512
max_pooling1d_5 (MaxPooling1D)	(None, 3, 128)	0
dropout_6 (Dropout)	(None, 3, 128)	0
flatten_1 (Flatten)	(None, 384)	0
dense_2 (Dense)	(None, 128)	49,280
batch_normalization_7 (BatchNormalization)	(None, 128)	512
dropout_7 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 6)	774

Categorical Crossentropy Loss Function

- Calculates the negative logarithm of the predicted probability for the correct class, penalizing incorrect predictions
- The loss is always non-negative, with **zero indicating perfect accuracy** and increasing as predictions become less accurate

ReduceLROnPlateau

- Used to **reduce the learning rate** when a monitored metric has stopped improving
- It automatically adjusts the learning rate without requiring manual intervention
- It prevents the optimizer from overshooting the optimal solution

HOW DID WE TRAIN THIS MODEL?

We downloaded 50 songs for each of the 6 song categories from the internet. (total **300** songs)

- Converted .mp3 to .csv, which contains the MFCC coefficients

BUT,

Neural Networks typically require a lot of training data.
We used data augmentation to generate **750** MFCC coefficient files.

What's done in **Data Augmentation**?

- By augmenting MFCC data, we can increase the dataset size and bring up diversity.
- **Scaling**, **shifting**, and **removing** frames were used to augment the data.

Encoding Technique Used: **Binary Encoding** - for target variables

The final training data set after pre-processing and data augmentation:

Click here for training dataset

Divide

750 files

450 **150** **150**

But why??

three words train, test, validation

VALIDATION SET

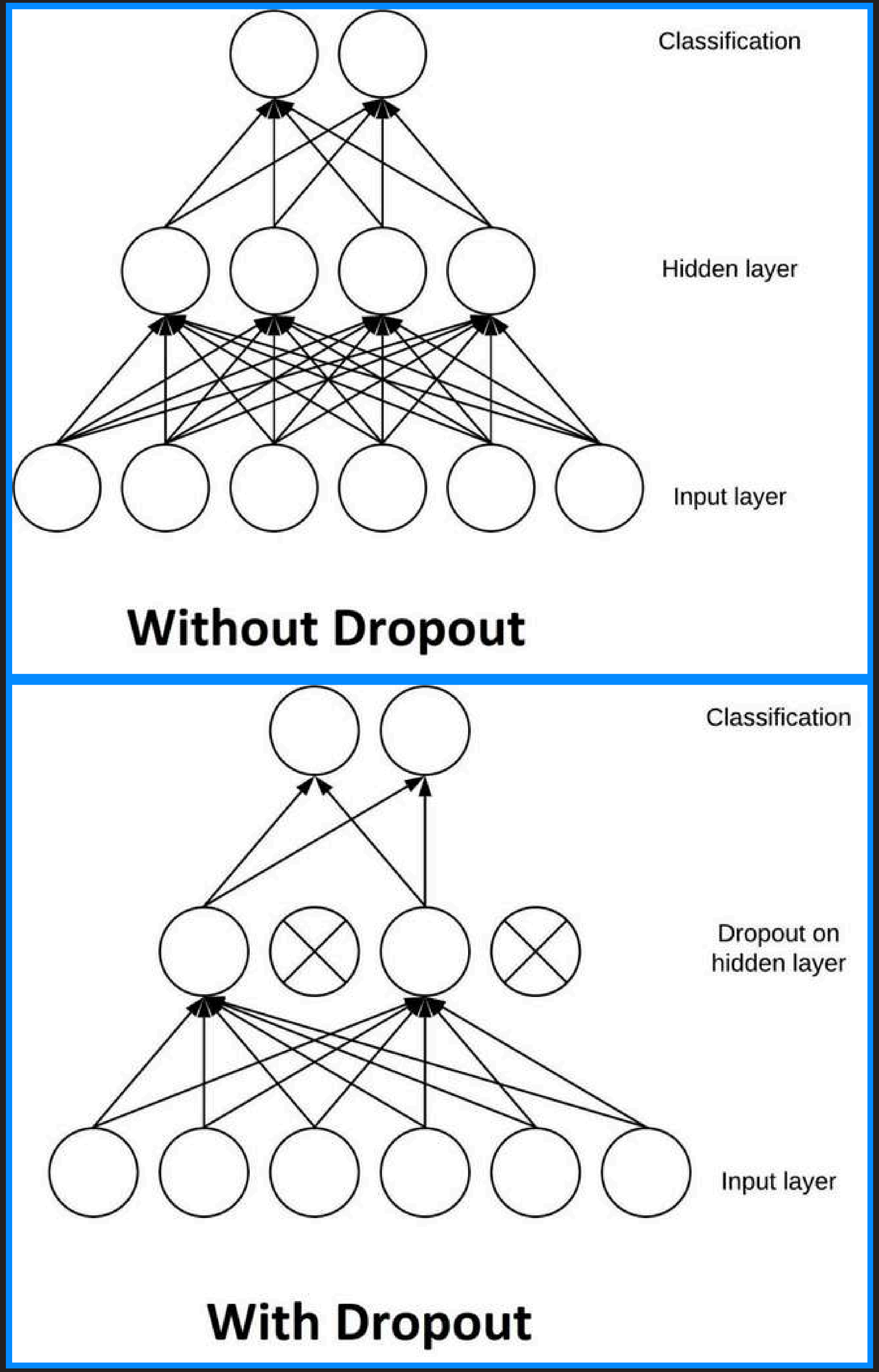
- Helps to evaluate the model's performance on data that it has **not seen during training**
- Provides **unbiased** evaluation of model’s performance
- Helps in **tunning the hyperparameters**
    - number of layers
    - learning rate
    - filters in each convolutional layer
    - kernel size of each convolutional layer

TACKLING OVERFITTING

1. **Batch Normalization**
2. **Dropout Layers**
3. **Early Stopping**

1. BATCH NORMALIZATION

- Used to improve the training of deep neural networks by normalizing the inputs of each layer
- Makes the network more robust to variations in the input data



2. DROPOUT

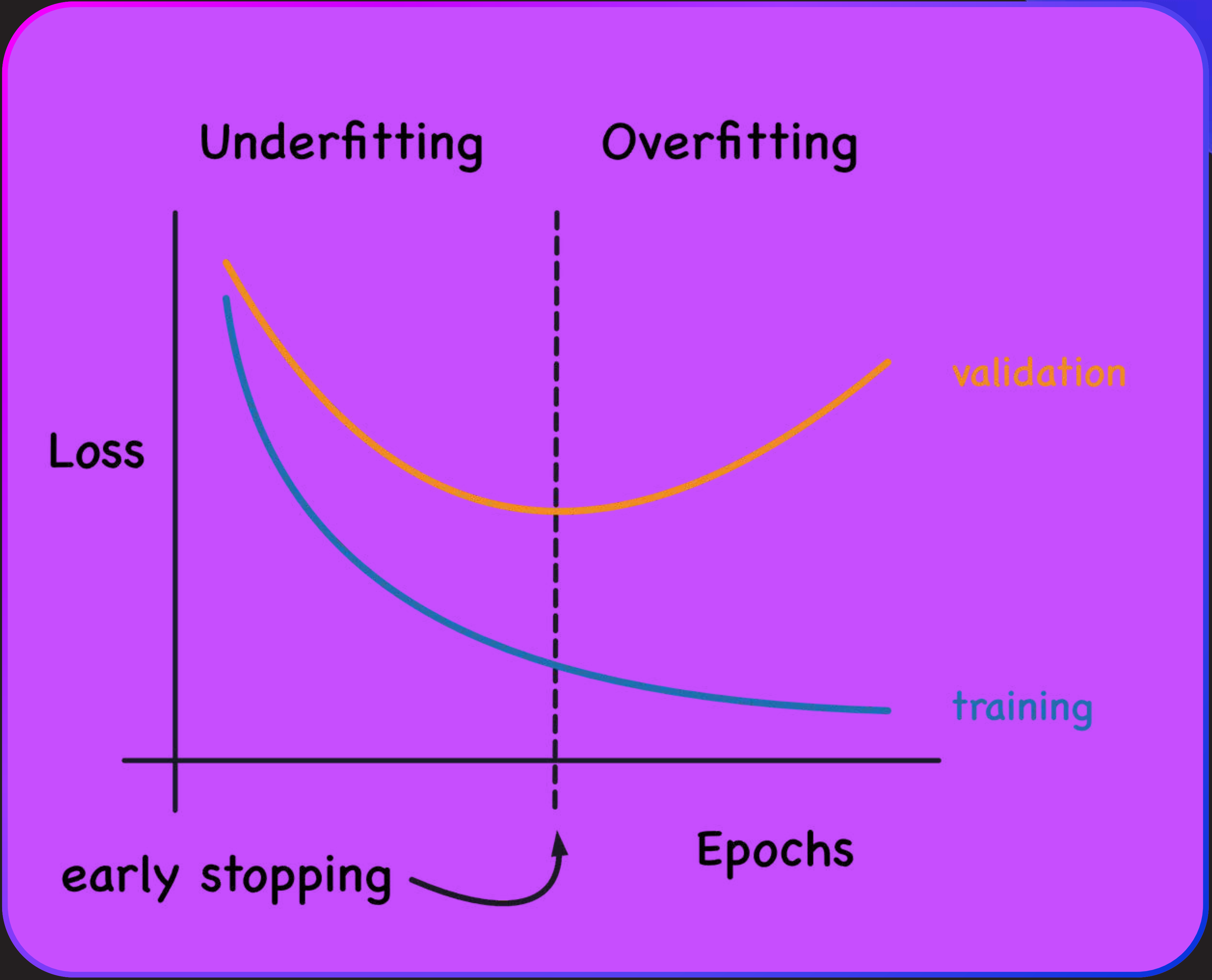
Works by randomly "dropping out" (i.e, setting to zero) a fraction of the neurons during the training process

This forces the network to learn more robust features that are not reliant on specific neurons

Improves generalization to new data

3. EARLY STOPPING

- Works by **monitoring the performance** of the model on a validation set
- Stops training if the metric does not improve for a specified number of **epochs**
- The metric used here to monitor the performance on the validation set is validation accuracy
- Prevents unnecessary training and **saves time**
- Restores the **best model weights**



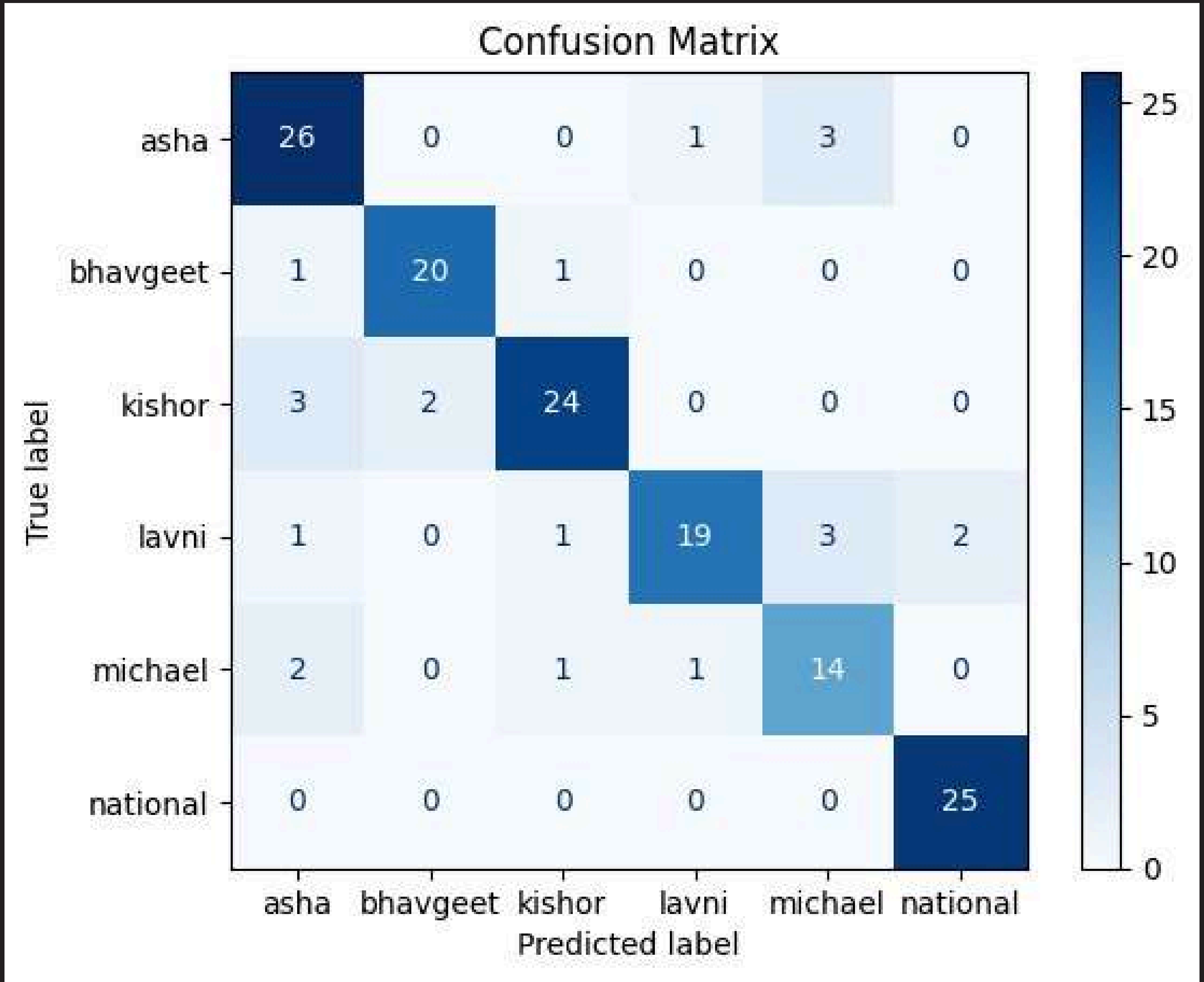
Model Metrics

```
Accuracy: 0.8533333333333334
Precision: 0.8579073272406607
Recall: 0.8533333333333334
F1 Score: 0.8529462909866046
```

	precision	recall	f1-score	support
asha	0.79	0.87	0.83	30
bhavgeet	0.91	0.91	0.91	22
kishor	0.89	0.83	0.86	29
lavni	0.90	0.73	0.81	26
michael	0.70	0.78	0.74	18
national	0.93	1.00	0.96	25
accuracy			0.85	150
macro avg	0.85	0.85	0.85	150
weighted avg	0.86	0.85	0.85	150

- We got a **accuracy of 0.85**, suggesting that the model is giving us appropriate results.
- The values aren't very close to 1, so we can conclude that model **isn't overfitting.**
- Different songs have have different lyrics but **National Anthem** has same for all, so it has high metric measures.

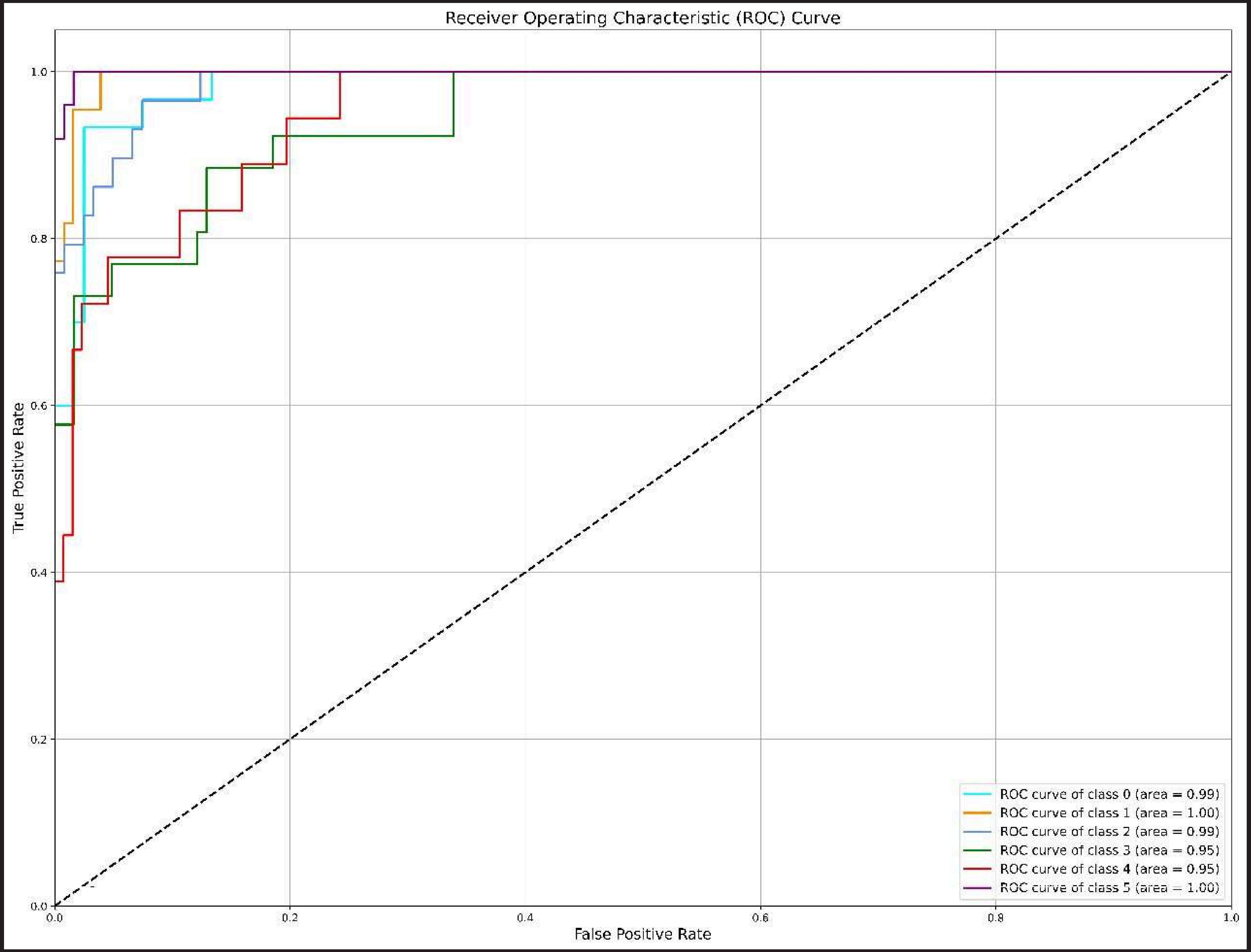
Confusion Matrix




```
Confusion Matrix:
[[26  0  0  1  3  0]
 [ 1 20  1  0  0  0]
 [ 3  2 24  0  0  0]
 [ 1  0  1 19  3  2]
 [ 2  0  1  1 14  0]
 [ 0  0  0  0  0 25]]
```

- **High diagonal values** suggest good accuracy for each class
- Low values are shown for **michael** indicationg that the model is not able to classify the songs of michael jackson
- This could be because there are **more hindi/marathi** songs as compared to english ones as our model is picking up on words

ROC Curve

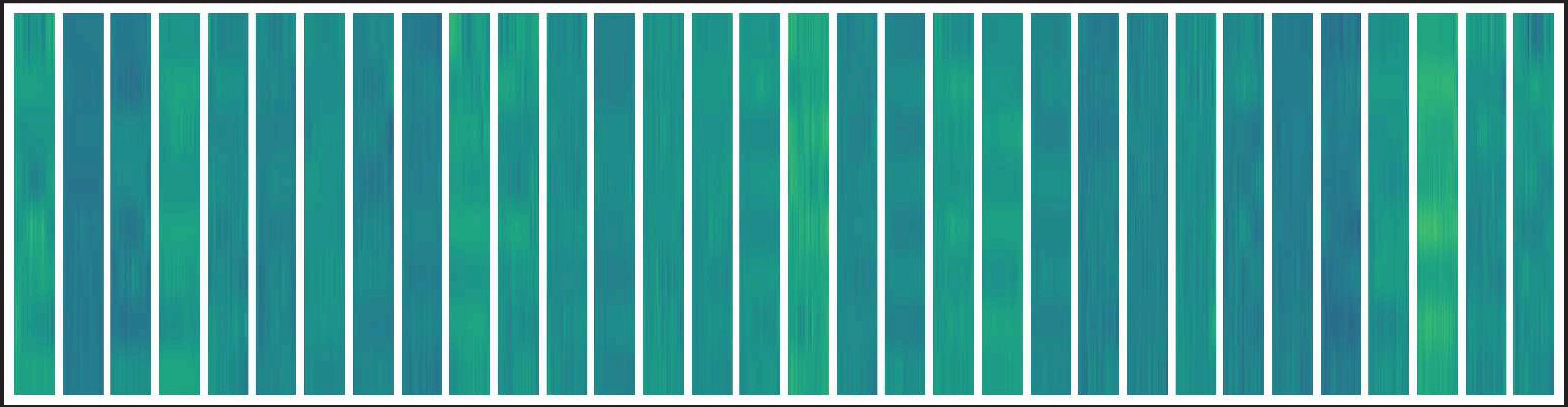



```
AUC for class 0: 0.99
AUC for class 1: 1.00
AUC for class 2: 0.99
AUC for class 3: 0.95
AUC for class 4: 0.95
AUC for class 5: 1.00
Micro-average AUC: 0.98
```

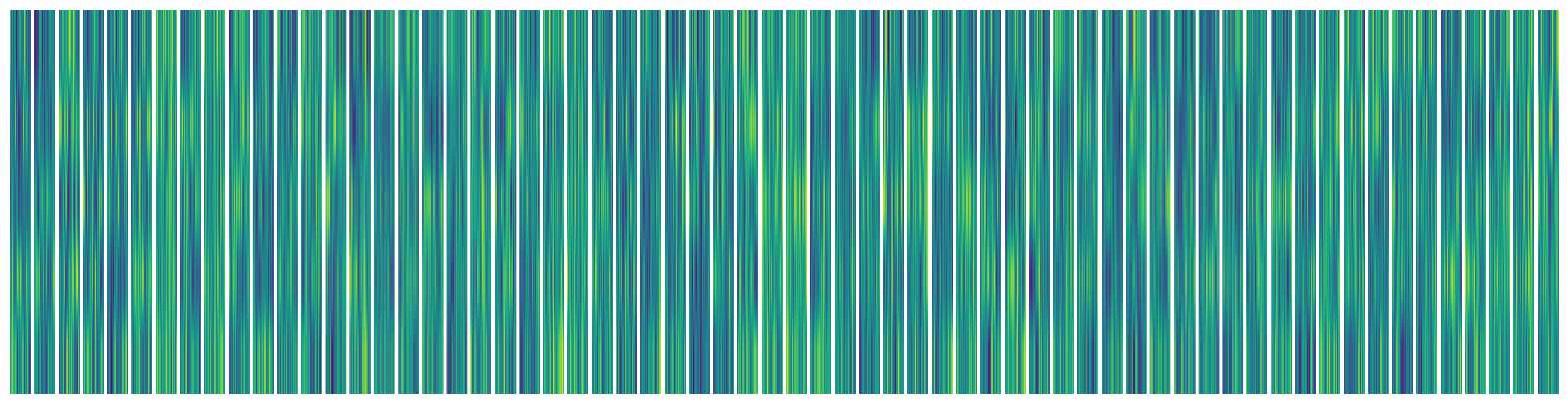
The auc values are close to **1** which indicates that the model is performing well

Visualization of what the CNN layers are actually doing??

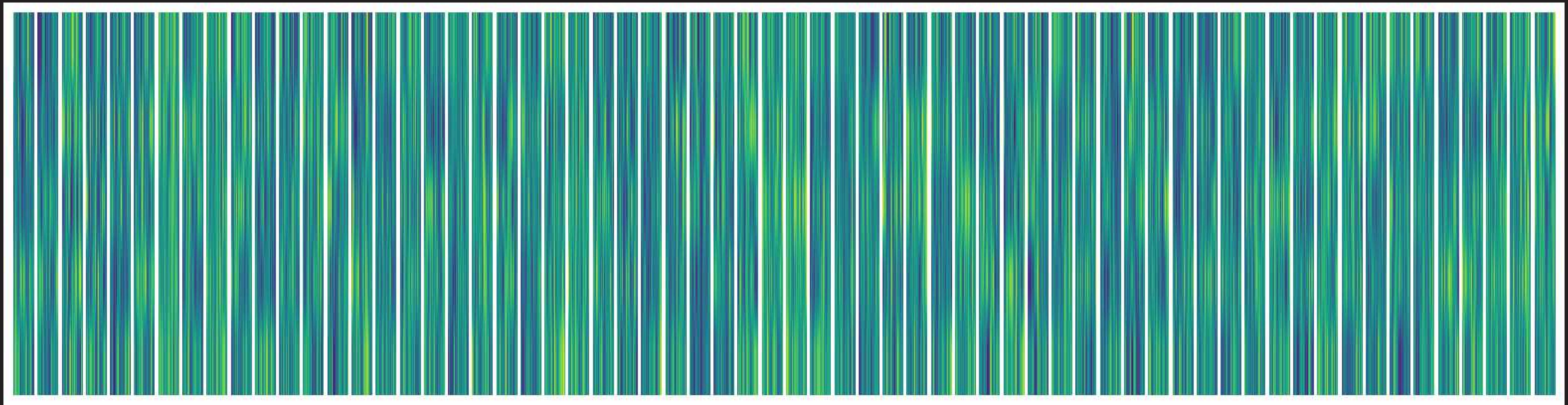
1st Convolutional Layer w/ 32 filters



2nd Convolutional Layer w/ 64 filters



3rd Convolutional Layer w/ 64 filters



Analysis of Graphs

- **Feature Extraction:** Each vertical line represent a specific feature map generated after applying filters in a CNN layer.
- **Activation Patterns:** High activation regions (bright yellow-green areas) indicate areas where the CNN detects prominent features.
- **Impact of Layers:** The maps demonstrate how each CNN Layer refines features from previous layers. (Hierarchial learning process)

Predictions of the Model

```
File_Name,Category
01-MFCC.csv,national
02-MFCC.csv,national
03-MFCC.csv,michael
04-MFCC.csv,asha
05-MFCC.csv,kishor
06-MFCC.csv,asha
07-MFCC.csv,kishor
08-MFCC.csv,asha
09-MFCC.csv,kishor
10-MFCC.csv,asha
100-MFCC.csv,asha
101-MFCC.csv,kishor
102-MFCC.csv,bhavgeet
103-MFCC.csv,michael
104-MFCC.csv,asha
105-MFCC.csv,asha
106-MFCC.csv,bhavgeet
107-MFCC.csv,national
108-MFCC.csv,national
109-MFCC.csv,kishor
11-MFCC.csv,asha
110-MFCC.csv,asha
111-MFCC.csv,bhavgeet
112-MFCC.csv,lavni
113-MFCC.csv,asha
114-MFCC.csv,asha
115-MFCC.csv,lavni
116-MFCC.csv,national
```

```
File_Name,Category
12-MFCC.csv,asha
13-MFCC.csv,asha
14-MFCC.csv,kishor
15-MFCC.csv,asha
16-MFCC.csv,national
17-MFCC.csv,national
18-MFCC.csv,kishor
19-MFCC.csv,lavni
20-MFCC.csv,asha
21-MFCC.csv,michael
22-MFCC.csv,kishor
23-MFCC.csv,asha
24-MFCC.csv,kishor
25-MFCC.csv,bhavgeet
26-MFCC.csv,bhavgeet
27-MFCC.csv,national
28-MFCC.csv,kishor
29-MFCC.csv,bhavgeet
30-MFCC.csv,asha
31-MFCC.csv,national
32-MFCC.csv,asha
33-MFCC.csv,lavni
34-MFCC.csv,kishor
35-MFCC.csv,national
36-MFCC.csv,michael
37-MFCC.csv,asha
38-MFCC.csv,bhavgeet
39-MFCC.csv,lavni
```

```
File_Name,Category
40-MFCC.csv,asha
41-MFCC.csv,lavni
42-MFCC.csv,bhavgeet
43-MFCC.csv,lavni
44-MFCC.csv,asha
45-MFCC.csv,asha
46-MFCC.csv,kishor
47-MFCC.csv,lavni
48-MFCC.csv,lavni
49-MFCC.csv,bhavgeet
50-MFCC.csv,bhavgeet
51-MFCC.csv,kishor
52-MFCC.csv,asha
53-MFCC.csv,asha
54-MFCC.csv,kishor
55-MFCC.csv,kishor
56-MFCC.csv,kishor
57-MFCC.csv,lavni
58-MFCC.csv,kishor
59-MFCC.csv,kishor
60-MFCC.csv,asha
61-MFCC.csv,national
62-MFCC.csv,lavni
63-MFCC.csv,kishor
64-MFCC.csv,asha
65-MFCC.csv,kishor
66-MFCC.csv,national
67-MFCC.csv,national
```

```
File_Name,Category
68-MFCC.csv,kishor
69-MFCC.csv,asha
70-MFCC.csv,asha
71-MFCC.csv,asha
72-MFCC.csv,lavni
73-MFCC.csv,asha
74-MFCC.csv,asha
75-MFCC.csv,national
76-MFCC.csv,kishor
77-MFCC.csv,lavni
78-MFCC.csv,lavni
79-MFCC.csv,lavni
80-MFCC.csv,asha
81-MFCC.csv,national
82-MFCC.csv,asha
83-MFCC.csv,kishor
84-MFCC.csv,kishor
85-MFCC.csv,lavni
86-MFCC.csv,michael
87-MFCC.csv,national
88-MFCC.csv,lavni
89-MFCC.csv,asha
90-MFCC.csv,national
91-MFCC.csv,lavni
92-MFCC.csv,kishor
93-MFCC.csv,kishor
94-MFCC.csv,bhavgeet
95-MFCC.csv,national
96-MFCC.csv,bhavgeet
97-MFCC.csv,bhavgeet
98-MFCC.csv,michael
99-MFCC.csv,lavni
```

Files Containing the National Anthem

National Anthem
01-MFCC.csv
02-MFCC.csv
87-MFCC.csv

Solo Songs

Asha Bhosale	Kishor Kumar	Michael Jackson
06-MFCC.csv	05-MFCC.csv	03-MFCC.csv
30-MFCC.csv	18-MFCC.csv	98-MFCC.csv
60-MFCC.csv	46-MFCC.csv	103-MFCC.csv

Optional Problem Statement

Problem Statement: Need to classify the given songs into songs of **female singers**, **male singers** and **both**

Our Approach:

- Created a training dataset 799 files containing
  - 300 songs of female singers,
  - 300 of male singers and
  - 199 songs of both

  using the methods disccused earlier

- Used the CNN model we had built for the previous problem statement after **fine-tunning** to classify the songs

The **final training data set** after pre-processing and data augmentation: click here

Model Metrics

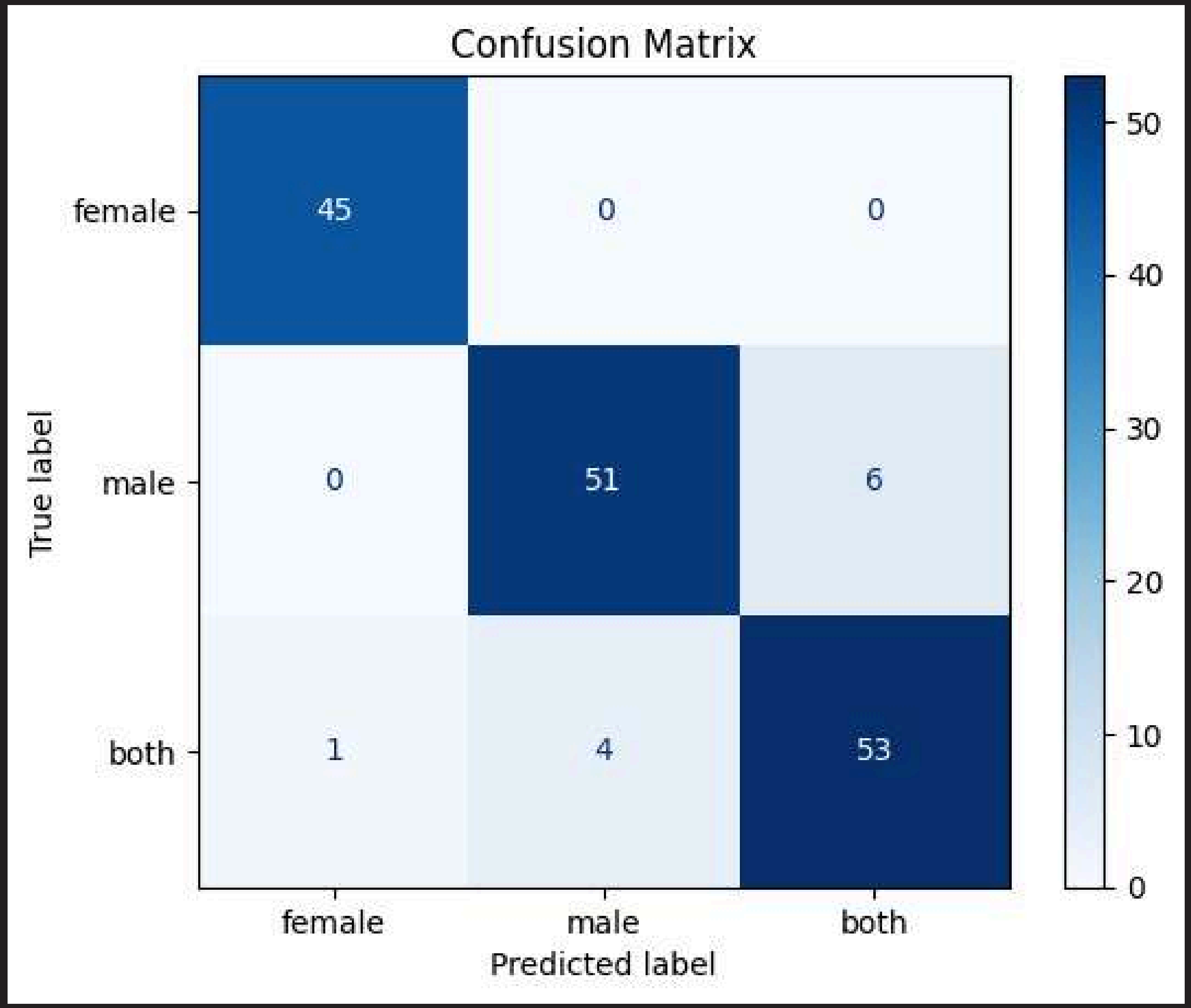
```
Accuracy: 0.93125
Precision: 0.9311123718764656
Recall: 0.93125
F1 Score: 0.9310201083638583
              precision    recall  f1-score   support

        both       0.98      1.00      0.99        45
      female       0.93      0.89      0.91        57
        male       0.90      0.91      0.91        58

    accuracy                           0.93       160
   macro avg       0.93      0.94      0.94       160
weighted avg       0.93      0.93      0.93       160
```

- We have got a **high accuracy** of **0.93**, suggesting that our model has given us good results.
- The Precision and Recall values, suggest that our model was **well trained** and could **recall** very well to make predictions.
- We have got high metric values for the **‘both**’ category, so our model works well with predictions related to songs of both male and female singers.

Confusion Matrix




```
Confusion Matrix:
[[45  0  0]
 [ 0 51  6]
 [ 1  4 53]]
```

- The matrix **summarizes** the performance of our machine learning model on set of test data.
- **High diagonal** values indicate good accuracy for each class

Predictions of the Model for Female-Male Classification

01-MFCC.csv	male
03-MFCC.csv	male
04-MFCC.csv	male
05-MFCC.csv	male
07-MFCC.csv	male
08-MFCC.csv	male
09-MFCC.csv	male
100-MFCC.csv	male
101-MFCC.csv	male
107-MFCC.csv	male
113-MFCC.csv	male
114-MFCC.csv	male
116-MFCC.csv	male
14-MFCC.csv	male
15-MFCC.csv	male
16-MFCC.csv	male
18-MFCC.csv	male
19-MFCC.csv	male
20-MFCC.csv	male
21-MFCC.csv	male
24-MFCC.csv	male
31-MFCC.csv	male
32-MFCC.csv	male
33-MFCC.csv	male
34-MFCC.csv	male
36-MFCC.csv	male
44-MFCC.csv	male
45-MFCC.csv	male

46-MFCC.csv	male
52-MFCC.csv	male
53-MFCC.csv	male
54-MFCC.csv	male
55-MFCC.csv	male
57-MFCC.csv	male
58-MFCC.csv	male
59-MFCC.csv	male
61-MFCC.csv	male
62-MFCC.csv	male
63-MFCC.csv	male
65-MFCC.csv	male
66-MFCC.csv	male
67-MFCC.csv	male
68-MFCC.csv	male
71-MFCC.csv	male
74-MFCC.csv	male
76-MFCC.csv	male
77-MFCC.csv	male
78-MFCC.csv	male
86-MFCC.csv	male
89-MFCC.csv	male
92-MFCC.csv	male
93-MFCC.csv	male
96-MFCC.csv	male
98-MFCC.csv	male

02-MFCC.csv	female
06-MFCC.csv	female
10-MFCC.csv	female
102-MFCC.csv	female
103-MFCC.csv	female
104-MFCC.csv	female
105-MFCC.csv	female
106-MFCC.csv	female
108-MFCC.csv	female
109-MFCC.csv	female
11-MFCC.csv	female
110-MFCC.csv	female
112-MFCC.csv	female
115-MFCC.csv	female
12-MFCC.csv	female
13-MFCC.csv	female
17-MFCC.csv	female
22-MFCC.csv	female
23-MFCC.csv	female
25-MFCC.csv	female
26-MFCC.csv	female
27-MFCC.csv	female
28-MFCC.csv	female
29-MFCC.csv	female
35-MFCC.csv	female
37-MFCC.csv	female
38-MFCC.csv	female
39-MFCC.csv	female
40-MFCC.csv	female
41-MFCC.csv	female
42-MFCC.csv	female
43-MFCC.csv	female
47-MFCC.csv	female

48-MFCC.csv	female
49-MFCC.csv	female
51-MFCC.csv	female
56-MFCC.csv	female
60-MFCC.csv	female
64-MFCC.csv	female
69-MFCC.csv	female
70-MFCC.csv	female
72-MFCC.csv	female
73-MFCC.csv	female
75-MFCC.csv	female
79-MFCC.csv	female
80-MFCC.csv	female
81-MFCC.csv	female
84-MFCC.csv	female
85-MFCC.csv	female
87-MFCC.csv	female
88-MFCC.csv	female
90-MFCC.csv	female
91-MFCC.csv	female
94-MFCC.csv	female
95-MFCC.csv	female
99-MFCC.csv	female

111-MFCC.csv	both
30-MFCC.csv	both
50-MFCC.csv	both
82-MFCC.csv	both
83-MFCC.csv	both
97-MFCC.csv	both

Learnings and Hurdles faced in the project

- We initially tried to solve the problem using **unsupervised** learning but the clusters and predictions weren't accurate enough.
- We tried to use **RBM** layers with CNN model but couldn't get good results as it requires proper initialization of weights.
- Explored different ways to improve the **accuracy** of CNN model, some of them are mentioned after the metrics table in slides.
- Learned about the concepts behind **mffc generation** and how features can be extracted from it.
- We tried **target and binary encoding** techniques. Binary gave us better results in this problem.
- Creation of Large Training data is a important aspect in building a model, we first downloaded songs and then made more data by **data augmentation**.
- We got only 6 songs classified as Michael Jackson because our model was **heavily trained** on hindi songs as compared to english.
- Explored the use of **spark** for big data analysis.

Thank you