REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

MUSIC GENRE CLASSIFICATION WITH DEEP LEARNING AND TRANSFER LEARNING

17011052 — Hüsrev YUMUŞAK
18011615 — Abdurrahman Ebrar YÜCEL

SENIOR PROJECT

Advisor
Dr. Ahmet ELBIR

JUNE, 2021

ACKNOWLEDGEMENTS

We can't give enough thanks to our project teacher Dr. Ahmet Elbir for his great and continuous support during this project.

He leaded and lighted the way for us by doing pre-teachings and solving the problems which we can't handle by ourselves.

Hüsrev YUMUŞAK
Abdurrahman Ebrar YÜCEL

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

BS	Bark Scale
CNN	Convolutional Neural Networks
MS	Mel Scale
STFT	Short Time Fourier Transform
TL	Transfer Learning

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

MUSIC GENRE CLASSIFICATION WITH DEEP LEARNING AND TRANSFER LEARNING

Hüsrev YUMUŞAK
Abdurrahman Ebrar YÜCEL

Department of Computer Engineering
Senior Project

Advisor: Dr. Ahmet ELBIR

Music refers to harmonic sounds which is in a regular and planned order. It is an art and profession which inspires and enjoys humankind. Music helps to keep brain active and ready. It improves the capability of memory. Music has potential to improve the human spirit and health.

Music is one of cultural universal aspects of all societies. General definitions of music include common elements such as pitch (which governs melody and harmony), rhythm (and its associated concepts tempo, meter, and articulation), dynamics (loudness and softness), and the sonic qualities of timbre and texture (which are sometimes termed the "color" of a musical sound). Different styles or types of music may emphasize, de-emphasize or omit some of these elements. The difference in people's musical tastes and some other reasons has increased the importance of music genres and led to the emergence of music genre classification systems.

So during this project after extracting spectrogram images of musics, genre classification will be made with Convolutional Neural Networks.

Keywords: Convolutional Neural Network, Deep Learning, Neural Networks, Voice Spektogram, Music, Genre, Classification

1
Introduction

1.1 Literature Review

Music Classification Problem is an interesting and popular topic and many academicians study on it. Some studies approaches to problem with machine learning methods[1]. These machine learning methods are k-nearest neighbor (kNN), k-means, multi-class SVM, and neural networks. Some other studies based on Daubechies Wavelet Coefficient Histograms (DWCHs) for doing music genre classification[2].

Some studies exist which has partly similarity with our project[3]. Common part is using Convolutional Neural Networks. Our project differentiates with usage of multiple feature extracting methods such as Mel Spectrogram, Bark Spectrogram and LPC (Linear Prediction Coefficients); also our project will include machine learning algorithms with pre-trained CNN models (Transfer Learning).

1.2 About Project

After extracting Spectrogram Images, We will be doing Music Genre classification by using Convolutional Neural Networks during the Project. On this purpose; after model training, Genre Classification will be made on Test Data.

Transfer Learning will be used with machine learning algoirthms during rhe project. By doing this, it is aimed to get higher accuracy results. Transfer learning means reusing model weights to solve classification problem[4].

Python Programming Language will be used with Librosa and Keras libraries during this project. Project will be runned on Google Colab Environment which provides free GPU source.

GTZAN dataset is used in this project. This data set includes musics with 10 different genres. Spectrogram, mel spectrogram, bark spectrogram, LPC are obtained from these samples and used as input for classification models.

1.3 Report Sections

This report includes Feasibility chapter, System Analysis chapter, System Design and Experimental results chapter and Conclusion chapter. In System Analysis chapter, used methods will be explained. In System Design data set structure and train and test splitting program will be mentioned. In Experimental Results chapter results will be shown. There will be a summary section in conclusion chapter.

2
Feasibilities

This section includes the technical feasibility, hardware feasibility, legal feasibility, economic feasibility and time feasibility of the project.

2.1 Technical Feasibility

Python used as the programming language in this project, because it has useful libraries for machine learning and sound analysis.

Python libriaries are used such as Matplotlib, NumPy, Librosa and Essentia.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations. NumPy is a library that brings the computational power of languages like C to Python.

Librosa package will be used for music and audio analysis. Pycharm IDE and Google Colab technologies are used during this project.

2.2 Hardware Feasibility

The features of the computers used in the project are as follows:

Table 2.1 Computer of student 1

Donanım	**Model**
CPU	Intel Core i7-4720HQ 2.6 GHz
GPU	NVIDIA GTX 960M 4GB GDDR5 128-Bit
Memory	8GB DDR3

Table 2.2 Computer of student 2

Donanım	Model
CPU	Intel® Core(TM) i-7 7700HQ
GPU	Nvidia Geforce Gtx 1050ti 4GB 128 bit
Memory	16GB

2.3 Economic Feasibility

Pycharm IDE, Google Colab and python libraries are offered free of charge to users. The computers used in the project are personal computers.Therefore, the project is costless.

2.4 Legal Feasibility

The Dataset, programming language and libraries used in the project are open source. The project is legally eligible.

2.5 Time Feasibility

Time Schedule with Gantt Diagram.

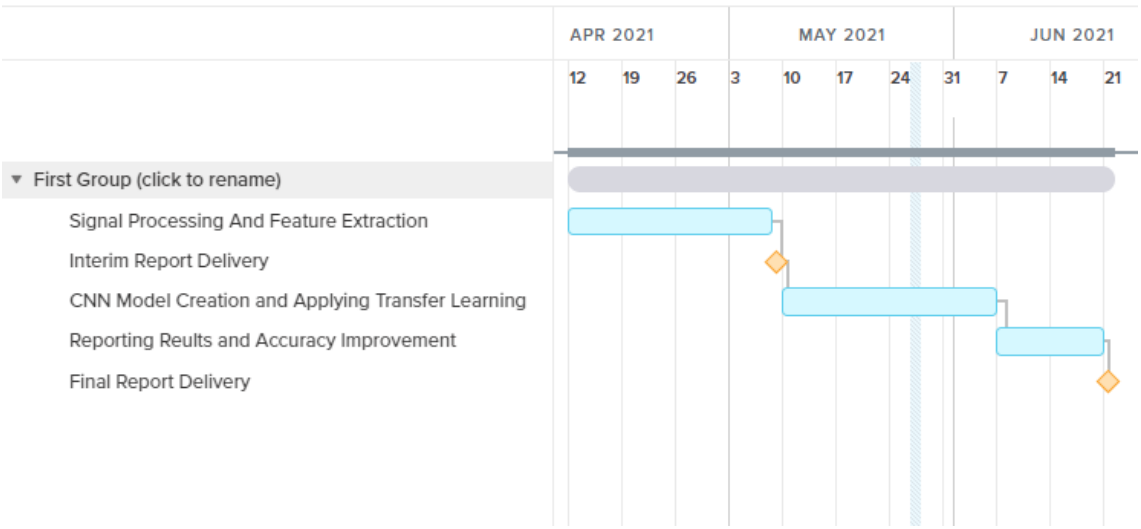

Figure 2.1 Gantt Diagram

3
System Analysis

This chapter consist of two main parts. First; Audio Analysis and feature Extraction. Second; deep learning with convolutional Neural Networks and Transfer Learning.

Audio Analysis and Feature Extraction includes Spectrogram, Short Time Fourier Transform, Mel Spectrogram, Bark Scale, Linear Prediction Coefficients.

3.1 Digital Audio Signals

First thing need to be understood is the Digital Audio Signal. Digital Audio Signal is samples of air pressure value over time. Value of the pressure is called as Amplitude of audio[5].

3.2 Fourier Transform

Periodic Audio Signals can be represented as sum of sinusoidal signals with different frequencies. This form of representation is called Fourier Transform of the signal[6]. Fourier Transform includes Amplitude values of sinusoidal signals besides frequencies. In other words time domain signal is transformed into frequency domain signal. The result is called as Spectrum.

Fourier Transform Formula:

$$X(w) = \int x(t) e^{jwt} dt \tag{3.1}$$

3.3 Short Time Fourier Transform and Spectrogram

Last section mentions about transforming a periodic signal into frequency domain form. But in most cases the signal changing over time so the signal is not periodic.

In this situation it is necessary to use another technique. It is called Short Time Fourier Transform. In this technique signal is being splitted into time windows, so each window includes a part of signal which is similar to a periodic signal. After that, Fourier Transform is applied for each window and amplitude value is converted into Decibels. After these operations the result is called as Spectrogram[6].

In this study, the data is doubled by applying data augmentation to the spectrograms. In other words, 10000 spectrograms are obtained from 10 classes in total. Data augmentation is done by applying time and frequency masking and time warping to the original spectrogram.The sample spectrogram can be seen in Figure 3.1. Data augmentation applied spectrogram can be seen in Figure 3.2 .

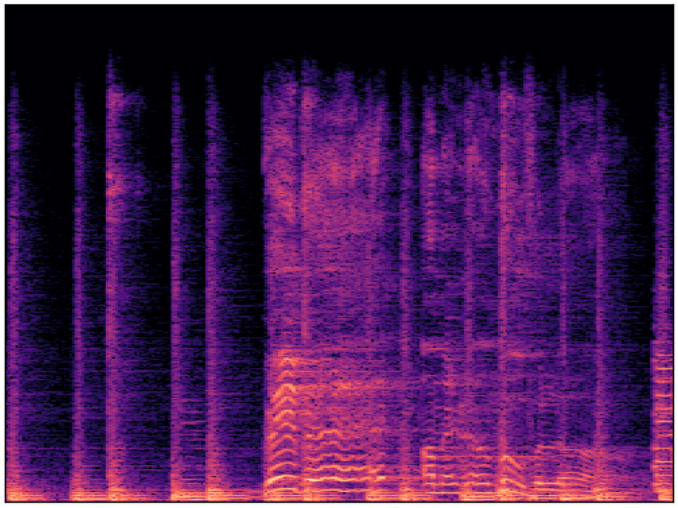

Figure 3.1 Spectrogram

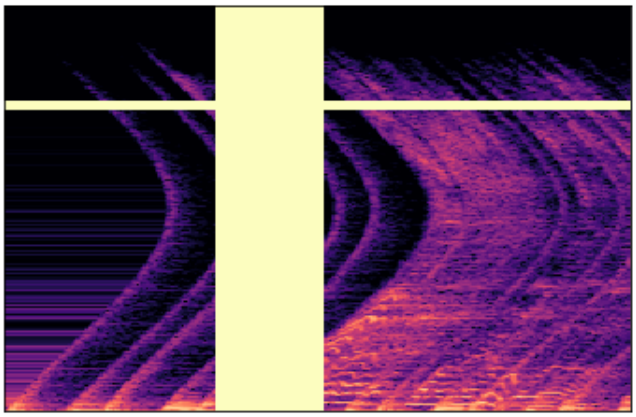

Figure 3.2 Augmented Spectrogram

3.4 Mel Spectrogram

In previous section, while creating spectrogram image of an audio neither Mel Scale nor logarithmic Scale is applied to frequency values. As it can be seen there is a huge space in high frequencies.

These spaces are not desired. In our case, for music genre classification, spectrograms are represented in a small area. So music genre characteristics are hard to be detected for deep learning model.

To get rid of these spaces there are some techniques which can be applied to spectrogram. In the mel spectrogram, the data is doubled by applying data augmentation as in the spectrogram.

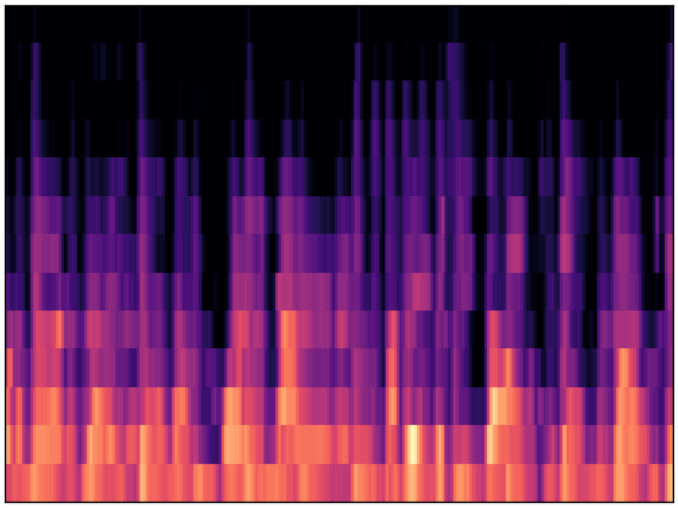

Figure 3.3 Mel Spectrogram

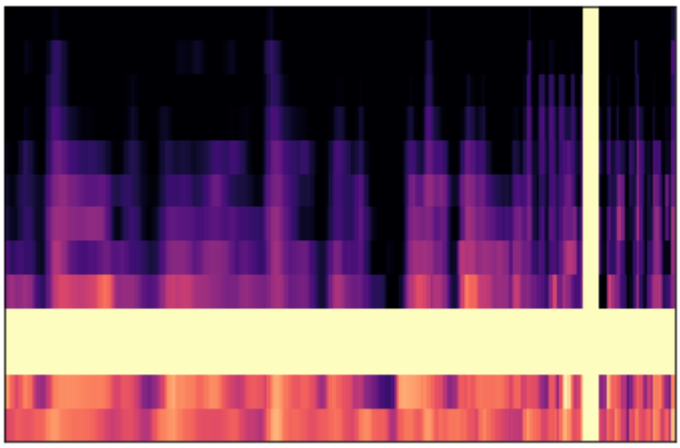

Figure 3.4 Augmented Mel Spectrogram

One of them is Log scale of frequency. As the name says this technique converts frequencies into logarithm of these frequencies. Another technique, which we will be using during this project is Mel Scale[7].

Mel Scale is also a type of logarithmic conversion with different formula. But this formula has a specific meaning.

Humans do not perceive frequencies on a linear scale. While we are good at detecting difference in frequencies on lower level, we cant tell the difference in high frequencies.

A human probably will not be able to tell the difference between 3000 and 3100 hz while he or she will be able to tell difference between 100 and 200 hz. This is what Mel scale is about. Mel scale splits frequency into parts that differentiates equally.

Mel Scale Formula:

$$M = 1127 * log(1 + F/700) \tag{3.2}$$

So, conversion is made by given formula to Spectrogram. As result Spectrogram images can be represented without spaces in high frequencies.

3.5 Mel Spectrogram - Delta MFCC and Delta-Delta MFCC

Another feature to be used in this study is the 1st and 2nd derivatives of the mel spectrogram.These derivatives also known as delta MFCC and delta-delta MFCC. While deriving derivatives here, MFCC values are obtained by applying the discrete cosine conversion process to the mel spectrogram. Then, when the MFCC values obtained for each frame in the audio signal are subtracted from the MFCC values obtained in the previous frame, the 1st derivative is obtained.

While obtaining the 2nd derivative, it is obtained by taking the difference of the 1st derivative values in the same way. So this time, instead of MFCC values, 1st derivative values are subtracted in frames. Figure 3.5 shows a sample of the 1st derivative of the mel spectrogram. Figure 3.6 shows a sample of the 2nd derivative of the mel spectrogram.

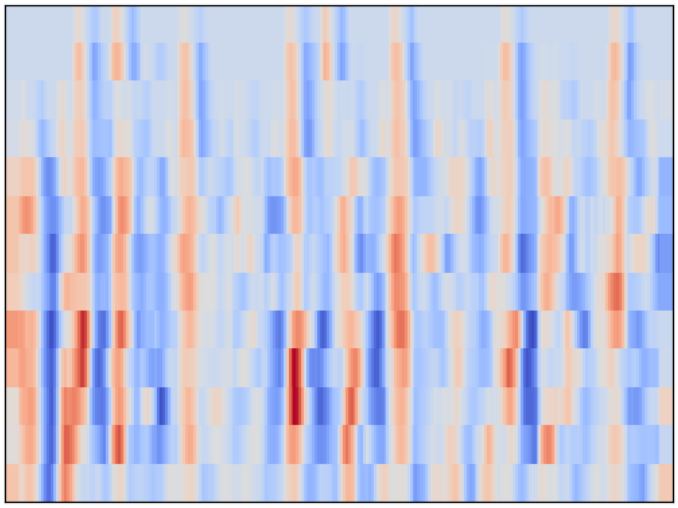

Figure 3.5 Mel Spektrogram Delta MFCC

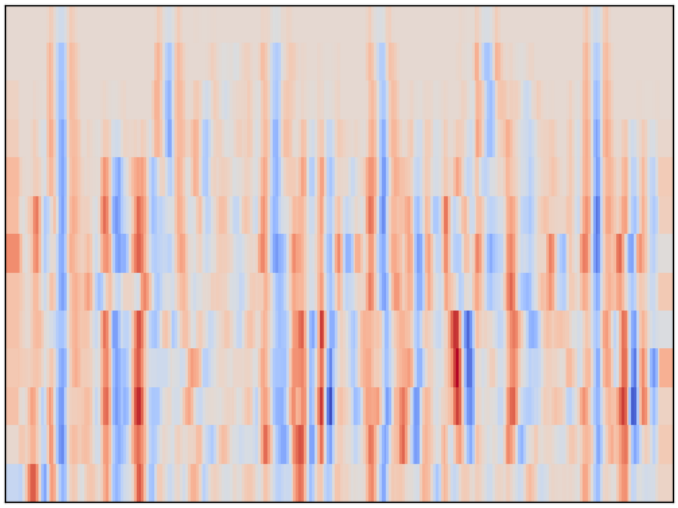

Figure 3.6 Mel Spectrogram Delta-Delta MFCC

3.6 Bark Spectrogram

Bark Scale results are similar with Mel Scale. It also converts frequency into logarithm like domain. Bark Scale is a frequency scale on which equal distances correspond with perceptually equal distances. The scale ranges from 1 to 24 and corresponds to the first 24 critical bands of hearing[7].

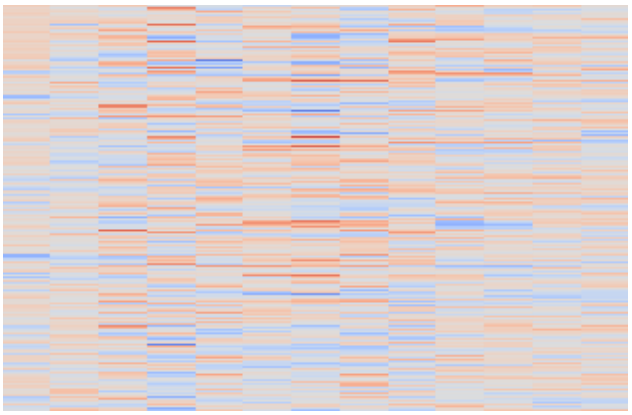

Figure 3.7 Bark Spectrogram

3.7 Linear Prediction Coefficients (LPC)

Linear prediction is a technique where future values are estimated from previous samples. Outputs also will be used in deep learning model for music genre classification[8].

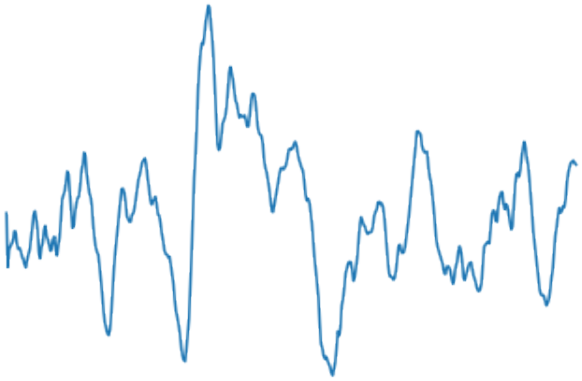

Figure 3.8 Linear Prediction Coefficients

3.8 Convolutional Neural Networks

Many problems occur nowadays can be solved with Convolutional Neural Networks which is really popular now.

In this chapter we will examine the Convolutional Neural Networks with layers. In deep learning, a Convolutional Neural Networks is a subclass of deep neural network which is used mostly in visual classification problems. For instance, detecting if a picture belongs to a cat or a dog.

In this project it is aimed to be able to make music genre classification. On this purpose, we will be creating a deep learning model which will be able to do this job.

CNN consist of layers. These layers are connected to each other with links which has weight values. These weight values are calculated with feedback mechanism. Calculation of weights is made by training with labeled samples[9].

In this way, an input to the model activates some neurons in a chain sequence in the various layers. Since similar inputs consist of similar sub-particles, Similar inputs activate same neurons in layers. The activation in the last layer determines which class the entered picture belongs to.

The training of the CNN model is achieved by giving the picture inputs which are labeled. In this way, the weight values connecting the layers are adjusted.

To summarize, the training of a CNN is all about finding the correct values in each of the filters, so when passed through multiple layers, an input image activates certain neurons of the last layer to predict the correct class.

3.9 Convolutional Neural Network Layers

Let's take a look at the structure of Convolutional Neural Networks to better understand CNN models. This section lists the layers of CNN to be examined.

1. **Convolution layer**
2. **Pooling Layer**
3. **Fully-Connected Layer**

1. **Convolution layer**

   The convolution layer is the core of the Convolutional Neural Network model and is responsible for detecting the features of the picture. In this layer, convolution filters are applied to the picture with certain size frames.

   [9].

   Convolutional Neural Networks use a large number of filters to detect multiple properties. The closer to the input layers, the smaller filter sizes and the smaller sized features are detected, while the closer to the output layers, the larger the filter sizes. There are different convolution layers to detect features in various parts of the CNN network[9].

   There are Non-Linearity layers, also called Activation layers, after the convolution layers. In these layers, activation functions are used that determine whether the weight values of the neurons from the previous convolution layers are activated or not.

   To summarize, the activation layer deals with which neurons in the next layer will activate and which will not activate the weight outputs resulting from the various layers. In other words, it determines which properties will be accepted as a result of convolution by using weights and activation functions.

2. **Pooling Layer**

   The pooling layer is responsible for simplifying and reducing the size of the content given as input to the network. As a result of the operations performed, the computational power required to process the data is reduced.[9].

The task of the Flattening layer, which is located between the convolution and pooling layers and the last layer called Fully-Connected, is to transform the outputs of the previous layer, which is in the matrix structure, into arrays. In this way, it can perform the last layer operations running on arrays[9].

3. **Fully-Connected Layer**

   The Classification layer, which is the last layer of the convolutional neural network, is where the class of the input is determined.

   This Classification layer, which takes the properties that were active at the end of the previous layers as input, is responsible for mapping these values with the relevant class outputs. It is important to create values in this layer during model training[9].

3.10 Transfer Learning

In this section, the Transfer Learning method will be mentioned. This method can be summarized as using pre-trained CNN models to solve current classification problem.

In our project, Transfer learning method is used with machine learning algorithms after feature extraction from pre-trained CNN model.

4
System Design

In this section code parts of the sections mentioned in System Analysis chapter and file structure of data set will be shown.

4.1 Data Set

Figure 4.1 Data Set

GTZAN includes 10 musical genres and it has 100 samples for each genre. These genres are: rock, classical, country, disco, reggae, jazz, metal, pop, hiphop, blues. Each sample is 30 seconds long. It is aimed to increase the performance of the classification models by increasing the number of data by splitting files into 6 seconds long parts.

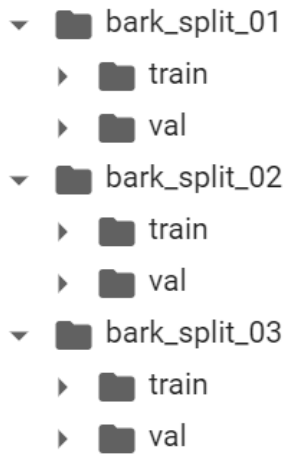

Figure 4.2 Train and Validation

After extracting Spectrogram, Mel Spectrogram, Bark Spectrogram and LPC images of audio files, each picture data set will be splitted to three different Train and Validation folders. So accuracy will be calculated multiple times.

4.2 CNN Model

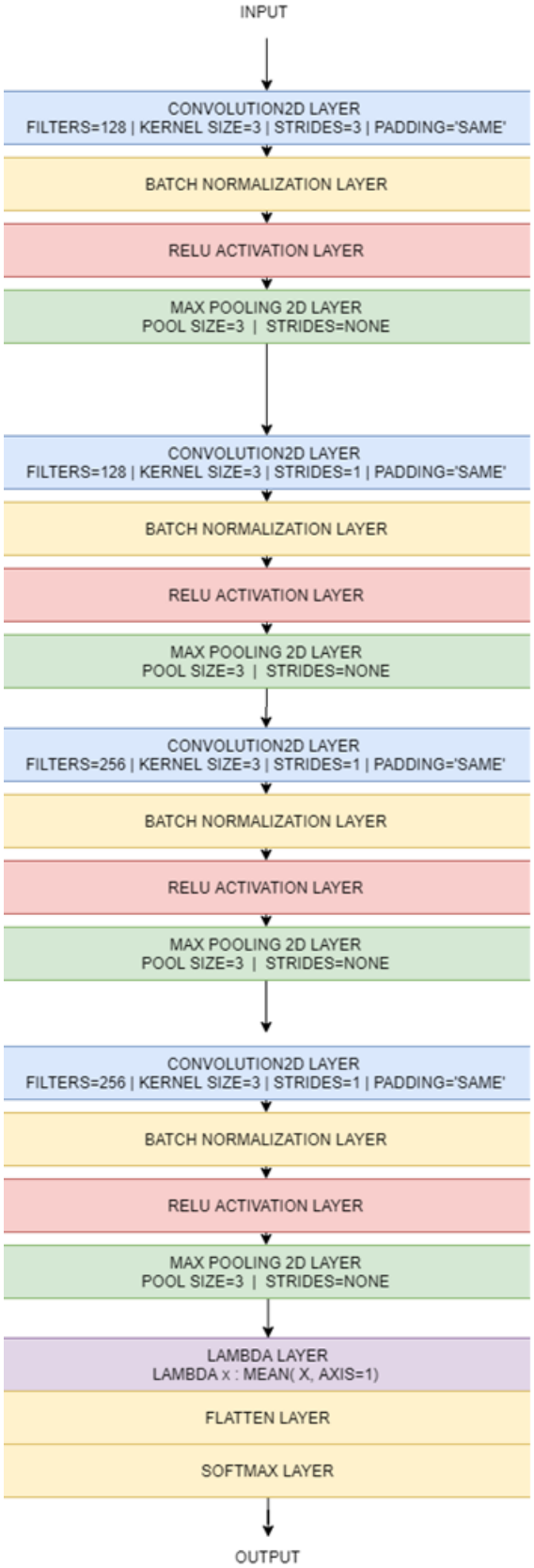

Figure 4.3 CNN model

Table 4.1 Model Parameters

	Model
Input Size	360x360x3
Number of Convolution Layers	4
Number of Pooling Layers	4
Total Number of Filters	768
Padding	'Same'
Activation	ReLu
Classifier	Softmax

Table 4.2 Model Hyperparameters

	Model
Number of Epochs	20
Learning Rate	0.001
Optimizer	ADAM
Mini-Batch Size	64

5
Experimental Results

5.1 Spectrogram

Table 5.1 Spectrogram Results

Split Folder	Accuracy	Recall	Precision
Folder 1	0.5365	0.5229	0.5603
Folder 2	0.5500	0.5293	0.5753
Folder 3	0.7135	0.7031	0.7313

5.2 Mel Spectrogram

Table 5.2 Mel Spectrogram Results

Split Folder	Accuracy	Recall	Precision
Folder 1	0.5656	0.54	0.5809
Folder 2	0.7350	0.6969	0.7467
Folder 3	0.5552	0.5406	0.5871

Table 5.3 Mel Spectrogram D1 Results

Split Folder	Accuracy	Recall	Precision
Folder 1	0.5268	0.5045	0.5446
Folder 2	0.5737	0.5469	0.5932
Folder 3	0.5826	0.58603	0.6092

Table 5.4 Mel Spectrogram D2 Results

Split Folder	Accuracy	Recall	Precision
Folder 1	0.5134	0.4911	0.5473
Folder 2	0.5000	0.4777	0.5258
Folder 3	0.5424	0.5357	0.5660

5.3 Bark Spectrogram

Table 5.5 Bark Spectrogram Results

Split Folder	Accuracy	Recall	Precision
Folder 1	0.4545	0.4040	0.5479
Folder 2	0.4747	0.4343	5375
Folder 3	0.4444	0.3737	0.5211

5.4 Linear Prediction Coefficients

Table 5.6 LPC Results

Split Folder	Accuracy	Recall	Precision
Folder 1	0.3348	0.2889	0.4040
Folder 2	0.3152	0.2889	0.3545
Folder 3	0.3645	0.2631	0.4200

Table 5.7 Total Accuracy Results

Data	Folder 1	Folder 2	Folder 3
Spectrogram	0.5365	0.5500	0.7135
Mel Spectrogram	0.5656	0.7350	0.5552
Mel Spectrogram D1	0.5268	0.5737	0.5826
Mel Spectrogram D2	0.5134	0.5000	0.5826
Bark Spectrogram	0.4545	0.4747	0.4444
LPC	0.3348	0.3152	0.3645

5.5 Transfer Learning

Table 5.8 Spectrogram Results

ML Algorithm	Split Folder 1	Split Folder 2	Split Folder 3
Multi-layer Perceptron	0.84	0.84	0.82
Logistic Regression	0.86	0.84	0.84
Random Forest	0.85	0.83	0.83
Linear Discriminant	0.86	0.83	0.82
K-Nearest Neighbors	0.89	0.89	0.87
Support Vector	0.86	0.85	0.85
Gaussian Naive Bayes	0.75	0.67	0.77
Gradient Boosting	0.82	0.85	0.82
Ada Boost	0.41	0.38	0.26
Linear SVC	0.86	0.86	0.83

Table 5.9 Mel Spectrogram Results

ML Algorithm	Split Folder 1	Split Folder 2	Split Folder 3
Multi-layer Perceptron	0.75	0.71	0.75
Logistic Regression	0.75	0.75	0.75
Random Forest	0.75	0.75	0.72
Linear Discriminant	0.76	0.74	0.75
K-Nearest Neighbors	0.79	0.79	0.78
Support Vector	0.76	0.76	0.75
Gaussian Naive Bayes	0.69	0.70	0.69
Gradient Boosting	0.74	0.73	0.74
Ada Boost	0.64	0.60	0.58
Linear SVC	0.75	0.74	0.74

Table 5.10 Mel Spectrogram D1 Results

ML Algorithm	Split Folder 1	Split Folder 2	Split Folder 3
Multi-layer Perceptron	0.59	0.63	0.62
Logistic Regression	0.61	0.67	0.68
Random Forest	0.61	0.64	0.66
Linear Discriminant	0.61	0.65	0.67
K-Nearest Neighbors	0.63	0.68	0.71
Support Vector	0.62	0.67	0.70
Gaussian Naive Bayes	0.60	0.62	0.67
Gradient Boosting	0.60	0.64	0.63
Ada Boost	0.49	0.50	0.43
Linear SVC	0.61	0.67	0.70

Table 5.11 Mel Spectrogram D2 Results

ML Algorithm	Split Folder 1	Split Folder 2	Split Folder 3
Multi-layer Perceptron	0.61	0.62	0.61
Logistic Regression	0.64	0.64	0.65
Random Forest	0.63	0.63	0.59
Linear Discriminant	0.63	0.62	0.62
K-Nearest Neighbors	0.66	0.67	0.65
Support Vector	0.64	0.66	0.64
Gaussian Naive Bayes	0.59	0.59	0.63
Gradient Boosting	0.59	0.63	0.61
Ada Boost	0.42	0.46	0.37
Linear SVC	0.64	0.64	0.64

Table 5.12 Bark Spectrogram Results

ML Algorithm	Split Folder 1	Split Folder 2	Split Folder 3
Multi-layer Perceptron	0.49	0.48	0.51
Logistic Regression	0.48	0.40	0.55
Random Forest	0.52	0.50	0.34
Linear Discriminant	0.53	0.41	0.47
K-Nearest Neighbors	0.48	0.44	0.54
Support Vector	0.50	0.43	0.48
Gaussian Naive Bayes	0.38	0.49	0.35
Gradient Boosting	0.44	0.46	0.47
Ada Boost	0.39	0.39	0.43
Linear SVC	0.48	0.45	0.54

Table 5.13 LPC Results

ML Algorithm	Split Folder 1	Split Folder 2	Split Folder 3
Multi-layer Perceptron	0.26	0.37	0.36
Logistic Regression	0.36	0.36	0.35
Random Forest	0.36	0.33	0.35
Linear Discriminant	0.33	0.32	0.34
K-Nearest Neighbors	0.37	0.36	0.37
Support Vector	0.31	0.29	0.34
Gaussian Naive Bayes	0.34	0.29	0.28
Gradient Boosting	0.25	0.34	0.34
Ada Boost	0.26	0.23	0.26
Linear SVC	0.36	0.37	0.38

6
Conclusion

Spectrogram accuracy results with CNN is 0.59 as avarage of three folders while 0.63 in Mel Spectrogram, 0.55 in Mel Spectrogram Delta, 0.51 in Mel Spectrogram Delta Delta, 0.45 in Bark Spectrogram and 0.34 in LPC.

When we look into differences between train and validation split folders, we can mention about 15 percent difference in some results. So we can say train and validation split can affact accuracy results.

As it can be seen highest accuracy results is taken with Mel Spectrogram. We can mention about good effects of data augmentation and fact that Mel Scale is the best technique for feature extraction.

Lowest accuracy results are in LPC. We comment this with fact that LPC samples only picture small parts of audio file, so it can miss significant information about classification problem.

About machine learning; it has better accuracy results than CNN in avarage.

Best accuracy result change between CNN and ML Algorithms is in Spectrogram Results. It changed 0.59 accuracy avarage to 0.81 accuracy avarage.

References

[1] M. Haggblade, Y. Hong, and K. Kao, "Music genre classification," *Department of Computer Science, Stanford University*, vol. 131, p. 132, 2011.

[2] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 282–289.

[3] C. Liu, L. Feng, G. Liu, H. Wang, and S. Liu, "Bottom-up broadcast neural network for music genre classification," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 7313–7331, 2021.

[4] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, IGI global, 2010, pp. 242–264.

[5] A. Lerch, *An introduction to audio content analysis: Applications in signal processing and music informatics*. Wiley-IEEE Press, 2012.

[6] H. J. Nussbaumer, "The fast fourier transform," in *Fast Fourier Transform and Convolution Algorithms*, Springer, 1981, pp. 80–111.

[7] A. V. Oppenheim, "Speech spectrograms using the fast fourier transform," *IEEE spectrum*, vol. 7, no. 8, pp. 57–62, 1970.

[8] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[9] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

Curriculum Vitae

FIRST MEMBER

Name-Surname: Hüsrev YUMUŞAK
Birthdate and Place of Birth: 24.01.1999, Sakarya
E-mail: hyumusal@gmail.com
Phone: +90 507 127 68 39
Practical Training: ANKAGEO, TURKSAT

SECOND MEMBER

Name-Surname: Abdurrahman Ebrar YÜCEL
Birthdate and Place of Birth: 27.12.1997, Giresun
E-mail: a.ebrartucell@gmail.com
Phone: +90 538 616 95 16
Practical Training:

Project System Informations

System and Software: Windows OS, Python
Required RAM: 2GB
Required Disk: 256MB