ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Serhat HIZLISOY

MUSIC EMOTION RECOGNITION USING CONVOLUTIONAL LONG SHORT TERM MEMORY DEEP NEURAL NETWORKS

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2020

ABSTRACT

PhD THESIS

MUSIC EMOTION RECOGNITION USING CONVOLUTIONAL LONG SHORT TERM MEMORY DEEP NEURAL NETWORKS

Serhat HIZLISOY

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Assoc. Prof. Dr. Zekeriya TÜFEKCİ
Year: 2020, Pages: 95
Juries : Assoc. Prof. Dr. Zekeriya TÜFEKCİ
: Assoc. Prof. Dr. Sami ARICA
: Assoc. Prof. Dr. Umut ORHAN
: Prof. Dr. Cebrail ÇİFLİKLİ
: Asst. Prof. Dr. Esra SARAÇ EŞSİZ

In this thesis, we propose an approach for Turkish music emotion recognition based on convolutional long-short term memory deep neural network (CLDNN) architecture. For this purpose, a new Turkish emotional music database composed of 124 Turkish traditional music excerpts with a duration of 30 seconds each is constructed. We used novel features obtained by feeding convolutional neural network (CNN) layers with log-mel filterbank energies and mel frequency cepstral coefficients (MFCC) in addition to standard acoustic features. Classification results show that the best performance is obtained when the new feature set is combined with the standard features using the LSTM + DNN (LDNN) classifier. The overall accuracy of %99.19 is obtained using the proposed system with 10-fold cross-validation. When new features are added to the standard features, 6.45 and 5.65 points improvements are achieved for native listeners and experts, respectively. Additionally, the results also show that the LDNN classifier yields 1.61, 1.61, 2.42 and 3.23 points improvements for native listeners and 5.65, 0.81, 4.84, and 6.45 points improvements for experts in music emotion recognition accuracies compared to that of K nearest neighbors (k-NN), Sequential Minimal Optimization (SMO), Naïve Bayes and Random Forest (RF) classifiers, respectively.

Keywords: Music Emotion Recognition, Convolutional Long Short Term Memory Deep Neural Networks, Turkish Emotional Music Database

ÖZ

DOKTORA TEZİ

EVRİŞİMLİ UZUN KISA SÜRELİ BELLEK DERİN SİNİR AĞLARINI KULLANARAK MÜZİKTEN DUYGU TANIMA

Serhat HIZLISOY

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Doç. Dr. Zekeriya TÜFEKCİ
Yıl: 2020, Sayfa: 95
Jüri : Doç. Dr. Zekeriya TÜFEKCİ
: Doç. Dr. Sami ARICA
: Doç. Dr. Umut ORHAN
: Prof. Dr. Cebrail ÇİFLİKLİ
: Dr. Öğr. Üyesi Esra SARAÇ EŞSİZ

Bu tezde, Türk müziği duygu tanıma için evrişimli uzun-kısa süreli bellek derin sinir ağı (CLDNN) mimarisine dayalı bir yaklaşım öneriyoruz. Bu amaçla, her biri 30 saniyelik 124 Türk geleneksel müziği alıntısından oluşan yeni bir Türkçe duygusal müzik veritabanı oluşturuldu. Standart akustik özelliklere ek olarak, evrişimli sinir ağları (CNN) katmanlarını log-mel filtre bankası enerjileri ve mel frekans sepstral katsayıları ile besleyerek elde edilen yeni özellikleri kullandık. Sınıflandırma sonuçları, en iyi performansın, yeni özellik setinin standart özelliklerle birleştirildiğinde ve LSTM + DNN sınıflandırıcı kullanılarak elde edildiğini göstermektedir. %99.19 genel doğruluk, 10 kat çapraz doğrulama ile önerilen sistem kullanılarak elde edilir. Standart özelliklere yeni öznitelikler eklendiğinde doğal dinleyiciler ve uzmanlar için sırasıyla 6.45 ve 5.65 puan iyileşme elde edilir. Ayrıca, LDNN sınıflandırıcısının, K en yakın komşu (k-NN), Sıralı Minimal Optimizasyon (SMO), Naïve Bayes ve Rastgele Ormanlar (RF) sınıflandırıcıları ile karşılaştırıldığında müzikten duygu tanıma doğruluk oranı için sırasıyla doğal dinleyiciler için, 1.61, 1.61, 2.42 ve 3.23, uzmanlar için 5.65, 0.81, 4.84 ve 6.45 puan iyileştirmeler sağladığını sonuçlar göstermektedir.

Anahtar kelimeler: Müzik Duygu Tanıma, Evrişimli Kısa Uzun Süreli Bellekli Derin Sinir Ağları, Türkçe Duygusal Müzik Veritabanı

EXTENDED ABSTRACT

In this thesis, general problems have been identified for the recognition of emotion from music, and it has been aimed to develop approaches to overcome these problems and increase the success of classification. For this purpose, in order to obtain a quality music emotion recognition system, a new dataset consisting of Turkish music excerpts was constructed, a new method was proposed to obtain features through CNN, and finally a deep learning-based classification method was used that has not been applied for emotion recognition before.

Recognizing emotion from music is still quite a difficult task today. People can feel different emotions from the same music because the emotion perception of people is inherently subjective. The perceived emotion can vary based on age, gender, character, lyrics, or environmental factors. Therefore, these factors cause listeners to evaluate music differently.

Music emotion recognition systems are used for many kinds of research such as music therapy and treatment of emotional disorders. In addition, many music suggestion applications that service according to the mood of people such as Spotify have been developed recently. These applications automatically analyze, recognize and classify the emotional content of music using signal processing and machine learning techniques.

In order to create a music emotion recognition system, firstly, a labeled emotional music database is needed. Labeling quality plays an important role in the success of the methods used on these data. There are two approaches to label emotions in music, categorical and dimensional. In the categorical approach, emotions are characterized using discrete labels, while in the dimensional approach, emotions are represented by independent axes. The advantage of dimensional models is that the uncertainty is reduced compared to the categorical approach. In this study, music excerpts were annotated according to the 2D model firstly. Later, when these annotations were examined, it was seen that the

annotations were gathered in 3 different quadrants and then they were adapted to 3 classes.

Another important factor affecting the success of emotion recognition from music is the feature extraction process that enables us to obtain acoustic features that fully meet the perceived emotion from music. There are also tools that allow us to obtain all these standard features in bulk. In our study, 7369 standard features were obtained by using OpenSMILE, MIRtoolbox and JAudio tools. In addition to these features, new features have been obtained by using convolutional neural networks (CNNs). By applying 1-dimensional convolution (1D-Conv) to the Mel frequency cepstrum coefficients (MFCCs) and log-mel filterbank energies obtained from the raw music signal, 768 more features were obtained from each of them.

The feature selection process was performed to remove unnecessary features for emotion recognition and to increase classifier performance. For this process, a correlation-based feature selection algorithm (CFS) is used in our study, which sorts sub-sets of features that are not related to each other but are highly related to the class through heuristic functions. In the experiments, 10-fold cross validation was used for all classifiers.

Within the scope of this study, emotion recognition from music was carried out using different machine learning methods. By using fully connected layers with long short term memory together as a classifier, the convolutional short term memory deep neural networks architecture, which has never been used before in emotion recognition from music, is completed. With this architecture, it is desired to achieve the best accuracy result by using many hyperparameters and their combinations intuitively.

Sequential minimal optimization (SMO), K nearest neighbor (k-NN), Random forests and Naïve Bayes methods, which are frequently encountered and widely applied in many studies, were used to compare the success of the system.

When the results of the study were examined, it was observed that the most successful recognition process for all classifiers was obtained when the standard

features were combined with the features obtained by using the Mel frequency cepstrum coefficients and log-mel filterbank energies. In the recognition made using these features, it was seen that the most successful classifier was the method in which the proposed long-short-term memory and fully connected layers were used together, and a recognition success of % 99.19 was achieved.

Apart from our dataset, 2 more datasets were used to test the success of the study. LDNN has also achieved the best result in the recognition process using these two datasets, and achieved higher recognition success than other studies using these datasets before. Experimental results are presented with the help of tables.

As a result, the experimental results showed the quality of the database created, the effect of new features on the success and the classifier used showed more effective results than other methods.

GENİŞLETİLMİŞ ÖZET

Bu tezde, müzikten duygunun tanınması için genel problemler tespit edilmiş, bu problemlerin üstesinden gelmek ve sınıflandırma başarısını artırmak için yaklaşımlar geliştirilmek istenmiştir. Bu amaçla, iyi bir müzik duygu tanıma sistemi elde etmek için, Türkçe müziklerden oluşan yeni bir veriseti oluşturulmuş, CNN yoluyla öznitelik elde etmek için yeni bir yöntem önerilmiş ve son olarak da daha önce müzikten duygu tanımaya uygulanmamış derin öğrenme tabanlı bir sınıflandırma metodu uygulanmıştır.

Müzikten duygu tanıma yapılması, günümüzde hala oldukça zordur. Çünkü, insanların duygu algısı özünde öznel oldugu için insanlar aynı müzikten farklı duyguları hissedebilirler. Hissedilen duygu; yaş, cinsiyet, karakter, şarkı sözleri veya çevresel faktörlere göre değişebilir. Dolayısıyla bu tür etmenler dinleyicilerin müzikleri farklı değerlendirmesine neden olur.

Müzikten duygu tanıma sistemleri, müzik terapisi ve duygusal bozuklukların tedavisi gibi birçok araştırma için kullanılır. Ayrıca son zamanlarda Spotify gibi insanların duygu durumuna göre hizmet veren birçok müzik öneri uygulamasının geliştirildiğini görmekteyiz. Bu uygulamalar, sinyal işleme ve makine öğrenme tekniklerini kullanarak bir müziğin duygusal içeriğini otomatik olarak analiz eder, tanır ve sınıflandırır.

Müzikten duygu tanıma sistemi oluşturmak için ilk olarak etiketlenmiş bir duygusal müzik veritabanına ihtiyaç vardır. Etiketleme kalitesi, bu veriler üzerinde kullanılan yöntemlerin başarısında önemli rol oynamaktadır. Müzikteki duyguları etiketlemek için kategorik ve boyutsal olmak üzere iki yaklaşım vardır. Kategorik yaklaşımda, duygular ayrık etiketler kullanılarak karakterize edilirken, boyutsal yaklaşımda duygular bağımsız eksenlerle temsil edilirler. Boyutsal modellerin avantajı, kategorik yaklaşımla karşılaştırıldığında belirsizliğin azalmasıdır. Bu çalışmada, müzikler önce 2 boyutlu modele göre değerlendirilmiştir. Daha sonra

değerlendirmeler incelendiğinde bu değerlendirmelerin 3 farklı bölgede toplanmış olduğu görülmüş ve 3 sınıfa adapte edilmiştir.

Müzikten duygu tanımanın başarısını etkileyen bir diğer önemli faktörde, müzikten algılanan duyguyu tam olarak karşılayan akustik öznitelikleri elde etmemizi sağlayan öznitelik çıkarma işlemidir. Bütün bu standart öznitelikleri toplu olarak elde etmemizi saglayacak araçlar da mevcuttur. Çalışmamızda OpenSMILE, MIRtoolbox ve JAudio araçları kullanılarak 7369 standart öznitelik elde edilmiştir. Bu özniteliklere ek olarak evrişimli sinir ağları kullanılarak yeni öznitelikler elde edilmek istenmiştir. Ham müzik sinyalinden elde edilen mel frekansı kepstrum katsayıları ve logaritmik mel-filterbank enerjilerine 1 boyutlu konvolüsyon uygulanarak her birinden 768 öznitelik daha elde edilmiştir.

Duygu tanıma işlemi için gereksiz olan öznitelikleri çıkarmak ve sınıflandırıcı performansını artırmak amacıyla öznitelik seçim işlemi yapılmıştır. Bu işlem için de çalışmamızda, birbiriyle bağlantısı olmayan ama sınıfla ilişkisi yüksek öznitelik alt kümelerini sezgisel fonksiyon aracılığıyla sıralayan korelasyon tabanlı öznitelik seçimi algoritması kullanılmıştır. Bütün sınıflandırıcılar için çapraz doğrulama sayısı 10 olarak ayarlanmıştır.

Bu çalışma kapsamında farklı makine öğrenmesi yöntemleri kullanılarak müzikten duygu tanıma işlemi gerçekleştirilmiştir. Uzun kısa süreli bellek ile tam bağlantılı katman bir arada sınıflandırıcı olarak kullanılarak, müzikten duygu tanımada daha önce hiç kullanılmayan evrişimli kısa süreli bellekli derin sinir ağları mimarisi tamamlanmış olmaktadır. Bu mimaride çok sayıda hiperparametre ve kombinasyonları sezgisel olarak kullanılarak en iyi sonuca ulaşılmak istenmiştir.

Sistemin başarısını karşılaştırmak için sık sık karşılaşılan ve birçok çalışmada yaygın olarak uygulanmış Sıralı minimal optimizasyon, K en yakın komşu, Rastgele ormanlar ve Naïve Bayes metodları kullanılmıştır.

Çalışma sonuçları incelendiğinde bütün sınıflandırıcılar için en başarılı tanıma işleminin standart öznitelikler ile, mel frekansı kepstrum katsayıları ve

logaritmik mel filterbank enerjileri kullanılarak elde edilen öznitelikler birleştirildiğinde elde edildiği gözlenmiştir. Bu öznitelikler kullanılarak yapılan tanıma da ise en başarılı sınıflandırıcının önerilen uzun kısa süreli bellek ile tam bağlantılı katmanların bir arada kullanıldığı yöntem olduğu görülmüştür ve %99,19 tanıma başarısı elde edilmiştir. Bizim oluşturduğumuz veriseti dışında 2 veriseti daha çalışmanın başarısını test etmek adına kullanıldı. LDNN tanıma işleminde bu iki verisetinde de en iyi sonuçları elde etmiştir ve bu verisetlerinin daha önce kullanıldığı çalışmalardan daha yüksek tanıma başarısı elde edilmiştir. Deneysel karşılaştırma sonuçları tablolar yardımıyla sunulmuştur.

Sonuç olarak deneysel sonuçlar önerilen veritabanının kalitesini, elde edilen yeni özniteliklerin başarıya etkisini ve kullanılan sınıflandırıcının diğer metodlardan daha etkili sonuçlar ortaya koyduğunu göstermiştir.

ACKNOWLEDGEMENT

I would like to present my gratitude to my supervisors Assoc. Prof. Dr. Zekeriya TÜFEKCİ and Assoc. Prof. Dr. Serdar YILDIRIM for their endless support, patience, guidance, and motivation for the successful completion of this thesis.

I wish to express my special thanks to Assoc. Prof. Dr. Sami ARICA, Assoc. Prof. Dr. Umut ORHAN and and Asst. Prof. Dr. Buse Melis ÖZYILDIRIM for their corrections, suggestions and their valuable help. I would like to thank members of the PhD thesis jury Prof. Dr. Cebrail ÇİFLİKLİ and Asst. Prof. Dr. Esra SARAÇ EŞSİZ for their constructive corrections and suggestions.

Special thanks to my beloved wife Tuğba, my father and my mother for their limitless support and patience.

Finally, I would like to thank TUBITAK (The Scientific and Technological Research Council of Turkey) for providing financial support during my thesis.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

LIST OF ABBREVIATIONS

MER	: Music Emotion Recognition
MIREX	: Music Information Retrieval Evaluation Exchange
AMG	: All Music Guide
FMA	: Free Music Archive
DEAM	: The Database for Emotional Analysis of Music
MIR	: Music Information Retrieval
ZCR	: Zero Crossing Rate
LPC	: Linear Predictive Coding
MFCC	: Mel Frequency Cepstral Coefficient
Log-mel	: Log-mel Filterbank Energy
CNN	: Convolutional Neural Network
RF	: Random Forest
DNN	: Deep Neural Network
LSTM	: Long Short-Term Memory
SVM	: Support Vector Machine
K-NN	: K Nearest Neigbors
DNN	: Deep Neural Network
DCNN	: Deep Convolutional Neural Network
GMM	: Gaussian Mixture Model
CLDNN	: Convolutional Long Short-Term Memory Deep Neural Network
FC	: Fully Connected Layer
RNN	: Recurrent Neural Network
TEM	: Turkish Emotional Music Database
HANV	: High Arousal and Negative Valence
HAPV	: High Arousal and Positive Valence
LANV	: Low Arousal and Negative Valence

MLP	: Multi-Layer Perceptron
RELU	: Rectified Linear Unit
DTFT	: Discrete Time Fourier Transform
DCT	: Discrete Cosine Transform
RMS	: Root Mean Square
STFT	: Short Time Fourier Transform
LLD	: Low Level Descriptors
HLF	: High Level Features
CFS	: Correlation-Based Feature Selection Algorithm
SMOreg	: Sequential Minimal Optimization Regression
MAE	: Mean Absolute Error
GUI	: Graphical User Interface
ARFF	: Attribute Relationship File Format
CLI	: Command Line Interface
CSV	: Comma Seperated Values
API	: Application Programming Interface
CNTK	: Microsoft Cognitive Toolkit
GPU	: Graphics Processing Unit
CPU	: Central Processing Unit
IDE	: Integrated Development Environment
SGD	: Stochastic Gradient Descent
ADAM	: Adaptive Moment Estimation

1. INTRODUCTION

Music can define as meaningful vibrations that have a rhythmic structure by coming together sounds that are compatible with each other. Music is an important phenomenon both socially and individually, which has existed in our lives since the beginning of humanity (Panda, 2010). Music is loved by almost all people because of its impact on human psychology and being seen as an entertainment tool.

Music has many effects on people. Many studies conducted in different disciplines such as computer science, musicology, sociology and even psychology have aimed to reveal these effects by using different methods and measurement tools. Emotion comes first among these effects. Because music is also known as an art form in which emotions are expressed.

Emotions are the most important thing that separates man from other living things. These emotions are gathered under various headings such as happiness, sadness, anger, excitement and calm. These emotions, which distinguish one from the other, play a key role in terms of their existing function as well as their common or unique features. Identifying these emotions can be realized through emotion recognition systems.

Audio and visual elements are used as important data in emotion recognition systems. In addition, many physiological parameters can be used, such as heart rate, blood pressure, brain waves. Speech emotion recognition systems or music emotion recognition (MER) systems can be given as examples to emotional recognition systems from audio elements.

Listening to music can affect people's emotional state. The emotional states of people can also be guiding on the music they will listen to. Therefore, people may tend to listen energetic music when they are happy, or listen to slow music when they are sad. Many music suggestion applications such as Spotify have been developed recently, which serve this purpose. These applications automatically

analyze, recognize and classify the emotional content of a music using signal processing or machine learning techniques. It does this through music emotion recognition systems. Music emotion recognition systems are also used for many researches such as music therapy and treatment of emotional disorders.

Identification of the emotional category of music is quite challenging due to the following reasons. People can feel different emotions for the same music because emotion perception is completely subjective. Because it can change from person to person (age, gender, and character) or it can be affected from to the song lyrics or environmental factors. Also, there are several issues that need to be addressed such as quality dataset, emotion labeling of music excerpts, feature extraction, and choice of the classification methods.

In this thesis, such problems for the recognition of emotion in music have been detected and approaches have been developed to overcome these problems and increase the classification success of MER.

First, an annotated emotional music database is needed to create a music emotion recognition system. The quality of the labeling is an important factor in the success of the methods that are trained on these data. There are two approaches, i.e., categorical and dimensional, for labeling the emotions in music. In the categorical approach, emotions are characterized using discrete labels. One of the earliest researches in this area was done by Hevner (1936). A categorical model with 66 emotion tags clustered in 8 categories is presented. For classifying emotions, Y. Feng et al. (2003) choose sad, happy, fear, angry, which have been used in first study about the MER field. Paul Ekman suggested that there are 6 basic emotions: sadness, anger, surprise, happiness, disgust, and fear (Ekman, 2005). Plutchik proposed eight basic emotions: disgust, joy, sadness, fear, trust, anticipation, surprise, and anger (Plutchik, 1980), and arranged them in a circular diagram. In addition, the five clusters and respective subcategories were utilized in Music Information Retrieval Evaluation eXchange4 (MIREX) for audio mood classification (Hu et al., 2008). Categorical models are problematic because

emotion words may not have exact translations in different languages and there is nonexistence of consensus on category numbers. This problem does not exist in the case of dimensional models.

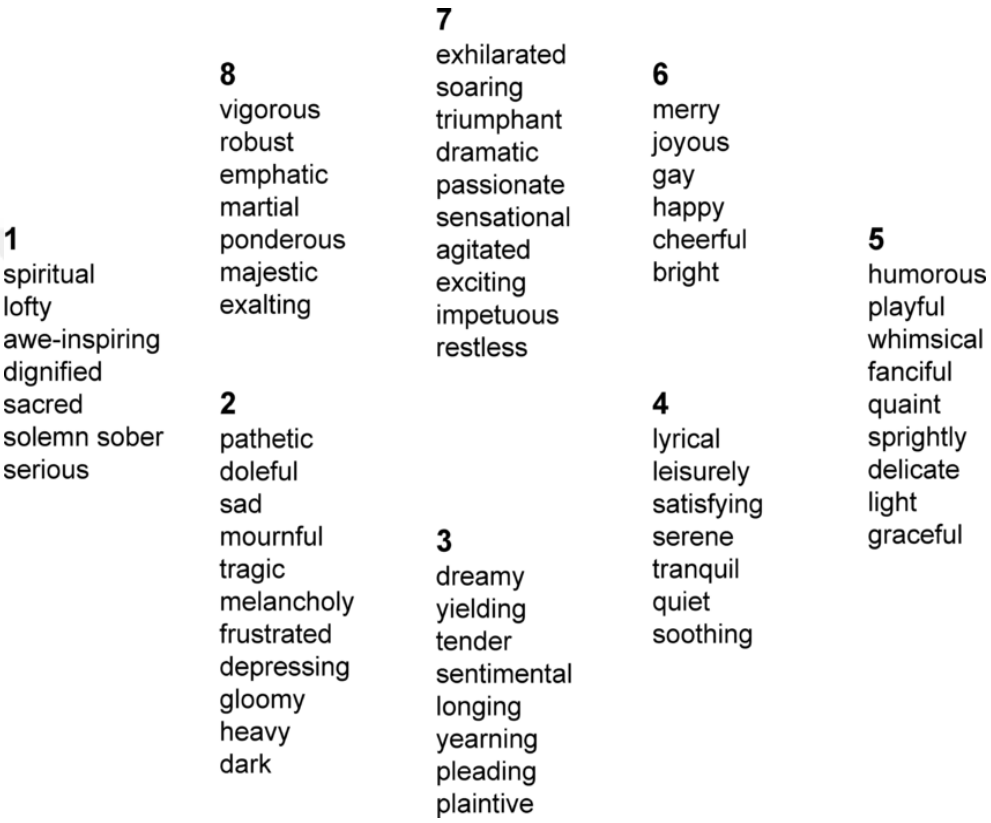

Figure 1.1. Hevner's model

In the second approach, emotions are represented several and independent axes in dimensional space. Usually two-dimensional model is used in studies. The third dimension, called dominance, is also used in some cases. The advantage of dimensional models is the reduced uncertainty when compared to the categorical models. Russell (1980) proposed a model that consists of two dimensions: valence and arousal. This model enables a reliable way to measure emotion for people into two independent dimensions. Also, well-known dimensional model for musical

definition is the Thayer's energy-stress model (Thayer, 1989). This model is adapted from Russell's circumplex model and consists of valence and arousal dimensions. These dimensions affect underlying stimuli that could influence mood responses (Lu et al., 2006; Mo and Niu 2017; Yang et al., 2008; Malheiro, 2018). The arousal is used to show calm to the excited axis, and the valence axis is utilized to show the notion of displeasure vs. pleasure. In this study, two-dimensional model is used to annotate music excerpts initially.

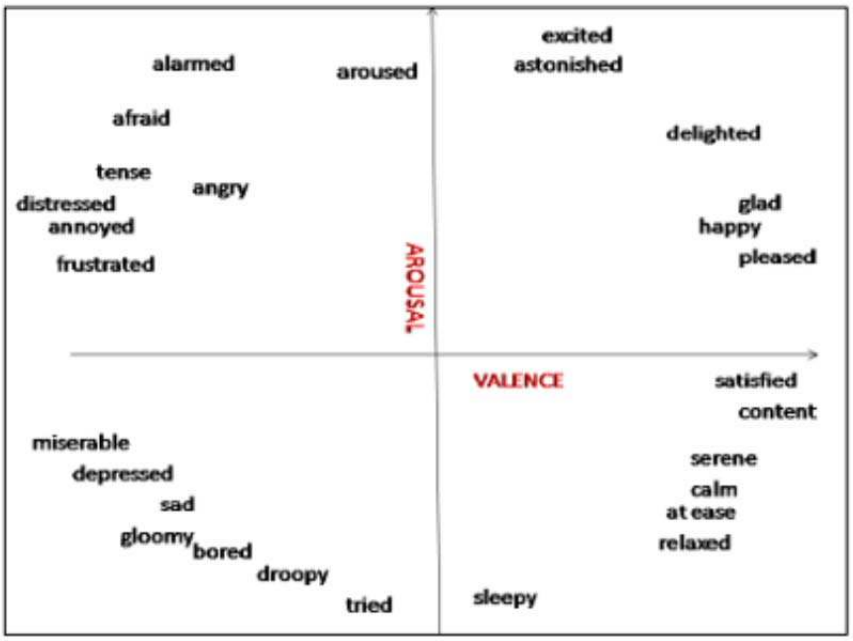

Figure 1.2. Russel's circumplex model

Datasets also play a crucial role in MER. Li and Ogihara (2003) also studied about it. One of the major problems with their work was the use of a single evaluator to classify music in the dataset because it was not sufficient. Panda et al. (2012) proposed a database of 903 audio clips labeled with 5 emotion clusters by the MIREX mood classification task. Y.C. Lin et al. (2011) use tags from AMG to create a database of 7922 music annotated with 183 emotional tag (Aljanaki,

2016). A database used by Y.H. Yang and H.H. Chen (2011) contains 1240 Chinese pop music annotated with valence and arousal using rankings scheme. Another large database consists of 744 music collected by M. Soleymani et al. (2013). In this database music excerpts were selected from the free music archive (FMA). AMG1608 contains 1608 30-second fragments of music in various music genres annotated by 665 subjects on valence and arousal (Chen et al., 2015). Google introduced AudioSet in 2017 that includes more than two million 10-s audio excerpts extracted from YouTube videos directly (Gemmeke et al., 2017). The audio excerpts are labeled using 527 categories (De Benito-Gorron et al., 2019). The database for emotional analysis of music (DEAM) consists of 1802 excerpts and full music annotated with valence and arousal (Aljanaki et al., 2017).

When the researches conducted to determine the emotion in music were examined, almost all of the researchers prepared the datasets by selecting the samples in western music. In our literature search, no study created for emotion recognition with Turkish music was found. Therefore, a Turkish emotional music database composed of 124 Turkish traditional music excerpts is constructed in this thesis.

MER is a subfield of music information retrieval (MIR) that deals with music classification and develops music similarity features. One of the important parts that affect the success of MER is the feature extraction that enables us to obtain acoustic features that fully meet the perceived emotion in music. A wide range of acoustic features have been explored by the researchers of MER recently. The features relevant to music and emotion were used such as timing, dynamics, articulation, pitch (Tzanetakis, 2002; de Cheveigne and Kawahara, 2002; Cabrera, 1999), melody, harmony (Harte et al., 2006; Lartillot and Toiviainen, 2007), tonality (McKay, 2009), and rhythm (Tzanetakis, 2002; Pampalk et al., 2002). Classical audio features also were used such as energy (Lartillot and Toiviainen, 2007), Zero Crossing Rate (ZCR) (Tzanetakis, 2002), MFCCs (Tzanetakis, 2002), log-melfilterbank energies (log-mels), Linear Predictive Coding (LPC)

(Tzanetakis, 2002), chromagram, and spectral shape features including centroid (Tzanetakis, 2002), spread, skewness, kurtosis, slope, roll-off (Tzanetakis, 2002; Pampalk et al., 2002), flux (Tzanetakis, 2002), contrast. There are also tools available to extract those features such as openSMILE (Eyben and Schuller, 2015), YAAFE (Mathieu et al., 2010), MIRToolbox (Lartillot and Toiviainen, 2007), jAudio (McKay, 2009), Psysound (Cabrera, 1999) and Marsyas (Tzanetakis and Cook, 1999).

In this study, some of these toolboxes like openSMILE, MIRToolbox and jAudio is employed to extract the aforementioned features. Further to the standard features widely used in music emotion recognition it is necessary find novel features that have the capability to represent the emotional characteristics. Deep learning maintains an alternative way to find suitable features for the recognition task. Compared to existing acoustic features, researchers have explored the effect of CNN to automatically learn affective data from the music signal via deep architecture that can provide a higher-level representation from annotated data. In the last few years, CNN (De Benito-Gorron et al., 2019; Sarkar et al., 2019; Maas et al., 2017; Santamaria-Granados et al., 2019; Simonyan and Zisserman, 2014) has demonstrated successful results in various research areas including image classification and recently speech recognition. However, a few studies were conducted to extract features using CNN (Liu et al., 2017; Liu et al., 2018). In previous studies, CNN was fed with the raw audio signal or spectrogram (De Benito-Gorron et al., 2019; Sarkar et al., 2019; Huang and Narayanan, 2017). In this paper, CNN was fed with MFCCs and log-mel filterbank energies extracted from music excerpts to extract new features.

Another important task of MER is choosing the classifier. When the studies conducted to determine the emotion in music are examined, it is observed that most of them use machine learning methods. Support vector machine (SVM) (Song et al., 2012, Rocha et al., 2013, Grekow, 2015), random forest (RF) (Zhang et al., 2016; Grekow, 2015), k-NN (Delbouys et al., 2018, Grekow, 2015), DNN

(De Benito-Gorron et al., 2019; Liu et al., 2019), LSTM (De Benito-Gorron et al., 2019; Han et al., 2009; Liu et al., 2019), deep convolutional neural network (DCNN) (Sarkar et al., 2019) and gaussian mixture model (GMM) (Sainath et al., 2015a) are used as a classifier for music emotion recognition by the researchers.

In this study, an approach based on CLDNN architectures that use MFFCs, log-mel filterbank energies as input is proposed for MER. The CLDNN architecture was previously applied to speech recognition and music detection (De Benito-Gorron et al., 2019; Huang and Narayanan, 2017; Sainath et al., 2015a; Sainath et al., 2015b). Our architecture consists of convolutional layers, LSTM layer and fully connected (FC) layers for music emotion recognition.

In this study, answers to the following sub-problems were sought in line with the goals and objectives. In summary, the research questions we pose are the following:

- Can other meaningful data be obtained with the features extracted from music signals?
- Can emotions in Turkish music be better interpreted with mathematical and statistical data compared to other languages music?
- Can emotions in Turkish music be better classified using deep networks instead of existing methods?
- Are the experts more successful than native listeners in labeling and classifying emotions in music?

This thesis is structured as follows. The details of the Turkish emotional music database are presented in Section 2. Section 3 provides an overview of the system including feature extraction, feature selection and classification. Section 4, it is described how the experiment was done. Section 5 presents the results and

discussions. Finally, Section 6 mentions possible future works and concludes the study.

2. NEW EMOTIONAL TURKISH MUSIC DATABASE

Creating a database is a time-consuming and labor-intensive process. Many databases in different languages have been created recently, but the lack of quality datasets forced researchers to create their own datasets. In addition, it was seen in the literature review that there was no Turkish music dataset among these studies publicly available. Thus, a new Turkish Emotional Music Database (TEM) is created in this study. It is thought that this dataset, which has been brought into the literature, will contribute to more collaboration in the field of engineering and music and it will increase success in the classification of emotional content.

There are many datasets from the past to the present. More and more data are reachable with the development of technology recently. Especially, online sources allow us to collect much bigger datasets. However, the quality of their annotations could be questionable because, the labels are assigned by online users (Last.fm, AMG), which in some cases may cause confusion and uncertainty.

In this section, detailed information will be given about our dataset that is created in line with all these inferences.

This database is composed of 124 Turkish traditional music excerpts with a duration of 30 seconds, representing the most salient part of the music, besides containing one single emotion. The reason why we chose the time 30 seconds is that there may be many emotional transitions during longer music. Thus, there may be contradictions between the features obtained and the emotions represented. Music excerpts were selected by experts and they belong to various genres mainly instrumental versions.

Generally, two approaches are used in MER for database creation. In the first approach, only music signals are available to the annotator, whereas in the second approach lyrics with metadata are also presented. In this study, the first approach is followed covering various genres and styles. Because we are interested in finding the connection between the information in the music signal with

subjective emotion that corresponds to the information. It was also thought that metadata or lyrics could change the emotion perceived by music.

The music excerpts were preprocessed and standardized before being presented to the annotator. First, our dataset is converted to standard wav format (mono channel, 44.100 Hz sampling rate and 16-bits quantization). Then, music excerpts are renamed to as 1.wav, 2.wav, ..., 124.wav, in order to hide singer and their title. Because, it might affect subject's assessment during annotation.

Obtaining a quality dataset, is a challenge for MER. Because music annotation is a time-consuming task and costs of annotations and music copyright pose problems. There are two ways of constructing a labeled dataset with emotion annotations: through social tag mining or data collection games. Even though different ways, such as social-tags and data collection games have been used, one of the most ideal approach is still manual labeling to create a quality dataset. Social tag mining allows for a large dataset collection; however, it lacks the homogeneity and control provided by experimental setting. In most cases, it is not possible to measure the level of agreement between multiple users on certain tags.

Most researchers believed that manually labelling enables better control regarding ambiguity. Therefore, a program that listeners could use to annotate music manually was needed. This program should be able to play all the music of our dataset to the annotator. A user interface program is used for the annotation called AnnoEmo (Yang et al., 2007), which is also seen in Figure 2.1. Through this program, annotators selected the values of arousal and valence for each music in the two-dimensional emotion plane. This plane is defined as a coordinate field where evaluation for arousal and valence can be made. The valence dimension represents a positive to negative axis whereas arousal represents excited to calm axis as mentioned before. During the annotation process, evaluators were allowed to listen to music excerpts many times, and they were able to change their annotations through this program.

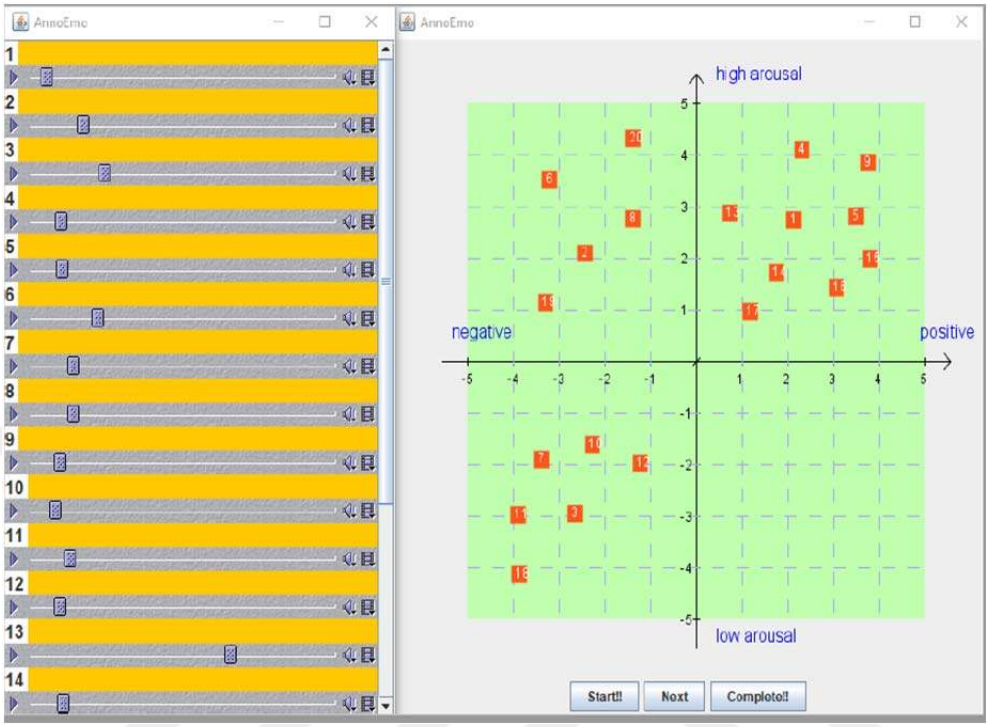

Figure 2.1. A snapshot of AnnoEmo

The following annotation methodologies were recommended to the annotators:

1. Annotators could use 11 possible values both for arousal and for valence [-5, 5].
2. Fine-tune the values. When tired, end the annotations and continue later. When in doubt, skip music and annotate another one.
3. Annotators were also advised not to research music on the Internet or elsewhere to further improve the quality of annotations.

The emotional tags in the music datasets prepared for such studies are usually taken from experts or determined by the experiments attended by a group of native listeners. In this study, the evaluations of both experts and native listeners

was used to compare their achievements and differences in the classification process. The music excerpts were evaluated by 2 very knowledgeable experts and 21 native listeners who could not only be music listeners but also use instruments.

Looking at the gender balance of the listeners, %22 was female and %78 was male. Each music excerpt was listened by all listeners. Firstly, the values were converted between -5 and 5 to -1 and 1 by normalizing.

The average of the annotations given by the experts as arousal and valence for each music are calculated. The same process was done for the native listeners. Using these evaluations separately in the classification stage, the results were calculated for experts and native listeners.

Many researchers from different disciplines have dedicated themselves to the music emotion modeling. There are two approaches for this issue: the dimensional models and the categorical models. Categorical models classify emotions using groups of words or single words to characterize them. Dimensional models consist of independent dimensions that maps different emotions. Well-known two-dimensional model for musical definition is the Russel's model. In some researches, the third dimension, called dominance, is used.

In this thesis, Russel's two-dimensional model was combined with a categorical model based on three basic emotions and adopted to our music emotion recognition system. As can be seen from Figure 2.2. that, music excerpts are distributed into three quadrants based on the distribution of the average of the annotator's ratings on valence-arousal dimensions. Therefore, it was transformed into these three classes instead of directly using valence and arousal values to train them. The upper right quadrant contains high arousal and positive valence (HAPV) emotions like happy, while the upper left quadrant contains high arousal and negative valence (HANV) emotions like anger, and the lower left quadrant contains low arousal and negative valence (LANV) emotions like sadness as shown in Figure 2.2. The distributions of music excerpts into three quadrants are 75, 11, and 38, respectively for native listeners and 76, 13 and 35 for experts.

Table 2.1. The distributions of the music excerpts in terms of classes.

	Listeners	
Class	**Native Listeners**	**Experts**
HAPV	75	76
HANV	11	13
LANV	38	35
Number of music	124	124

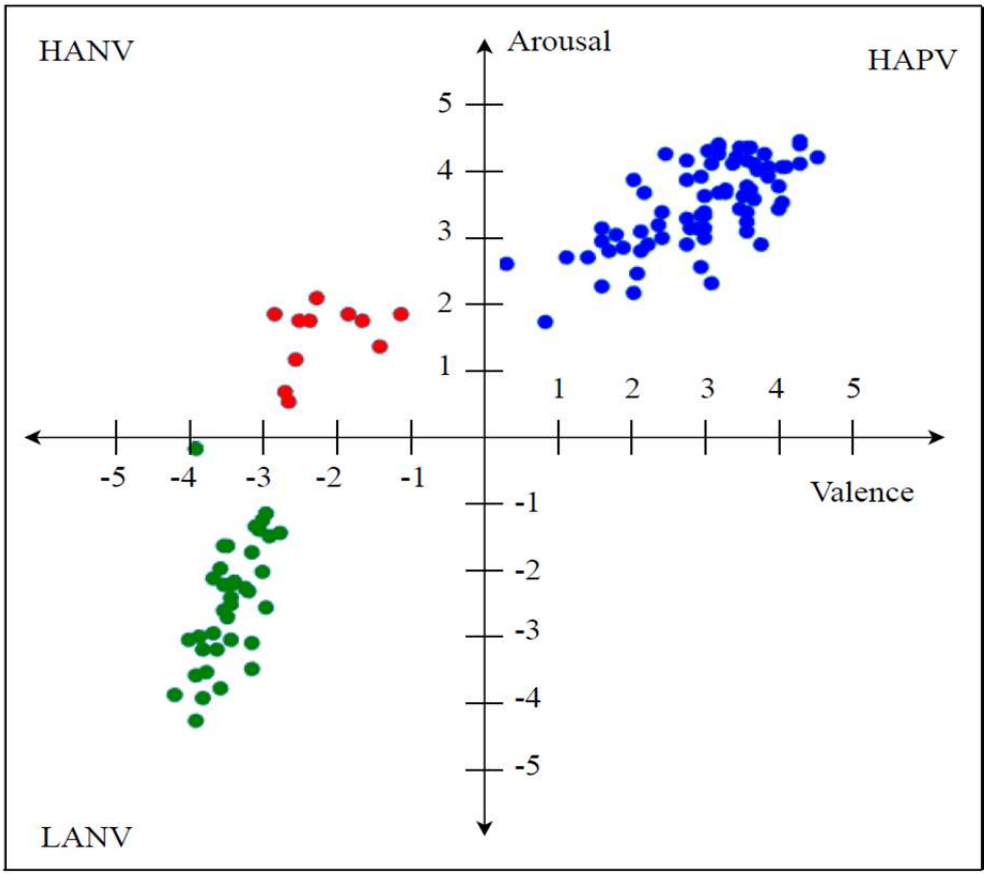

Figure 2.2. The distribution of the average of the annotator's ratings of music excerpts on valence-arousal dimensions.

The average of all annotator's ratings was then used as the ground truth of each music excerpt. Figure 2.2. shows the distribution of the average of the annotator's ratings of music signals on valence and arousal dimensions.

The standard deviation of the annotations was calculated to evaluate the consistency of the dataset for native listeners and experts and is defined as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i-\mu)^2}{N}}. \quad (2.1)$$

where $\{x_1, x_2, \ldots, x_N\}$ are the observed values of the listeners annotation and μ is the mean value of these observations, while the denominator N stands for the number of the listeners. The standard deviation σ (sigma) can be express as the square root of the variance of x, or square root of the average value of $(x_i - \mu)^2$.

Also, the distributions of the standard deviations are given in Table 2.2. For native listeners, the averages of standard deviations are 0.243 and 0.276 for arousal and valence, respectively. For experts, the averages of standard deviations are 0.062 and 0.70 for arousal and valence, respectively. Almost all music excerpts had a standard deviation under 0.3 for arousal and valence.

Table 2.2. The distributions of the standard deviations in terms of listeners.

	Native Listeners	**Expert**
Arousal	0.243	0.062
Valence	0.276	0.070

Lack of agreement between the annotator's leads to ineffective information and low discrimination ability of the methods. To evaluate the agreement among annotators in terms of categorical classes, Krippendorff's α (Krippendorff, 2007)

and Intraclass correlation coefficients (Koch, 1982) were used as inter-rater reliability measures. The results given in Table 2.3 show that the agreement between the annotators is quite high. While it was observed that the agreement between the annotators on arousal values was higher than the valence values. It can be inferred that people are more able to distinguish the arousal rather than valence.

If the results of inter-rater reliability are wanted to be evaluated, it can be said that the highest agreement result in terms of both arousal and valence is in the Intraclass Correlation Co. compared to the other. Likewise, if the listeners is compared in terms of agreement, it can be said that the experts are higher than the native listeners.

Table 2.3. Annotation agreement between annotators in terms of arousal and valence values.

	Arousal			**Valence**		
	Native	**Expert**	**All**	**Native**	**Expert**	**All**
Krippendorff's α	0.750	0.891	0.751	0.747	0.892	0.744
Intraclass Correlation Co.	0.808	0.942	0.810	0.806	0.929	0.800

For assessing the quality of the annotations in terms of classes, the Fleiss' kappa metric was used. The kappa, K is defined as,

$$K = \frac{P_a - P_c}{1 - P_c}, \tag{2.2}$$

where P_c is the proportion of times we would expect the n annotators to agree by chance and P_a is the proportion of times that the n annotators agree. The details of how P_a and P_c can be calculated are as in (Fleiss, 1971). If K values are interpreted according to the table presented by Landis and Koch (1977);

$K = 0$, There is no agreement among the native listeners and experts,

$K = 1$, There is full agreement among native listeners and experts.

Looking at the kappa value between native listeners and experts in our study, it is seen that this value is 0.906. From this result, it can be said that there is almost full agreement between native listeners and experts in terms of classes.

3. PROPOSED CLDNN ARCHITECTURE FOR MER

Deep Learning is a new area of machine learning that has been gaining popularity over the past few years. Deep learning mimics the abilities of the human brain such as observation, analysis, learning and decision making for complex problems. It performs operations such as feature extraction and classification using large amounts of data, with supervised or unsupervised. It can be also thought that deep neural networks combine the classification process and the extraction of features. Neural networks form the basis of deep learning. The features of the problem are learned in each layer, and these learned features are used as input for the next layer. Thus, while moving from the input layer to the output layer, a structure is established in which is the simplest and the most complex features can be learned. Therefore, deep learning can be described briefly as a whole of neural networks based on deep architecture and the methods developed for them.

Neural networks with deep architecture have been widely applied to music emotion recognition recently. In this section, the proposed Convolutional, Long Short-Term Memory, Deep Neural Networks (CLDNN) architecture for music emotion recognition was described. Although it only seems to be a classifier, CLDNN (Sainath et al., 2015a, Sainath et al., 2015b) architecture consists of two basic parts. The first part is the Convolutional Neural Network for feature extraction, while the other part is the classifier consisting of Long Short-Term Memory with Deep Neural Networks. Its name comes from the combination of the first letters of these methods. The CLDNNs was applied to MER in a unified architecture. Because of the CNNs, LSTMs and DNNs are complimentary in their modeling capability. If it is explained briefly, the architecture passes features that are output of CNN by reducing spectral variations as an input to LSTM for temporal modeling, and DNN transforms the features that are output of LSTM into a more separable space.

The LSTM layer consists of 200 hidden units. The output of the LSTM is connected to two fully-connected DNN layers, which predict output targets easier. Each fully connected layer has 100 hidden units. Finally, an output layer is used to obtain final decision. The model architecture of the proposed system has been shown in Figure 3.1. to help understand the framework.

In this section, it was also given detailed information about the standard features, toolboxes and the reason why chosen MFCCs and log-mel filterbank energies as the input acoustic features.

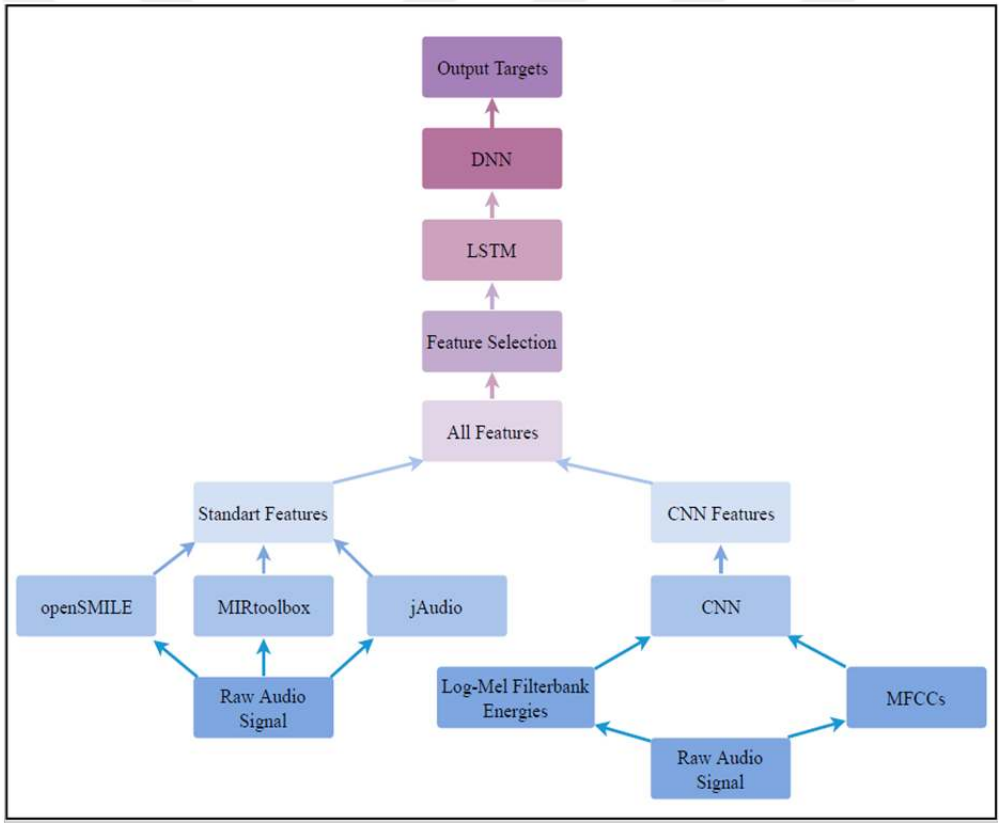

Figure 3.1. The architectures of a CLDNN.

3.1. One Dimensional CNN for Feature Extraction

Deep learning is becoming a popular machine learning subset thanks to its successful performance against many types of data. In deep learning, in many cases, large amounts of data are required to obtain quality features that represent classes. In this way, satisfactory classification performance rates are achieved, and the error difference between training data and test data is reduced. In cases where the data size is limited, the data augmentation process is necessary to deal with this problem. Data augmentation is the creation of additional data samples by applying a series of deformations to the data in the dataset. The basic principle in data augmentation is that the labels of new data created after the operation applied to the tagged data are not changed. There are many techniques for data enhancement, such as adding noise to the music or dividing each part into several equal parts. In this study, it was decided to divide each data sample into three equal parts of 10 seconds. Thus, the size of our dataset has been tripled exactly for feature extraction. Since the smaller parts will not be sufficient enough to express emotions, obtained parts are not divided into subparts.

Many researchers have investigated to find the most correlated features for music emotion recognition. Lots of features have been explored that play an important role in establishing a relationship between emotion and music. It is difficult to extract new features from music excerpts, because of the complexity of extracting relevant features. However, some new features were needed to train a model and improve classification success by ensuring the correct classification. These new features should summarize a large amount of complex multimedia data in the signal, without sacrificing the acoustic properties present in the signal and reflect the character of the signal for training models. Therefore, new features were obtained by using CNNs in our study to give a better definition of the emotions in music excerpts.

CNN’s are one type of Multi-Layer Perceptron (MLP). CNN models were developed for image processing initially. CNN’s are applied in many fields such as

audio processing (speech emotion recognition, speech recognition, music emotion recognition) and natural language processing later. CNN's are also a class of feed-forward neural networks, which consist of one or more convolutional layers. Each convolution layer has a set of filters that convolve with the corresponding layer to the inputs and generates feature maps. Convolution layers are usually followed by the pooling layer where the convolved data is down sampled usually by taking into account the average or maximum values for each small subsection of the matrix. Pooling layers allows reducing the number of parameters in the model for feature extraction. The depth of the network is very substantial to achieve better accuracy. However, as the depth of the network increases, accuracy can reach saturation and then decrease.

Although CNN's have 1, 2, or 3 dimensions, they share the same features and work in the same way. The main difference is the dimensions of the input data and how the feature detector, moves across the data. During the feature learning process, the 2D CNN model uses a two-dimensional input that represents pixels of image and color channels. This structure has been widely applied for audio processing in many studies by using MFCC features and log-mel spectrograms. MFCC features and spectrograms represent the frequency content in the audio as colors in images. These images fed to CNN to classify the emotion of music until now.

This same process is also applicable to one-dimensional sequences of data. As an alternative way, a modified version of 2D CNNs called 1D CNNs has recently been developed. A 1D CNN is a model that has a convolutional layer that runs on a 1D sequence. This may be followed by a second convolutional layer in some scenarios (long input sequences), or a pooling layer whose job is to separate the output of the convolutional layer to the most appropriate elements. After all this, the flatten layer is used between the convolutional layers to reduce the feature maps to a single one-dimensional vector. Thus, the CNN model extracts the best features from sequences data and it maps the internal features of the sequence.

1D CNNs performed well in many studies that have high signal variations and limited labeled data. These studies have demonstrated that in some cases 1D CNNs are more advantageous and therefore preferable to 2D CNNs in dealing with audio signals. For example, the computational complexity of 1D CNNs is very low compared to 2D CNNs. Generally, 1D CNNs have a small number of hidden layers and neurons that able to handle challenging tasks, so it is much easier to implement and train. A standard computer is usually enough to train and implement in 1D CNN, even extra hardware may not be required. Due to its low computational requirements, less memory and no need special hardware, 1D CNNs are particularly suitable for real-time applications in mobile or similar devices. In this thesis, 1D CNNs were used to extract features for music emotion recognition.

1D CNN can be applied to any kind of signal data over a fixed-length period such as audio signals for feature extraction. The features can be obtained from audio signals in two ways: time-domain and frequency-domain. Recently, a waveform based model has been used in many studies that receive direct raw signals in the time domain. For example, Dieleman and Schrauwen (2014) used this model on CNN models for music auto-tagging. Besides Dai et. al. (2017) used this model to Deep Convolutional Neural Networks (DCNN) for environmental sound recognition task and additionally Sainath et. al. (2015) used this model to CLDNN for speech recognition.

Considering music emotion recognition studies, it is thought that the information obtained in the frequency domain contains more important data than those obtained by using the waveform signal in the time domain. Because frequency domain features reveal deeper patterns in music signals that can help us to identify the underlying emotion under limited data. For this reason, it is aimed to construct the feature extraction process using the information in the frequency domain such as Mel Frequency Cepstral Coefficient (MFCC) and log-mel filterbank energies as input vectors.

In the thesis, two different kinds of models using MFCC features and log-mel filterbank energy features as inputs to 1D-CNN are proposed to predict the class probabilities. Because it is believed that these features will represent the raw signal in the best way. These features contain the necessary information. Another reason for using these models is that MFCCs and log-mel filterbank energies are compatible with 1D CNN. Because they are unrelated, such as the channels of an image (eg Red, Green, Blue).

In addition to CNN based MFCC and log-mel filterbank energy features, CNN based raw audio features are also obtained. However, CNN based raw audio features seem to not have too much effect on the MER performance as mentioned before (Sarkar et al., 2019). Therefore, CNN features based on raw audio were not utilized in our study.

Figure 3.2. shows the steps for computing log-mel filterbank energies and MFCCs. In the first step, the music signal is provided to pass through a pre-emphasis filter to amplify higher frequencies with smaller magnitudes compared to lower frequencies and it is divided into overlapping frames using a Hamming window with a frame size of 30 milliseconds and a frame shift of 20 milliseconds. In the second step, the discrete-time Fourier transform (DTFT) is computed for each frame. Then the square of magnitude spectrum is computed. The fourth step gives filterbank energies which are computed by summing all energies in each mel-scaled triangular filterbank. The output of the fifth step is the log-mel filterbank energies which are computed by taking the logarithm of mel filterbank energies. In the last step, MFCCs are computed by taking the discrete cosine transform (DCT) of log-mel filterbank energies. These features are fed to CNN as shown in Table 4 to generate 256 CNN based features for every 10-second music.

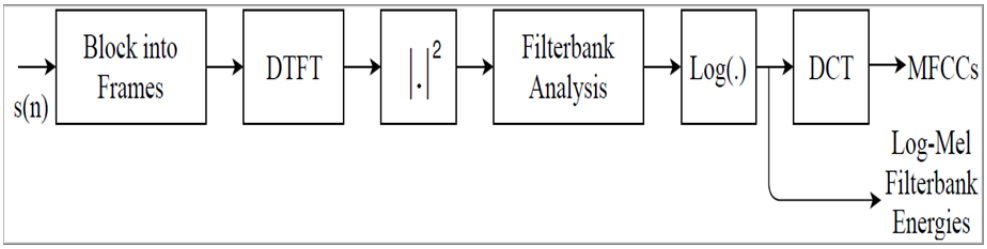

Figure 3.2. Extraction of the MFCCs and Log-mel filterbank energies

While machine learning uses a lot of mathematical and statistical methods for feature extraction, deep learning makes the feature extraction process interesting and successful thanks to convolution. As mentioned above, a great way to use deep learning to get new features is to create a convolutional neural network (CNN). The Keras library in Python makes it very easy to create a CNN. The proposed architecture is made of four convolutional layers, with four max-pooling layers and a flatten layer for feature extraction. Figure 3.3. and Table 3.1. illustrate the detailed architecture of the proposed CNN. We can also show how to develop 1D-CNN for our models as follows.

Table 3.1. Configuration of the convolutional layers (C), pooling layers (P) and flatten layer (F) for the CNN considering different types of inputs and input sizes.

Description		Log-Mel Filterbank Energies				MFCCs			
Input Shape		500 x 26				500 x 13			
		Dim	Filter	Filter Size	Stride	Dim	Filter	Filter Size	Stride
Layers	C1	491	64	10	1	491	64	10	1
	P1	122	64	4	4	122	64	4	4
	C2	120	128	3	1	120	128	3	1
	P2	30	128	4	4	30	128	4	4
	C3	28	128	3	1	28	128	3	1
	P3	7	128	4	4	7	128	4	4
	C4	6	128	2	1	6	128	2	1
	P4	2	128	3	3	2	128	3	
	F	1	256			1	256		

Let' see what is happening step by step:

- **Input data:** CNN's can adapt to different input sizes. The key in the description is the form of the input, that is what the model needs in terms of the number of features and the number of time steps as input for each sample. CNNs can be used for parallel input time series as separate channels, like blue, red, and green components of an image. Channels are different views of data. Each music excerpt divided into three 10 second segments as mentioned above, and then 13 MFCCs and 26 log-mel filterbank energies are calculated using 30 ms Hamming window every 20 ms. MFCC can be considered as an input with 13 separate channels and the log-mel filterbank energies can be considered as an input with 26 separate channels corresponding to the number of features. Since window step 20 ms is selected, there will be 500 frames corresponding to the number of time steps, in 10 seconds. As a result, 500 x 13 MFCCs and 500 x 26 log-mel filterbank energies are calculated for every 10-second of music. Therefore height, which is the length of a dataset to feed the network, is 500. The width, which is the width of the dataset to feed the network, is 13 for MFCC and 26 for log-mel filterbank energies. It is also called to as the depth.
- **First 1D Convolution Layer:** Filter size is the height of the sliding window that convolves across the data. It is also called kernel size. In 1-D CNN, kernel slides along one dimension. Filters show how many sliding windows will run through the data. It is also referred to as feature detectors. Defining only one filter allows the neural network to learn one feature on the first layer. 64 filters have been defined as it may be insufficient (a filter size of 10 applied to each filter). This enabled us to get 64 features from the first layer of the network. With a height of 500 and a filter size of 10, the window will slide through the data for 491 steps (500-

10+1). Therefore, resulting in a 491 x 64 output matrix. This initial layer will learn basic features.

- **First Max-pooling layer:** An important aspect of CNNs is pooling layers that are generally applied after convolutional layers. Pooling layers subsample inputs. The popular way to do pooling it to apply a max operation to the result of each filter. Max pooling layers prevent overfitting of the learned features by taking the max value of multiple features. In our study a size of four is chosen for this stage. This means that the size of the output matrix of this layer is only a fourth of the input matrix. A sliding window approach is used again. The pooling layer slides a window of height 4 across our data and replaces it with maximum value, so discarding %75 of our values from the previous layer. Results in a 122 x 64 matrix.
- **Second 1D Convolution Layer:** The result from the first max-pooling layer was fed into the second CNN layer. At this time, 128 different filters (a filter size of 3 applied to each filters) to be trained at this level were defined. With a height of 122 and filter size of 3, the window will slide through the data for 120 steps (122-3+1). Therefore, resulting in a 120 x 128 output matrix.
- **Second Max-pooling layer:** Filter size shows the size of the convolutional window and 4 was selected again. The strides parameter is demonstrating the shift size of the convolution window. So, the max pooling stage, the filter size value is also the stride value. The pooling layer slides a window of height 4 across our data and replaces it with maximum value again, so results in a 30 x 128 matrix.

- **Third 1D Convolution Layer and Third Max Pooling Layer:** Same structure of the previous step was applied again. After applying additional 1D convolution layer and max pooling layer set, the output matrix is a 7 x 128 matrix.
- **Fourth 1D Convolution Layer and Fourth Max Pooling Layer:** Another sequence of 1D CNN layer and max-pooling layer follows in order to learn higher level features. The output matrix after those two layers is 2 x 128.
- **Flatten layer:** After completing the previous steps, we need to have a pooled feature map. As the name of this step suggests, our pooled feature map will literally be flattened into a column, because it transforms the features into 1-dimension vectors. What happens after the flattening step is that you get a long vector of input data that you pass through the classification methods to have it processed further. As a result of all these, 256 features were obtained from every 10-second music, and in fact, 768 features were obtained from every 30-second music for each different input namely, MFCCs and log-mel filterbank energies.

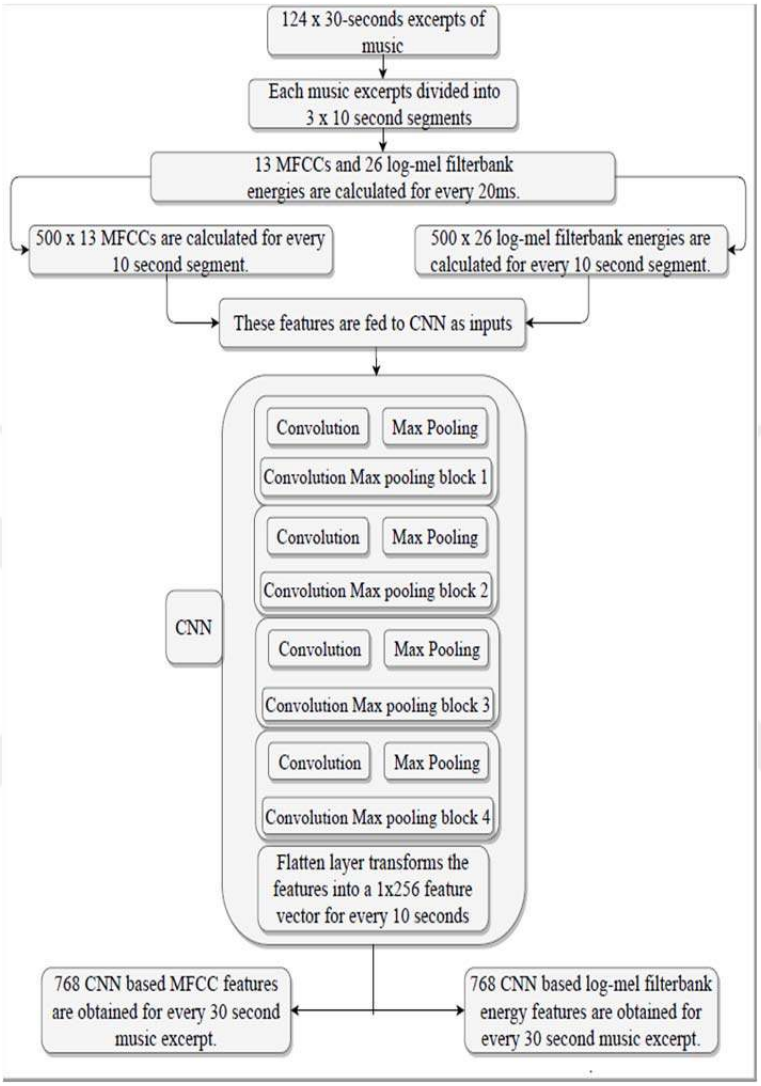

Figure 3.3. The structure of a 1D CNN, consisting of convolutional layers, max-pooling layers, and flatten layer.

3.2. Standard Features

In audio signals, it is necessary to obtain simple and meaningful data to reflect the character of the signal for training models. Feature extraction does this and it converts high-dimensional characteristic information into low-dimensional characteristic information and gets distinctive features. Although some research

has focused on seeking the most relevant features for emotion recognition, no dominant single feature has found in determining emotion. There are many acoustic features that can be calculated from the raw sound waveform of music excerpts. The success of the emotion recognition process depends on the best representation of the class properties of the features from the datasets. These features have different categories, ranging from low to high level depending on the complexity. Musical features are generally clustered into four to eight categories. Here, a four categories organization is followed, employing dynamics (energy), timbre (spectrum), harmony (melody), rhythm (tempo). These features are extracted in studies and many of these features have been used for MER. This part explains their relationship to music emotion, the meanings of these features and how they are extracted. Different emotion perceptions of music are generally related to different patterns of acoustic information. As many previous studies on MER have pointed out, arousal is usually much easier to model than valence. While there are a number of features relevant to arousal, there are few salient features for valence. For example, while valence is related to harmony (dissonant /consonant), arousal is generally related to rhythm (slow / fast), energy (low / high), and timbre (soft / bright). It is also noted that sometimes emotion perception is not only dependent one music factor but also combination of them. For example, anger is associated with high sound level, fast tempo, and elevated high-frequency energy in music.

3.2.1. Features Type

3.2.1.1. Energy Features

One of the most important features that can relate between music and emotion is energy. Because, the energy of a music is often highly associated with the perception of arousal. It can be said that music excerpts with high energy are related to happy and anger, while music with low energy is about calm and sad. For this reason, energy features are very important in the definition of high-level

features. Two basic energy features are root-mean-square (RMS) and low energy. They are the most used features to represent the dynamics in the signal.

3.2.1.2. Rhythm/ Tempo Features

Rhythm is one of the fundamental features that characterize music and the use of rhythm is also very common. In music, it consists of regular repetitions of strong or weak melodic beats. Rhythm is also closely related to the expression of emotion. Because music with fast tempo is often correlated with high arousal and slow tempo is correlated with low arousal. So, fast tempo is usually associated with emotion of happiness or anger, while slow tempo is associated with emotions of sadness or calm. In this thesis the following rhythm-related descriptors such as tempo, rhythm strength are considered.

3.2.1.3. Spectral Features

Spectral Features are highly correlated to the Timbre. Although it has been used in many studies as a feature that represents the timbre of audio and music signals, spectral features are important in distinguishing different emotions. For example, in happy music, the high frequency / brightness ratio is generally higher than sad music. Music with happy emotions usually has a larger spectral energy than music with sad emotions. Spectral features which are computed using the Short Time Fourier Transform (STFT) of an audio signal. Five basic spectral features are spectral centroid, MFCCs, spectral skewness, spectral kurtosis, log-mel filterbank energies.

3.2.1.4. Harmony/ Melody Features

The audio waveform is a periodic wave. It is possible to express a periodic wave as a combination of a series of sinusoidal waves. Sinusoidal waves that form a periodic signal are called harmonics. One of these waves, which has the same frequency as the frequency of the periodic signal, is called fundamental frequency

f0. A lot of natural sounds, especially musical ones, are harmonic. Harmonic features are related to structural information of music, and they are highly correlated with the perception of valence. Therefore, they are important for music emotion recognition. For example, simple, consonant, harmonies usually relate to emotions such as happy or relaxed. On the other hand, dissonant, complex harmonies relate to emotions such as sadness or angry, as they create instability in a musical motion. Mode, chroma, or f0 are examples of basic harmonic features that represent emotions in music.

3.2.2. Toolboxes

Nowadays, it is common to use available audio frameworks to extract standard features due to the complexity to obtain meaningful musical features. These frameworks are effective and extensible software that are prepared to extract important features based on sound analysis and synthesis. Some of these features, also called low level descriptors (LLD), are usually derived from the short-time spectra of the audio waveforms such as roll-off, slope, flux, decrease, centroid, kurtosis, spread, contrast, skewness or MFCCs (Panda et al., 2015). Other higher-level features are also extracted such as tempo (rhythm), tonality, harmony and key. In this thesis, three audio frameworks (OpenSMILE, MIRtoolbox and JAudio) were used to extract these standard features from the music excerpts.

Such frameworks represent features of audio such as timbre, loudness, harmony, tempo, pitch, rhythm and so forth to build a relationship between emotions and music. These frameworks also have many differences between each other, such as performance, ease of use, stability, the number, and type of features available.

3.2.2.1. OpenSMILE

OpenSMILE is a flexible and modular toolkit used for feature extraction in machine learning and signal processing applications. Even though OpenSMILE

was originally designed for speech processing, it has shown good performance on emotion recognition in many studies. The large openSMILE emotion feature set contains a greatly enhanced set of low-level descriptors (LLD) by applying several functionals, filters and transformations to these. The set contains 6553 features as 39 functionals of 56 acoustic LLDs which are mostly related to speech recognition such as spectral, energy, pitch, voice quality, mel-frequency and corresponding first and second order delta coefficients. These features obtained the emo-large feature set of OpenSMILE. In terms of number of features, Emo-large feature set is the largest emotion specific feature set known to date. This feature set is chosen because it would allow us to get more emotion correlated features

3.2.2.2. MIRtoolbox

The MIRtoolbox offers a set of functions to extract musical features from music excerpts such as timbre, tonality, and pitch. It is a software prepared to provide an overview of computational approaches in the field of extracting information from musical data. The main advantages of this toolbox are its integration with MATLAB (extensive documentation, community help, ease of use) and its ability to extract a large number of both low and high-level musical features. Because of these advantages, MIRtoolbox was also used in this study. There are a lot of high and low-level features in this toolbox including harmonic change detection function (hcdf), mode, inharmonicity, tonal, chromogram, key clarity, tempo, and fluctuation and 348 features were obtained using this toolbox.

3.2.2.3. JAudio

Finally, the jaudio toolbox, which ensures a comprehensive solution to the problems encountered during feature extraction, was used. JAudio is an open source library that can extract features both with a user interface and in command line mode. This toolbox unlike OpenSMILE and MIRtoolbox, allows general use of a great number of features that are both extensible and easy to use such as root

mean square, low energy, strongest beat, linear predictive coding (LPC), compactness, spectral variability, spectral smoothness. JAudio is also a tool to extract music-related features from the music excerpts such as hcdf, mode, inharmonicity, tonal, chromogram, key clarity, tempo, and fluctuation. It also uses meta features that can be applied against any feature to create new features including standard deviation, derivative, and mean. It also combined to produce two more metafeatures (derivative of the mean and derivative of the standard deviation). By obtaining 468 features with this toolbox, more features can be extracted than MIRtoolbox.

As abovementioned, frameworks such as the MIRtoolbox, OpenSMILE and JAudio offer a large number of computational audio features. All default options were used for all toolboxes in this work, and low-level features were extracted related to timbre and energy with OpenSMILE toolkit, and a more musically motivated feature set, containing high-level features related to mode, tempo, tonality, key, rhythm, and harmony, with mirtoolbox and jAudio. The features obtained from these 3 toolboxes were combined and a feature pool was created. The aim of this study was to achieve a maximum number of features that would allow us to detect emotions and achieve maximum success. Strengths and weaknesses were learned by using these tools in our studies. In addition, insight was obtained about their structure, and their benefits in emotion recognition were detected. In total, 7369 standard features have been extracted for each music excerpt, using these toolboxes. Later, these features were added to new features obtained via CNN. Most importantly, the deficiencies of these features were covered with the CNN features. It was also thought that combining standard features with CNN features would be more useful for classification.

Table 3.2. Major standard features used in toolboxes

Toolboxes	**Features**
MIRtoolbox	Harmonic change detection function (HCDF), mode, inharmonicity, tonal, chromogram, key, clarity, tempo and rhythmic fluctation
JAudio	Root mean square (RMS), low energy, linear predictive coding (LPC), compactness, spectral variability, spectral smoothness and strongest beat
OpenSMILE	Energy, MFCC, pitch, spectral centroid, roll-off, flux, skewness, slope, kurtosis, zero crossing rate, shimmer, jitter and loudness

3.3. Long Short-Term Memory

Neural network architectures have been used more frequently after 1990s. Two of these architectures stand out compared to the others. The first is the Recurrent Neural Networks developed by Hochreiter in 1991. These networks contain loops that allow information to continue and these loops are no different from a standard neural network. RNNs can be said as multiple copies of the same network, each sending a message to another.

RNN has been used successfully in many studies such as natural language processing, speech recognition, image classification and translation. However, the learning ability of these networks has been found to be limited in some cases. Sometimes, it is not enough just to look at the latest information to handle the challenging task and in cases where more content is needed, it may not be enough.

These problems in the training of standard RNN neural networks have been overcome with long short term memory (LSTM) developed by Hochreiter and Schmidhuber in 1997. It accomplished this by adding memory cells and various gates to the network.

LSTM is a very special kind of recurrent neural network that works much better than the standard version for many tasks and can learn long-term

dependencies. They exhibit the behavior of remembering information for a long time. Since this architecture can learn long-term dependencies with its memory transitive mechanism, it is widely used in temporal sequence modeling problems such as speech recognition, natural language processing, music production, and emotion classification.

However, unlike RNN, the basic LSTM architecture has 3 gates, such as the input gate, output gate and forget gate, instead of a single gate with one or more memory cells.

- Input gate, controls the entry of input activations into the memory cell.
- Output gate, transfers the output activations of the memory cell to the rest of the network.
- Forget gate, controls the flow of information from the memory block to the memory cell, and it deletes the memory of the cell.

Equations of LSTM architecture are given below:

$$f_t = \sigma\left(W_{xf}\, x_t + W_{hf}\, h_{t-1} + b_f\right), \tag{3.1}$$

$$i_t = \sigma(W_{xi}\, x_t + W_{hi}\, h_{t-1} + b_i), \tag{3.2}$$

$$\tilde{C}_t = \tanh(W_{xc}\, x_t + W_{hc}\, h_{t-1} + b_c), \tag{3.3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{3.4}$$

$$o_t = \sigma(W_{xo}\, x_t + W_{ho}\, h_{t-1} + b_o), \tag{3.5}$$

$$h_t = o_t \odot \tanh(C_t). \tag{3.6}$$

In the above equations, x_t and h_t represent input and output sequence; f_t, i_t, o_t represent forget, input and output gates, respectively. In LSTM architecture, it is decided whether to forget the information by using the information x_t and h_{t-1}. This is done using the equation 3.1. (f_t). In the forget layer and *sigmoid* is

used as activation function. Then the input layer is activated and new information is determined using equation 3.2. (i_t). The information is updated with the *sigmoid* function. $\tilde{C}_t$, is the stage where the candidate information that will create new information is determined by the *tanh* function, taking into account the input and previous situation. C_t is the step to update the state of the cell. Output data is obtained with the o_t and h_t equations. The above process continues repeatedly.

W shows weight matrices, b bias vectors and, and these parameters are estimated by the model to minimize the difference between real training values and LSTM output values. $\sigma(.)$ and *tanh* denotes the activation functions, $\odot$ denotes the element wise multiplication and t shows the time steps.

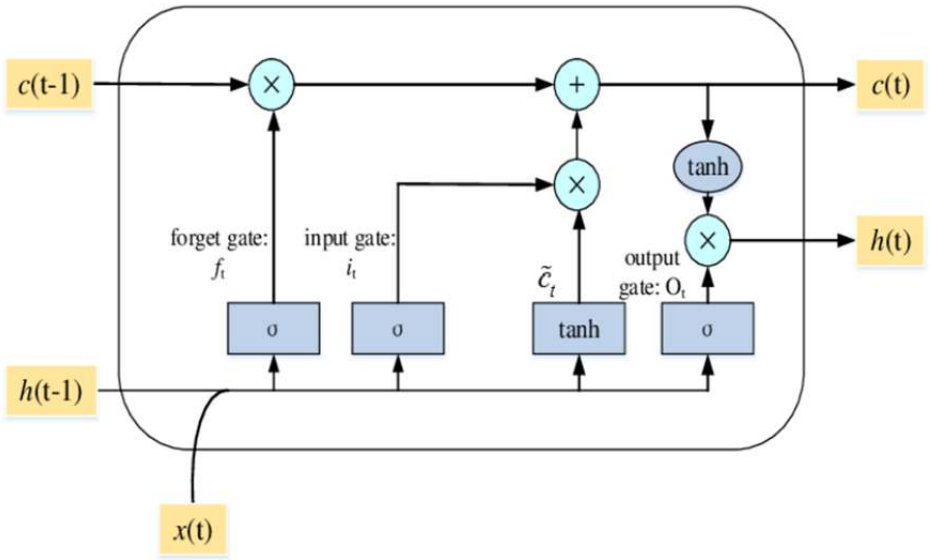

Figure 3.4. The structure of LSTM.

3.4. Fully Connected Deep Neural Networks

In addition to convolutional neural networks and long-short term memory networks, fully connected (FC) deep neural networks can be also applied to audio signals. These networks evaluate each value in the input data as time-independent features; therefore, they are not suitable for learning and recognizing patterns in audio signals directly. In fully connected deep neural networks, each neuron in a layer is connected to all neurons in the previous and next layers. Hence, the temporal context obtained after the LSTM can be introduced by feeding these networks with the information of the previous and future time steps next to the current one.

FC layers are used in the last step to solve the classification problem, as they improve output estimates by further deepening the matching between hidden units and output. These layers can also be used for classification by combining with another neural network. For example, the complementarity of LSTM and DNN layers between their modeling capabilities has proved to be successful in many studies. In addition, these layers can form part of other architectures. Within the CLDNN architecture, FC layer performs the discrimination process and, it is the last layer of the CLDNN architecture hence no other layers can be placed after this layer.

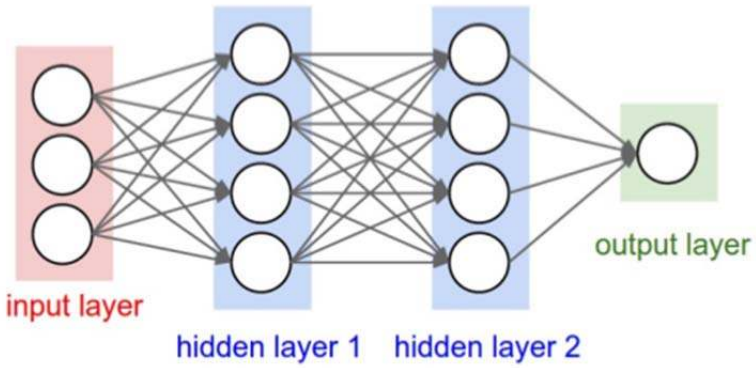

Figure 3.5. The structure of FC layers

Deep neural networks consist of at least two hidden layers apart from the input and output layers. It is often used to convert matrices into vectors. It requires a lot of computation as it uses back and forward propagation. When a sequence of input is given to the input layer of the LSTM, it is passed to the output using the previous status information. After the last element of the input is directed in this way, the last process is started. For this, two fully connected layers and a non-linear activation function called softmax are used to convert the output of the layer into probabilities. In this way, as many results are produced in the output layer as the number of classes. The final layer will reduce the vector of height 100 to a vector of three since there are three classes to be predicted (HAPV, HANV, and LANV). It is generally used in applications such as emotion classification, speaker and language recognition, where the input is processed and classified sequentially.

3.5. Feature Selection

The quality of the features obtained from the dataset is an important factor affecting the success rate as well as the classification methods. After the feature extraction process is over, there may be hundreds or thousands of features derived from music. If there are too many features to be given to the model, this makes the work of the model difficult and it increases training time and process load. Also, these features are not equally important, and it is necessary to choose the best among the available features. However, it is not clear which features influence the final accuracy. In order to eliminate unnecessary features that negatively effect on emotion recognition performance during training, various feature selection and reduction methods have been used frequently on datasets.

Feature selection is an important step for music emotion recognition. Because, feature selection methods have the aim of maximizing the prediction accuracy and find the optimal feature set. For this purpose, a feature evaluator and a search method are required to select the most relevant features available in our feature set. In our study, given the high number of extracted features, the feature

space dimensionality is reduced the initial number of features into smaller and easier to work with via feature selection. Correlation-based Feature Selection algorithm (CFS) (Hall and Smith, 1997) is applied as a feature evaluator. This method is an algorithm that aims to find lists the subset of features, which does not connect with each other but has high relation with the class by evaluating each feature subset using an objective function.

Since it is not possible to perform extensive searches along with all possible feature subsets, the sub-optimal but faster search functions such as hill climb, genetics, greedy, random and best first are generally selected in many studies. The best first method is used as a search method for selecting a good feature subset in our study. In order to perform the feature selection process, the tools implemented in the Weka 3.8. (Hall et al., 2009) are used.

Using Equation 3.7. CFS calculates a heuristic measure of 'merit' of the feature subset F as:

$$Merit_F = \frac{nt_{cf}}{\sqrt{n+n(n-1)t_{ff}}}, \tag{3.7.}$$

where n is the number of features in sub-feature space F, t_{cf} is the average class-feature correlation, and t_{ff} is the average feature-feature inter-correlation CFS calculates t_{cf} and t_{ff} using a symmetric information gain (Hall and Smith, 1999) .

The results were calculated separately for two different categories. The first one is classification that according to the annotations made by the experts and the other one is the classification that according to the annotation made by the native-listeners. As a result of feature selection process, different sets of features were obtained.

3.6. Classification Methods

Machine learning methods are algorithms that can learn the model over existing data and make predictions about the unknown. Machine learning is generally classified into three broad categories. These are unsupervised learning, supervised learning, and reinforcement learning. Basically, supervised learning is learning in which the machine is trained using well-labeled data. Classification is a supervised learning problem as the class label for each training group is provided.

Classification is one of the popular and essential tasks of machine learning. It is the process of determining which class the sample belongs to with the help of a classifier by using the feature vectors of the sample. In classification problems, each element in the output space is called a class, and the algorithm that solves the classification problem is called a classifier.

In order to design a classifier, the feature database obtained from the samples is divided into two groups as "training" and "test", some are used to train the classifier and the other to test. There must be a dataset that belongs to certain classes beforehand for training. When new samples are added to the dataset for testing, the aim is to place them in the appropriate class from the existing class with the highest accuracy. In other words, the purpose of the classification is to create a model by using the features of the samples known to which class they belong and use this model in the classification of new samples with unknown features. If the success rate of the model is sufficient, it can be said that the obtained model is effective in solving the current problem.

In the success of recognition systems, the method of classification is as important as feature extraction, and often it cannot be said with certainty which classification method is the best or the most suitable. Factors such as the structure and the number of samples, processing time, and complexity can make one classification method more successful than another.

The WEKA tool was used in the classification stage of this study. There are many data mining methods such as classification and clustering algorithms in

this tool. When the studies on emotion analysis are examined, it is seen that supervised machine learning methods are more preferred. Sequential minimal optimization, Random forest, K-nearest neighbor, Naïve Bayes and LDNN methods were used in the study of emotion recognition from music. The parameters of the algorithms to be compared are left to be the default values of the WEKA program, and changes that may affect the model performance positively or negatively are avoided.

3.6.1. K-NN

K nearest neighbors (k-NN) is one of the most popular machine learning algorithms because it is used for both classification and regression problems, is simple. It generally provides efficient performance, and in some cases, its accuracy is higher than modern classifiers.

This method has been proposed by T.M. Cover and P. E. Hart (1967). The k-NN algorithm has three key elements: samples whose class is known, the distance metric for calculating the distance between samples, and the k value for the number of nearest neighbors.

The easy output interpretation and predictive power are regarded as advantages over other algorithms. It can be seen as its disadvantage that the calculation cost is high, and it works slowly compared to other methods. Because the distance from each new sample to each sample in the training data set is calculated individually and the computational complexity increases with the size of the training data set.

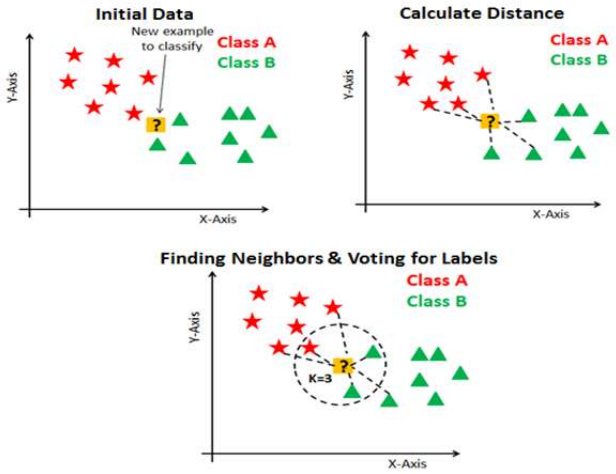

Figure 3.6. The structure of k-NN

In this method, the classification process is made by taking into account the calculation of the distances between the sample to be classified and the existing classified sample. The closest k neighbors to the sample to be classified are determined and ranked, then the classification is made by looking at the class labels of these k neighbors. Therefore, this algorithm is briefly called k-NN. By giving many different values for k, the best "k" value can be found. Various distance criteria such as Euclidean, Minkowski, and Manhattan can use to calculate these distances. 'IBk' under the 'lazy' group is selected in Weka tool to implement the model using K-nearest neighbors with default parameters.

3.6.2. Random Forest

The Random Forest (RF) algorithm was developed in 2001 by Breiman. RF is one of the most used machine learning algorithms. Because it is an easy-to-use, flexible and successful method. The advantage is that it can be used for both classification and regression problems and the training time is less than other classifiers. It can also maintain accuracy when a large proportion of data is missing.

RF is a method that operates by constructing multiple decision trees during the training stage. According to this method, more than one decision tree is created, and it is aimed to combine the resulting decisions to obtain a more accurate and stable prediction as a result of training each of the multivariate trees with different training sets. Decision Tree is a tree-shaped diagram used to determine a course of action. These decision trees are randomly selected subsets from the dataset. It reduces the risk of overfitting since it uses more than one decision tree. Each decision tree created is left in its widest form and is not truncated. Each branch of the tree represents a possible reaction or decision. For classification, trees are created so that each leaf node contains members of only one class. Therefore, the classification or decision are carried in leaf node. Decision trees that are established different from each other form the decision forest community that will lead us to results. The results obtained during the formation of the decision forest are combined and the final estimate is made. The decision of the majority of the trees is selected as the final decision by the random forest.

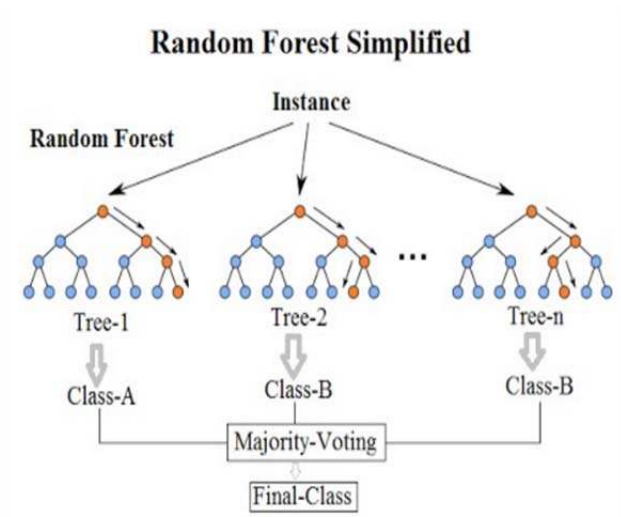

Figure 3.7. The structure of Random Forest

This algorithm is based on 2 different parameters, the number of variables to be used in the node and the number of trees to be developed, respectively. 'RandomForest' under the 'trees' group is selected in Weka tool to implement the model using Random Forest with default parameters.

3.6.3. Naïve Bayes

Naïve Bayes classifier, a classifier named after the English mathematician Thomas Bayes, who lived in the 17th century, and based on the probability theorem, is applicable to high dimensional data. It is a simple, easy and well-performing algorithm that is widely applied. The probabilistic model of Naive Bayes classifiers is based on Bayes' theorem. The underlying rationale is to calculate the highest probability output for a group of observed samples to predict the class.

This classifier goals to determine the class of data which are sent to the system for testing, with a series of calculations defined according to probability principles. The Naïve Bayes method assumes that every feature of a class is equal and independent of other features. That is, all features affect the probability of outcome independently and the presence or absence of any feature does not affect the other features.

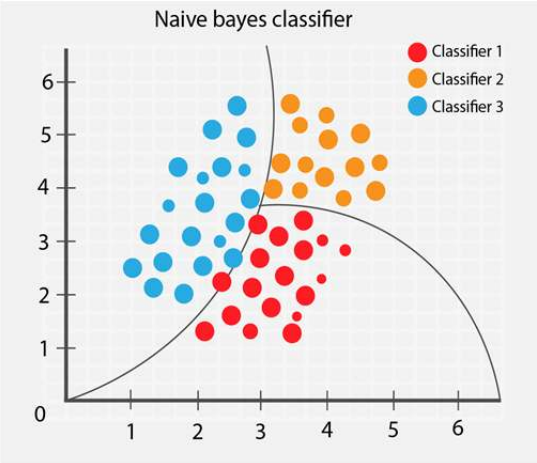

Figure 3.8. The structure of Naïve Bayes

Its advantages are that it can be applied quickly and easily in data classification and performs well in multi-class predictions. Another advantage of this classifier is that it needs only a small amount of training data to predict the parameters required for classification. The disadvantage is that if the features are not really independent, the 'independent feature model' cannot accurately model the problem and it may not always be easy to find independent features in real life. Another disadvantage of this classifier is that if a categorical variable that is not in the training dataset occurs during the test, the Naïve Bayes model calculates the zero probability and cannot make a prediction.

'Naïve Bayes' under the 'bayes' group is selected in Weka tool to implement the model using Naïve Bayes with default parameters.

3.6.4. Sequential Minimal Optimization

The Sequential Minimal Optimization (SMO) Algorithm is a support vector machine (SVM) based algorithm developed by John Platt (1998). It is a conceptually simple, easy to implement, and popular SVM training algorithm. It is often used to solve optimization problems during the training process of support vector machines. It aims to achieve the smallest possible optimization result at every stage. One reason for this is that SVM's training algorithms are slow, especially in big problems. Another reason is that SVM training algorithms are complex and difficult to solve. This algorithm is usually better and for faster difficult problems than the standard SVM training algorithm.

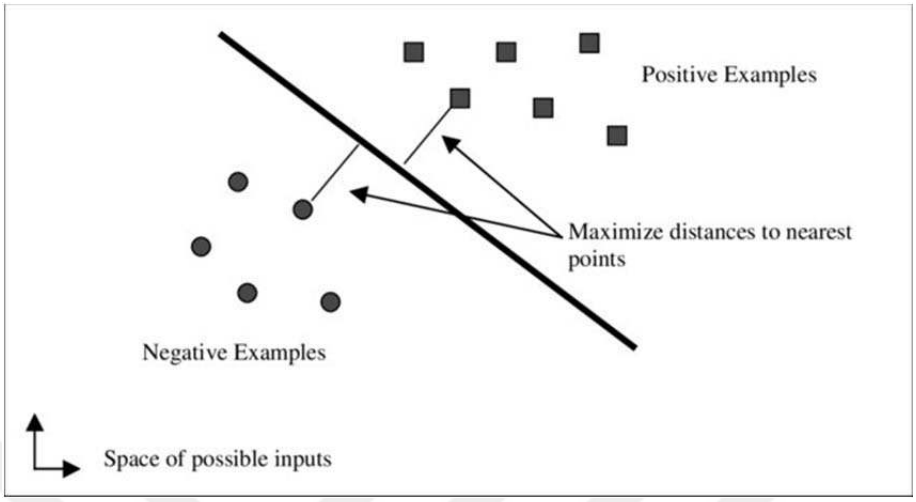

Figure 3.9. The structure of SVM

SVM requires a large number of quadratic programming calculations. Unlike SVM, SMO has been developed to solve these calculations faster by turning them into sub-problems without extra matrix storage. The first step is to set the starting parameters. The second step is the loop procedure. Loop procedure; it performs three basic operations: selecting two Lagrange multipliers, following their joint improvement, and updating SVM parameters. It consists of 2 loops, internal and external. While the best data is selected in the external loop, there are two Lagrange multipliers based on these data in the internal loop. At each stage, SMO finds optimal values for the two lagrange multipliers, and SVM parameters are updated to reflect the best values. The advantage of SMO is that it can solve two lagrange multipliers analytically. Thus, the problem of quadratic programming is prevented. Loops are run until all samples reach the desired level.

In addition to classification problems, SMO is also used as SMOreg in regression processes. ‘SMO’ under the ‘functions’ group is selected in Weka tool to implement the model using sequential minimal optimization with default parameters.

3.6.5. LDNN

CLDNN has been widely accepted in many tasks before, as the CNN, LSTM, and DNN modules being complementary in their modeling capabilities and were shown to outperform LSTM-only models. Thus, music emotion recognition performance can be improved by combining CNNs, LSTMs, and DNNs in a unified framework.

CNNs excels at extracting location knowledge, LSTMs are good at temporal modeling and DNNs are proper for mapping features into a separable area. CNN, LSTM and DNN can be also used separately. However, in this study, they are unified in a framework and trained jointly.

The CNN module in CLDNN extracts features of the inputs. The outputs of CNNs are used as features and combined with other standard features. After obtaining the musical features and the ground truth labeling, the next process is to train a deep learning model to find the correlation between features and emotion labels. Therefore, features are passed to the LSTM module to model temporally the sequence input. The LSTM is added between CNN and DNN as it has been discovered to show better performance, when higher quality features are provided. DNN module provides a mapping between outputs and hidden units.

A LDNN (LSTM+DNN) model, consisting of a long short-term memory (LSTM) layer followed by two fully connected neural layers (FC), serves to be the common sub-network architecture for each of the proposed models. Hence, LDNN is utilized to classify each model and then an output softmax layer is used to complete the system. Thus, CLDNN architecture is completed.

Spectral-temporal features are considered as input to a CLDNN model such as MFCCs and log-melfilterbank. Because an emotional utterance is represented by a sequence of spectral features. In order to understand the role played by a convolutional layer in learning affective information in music, two models in which MFCCs and log-mel filterbank energies are used as input are presented.

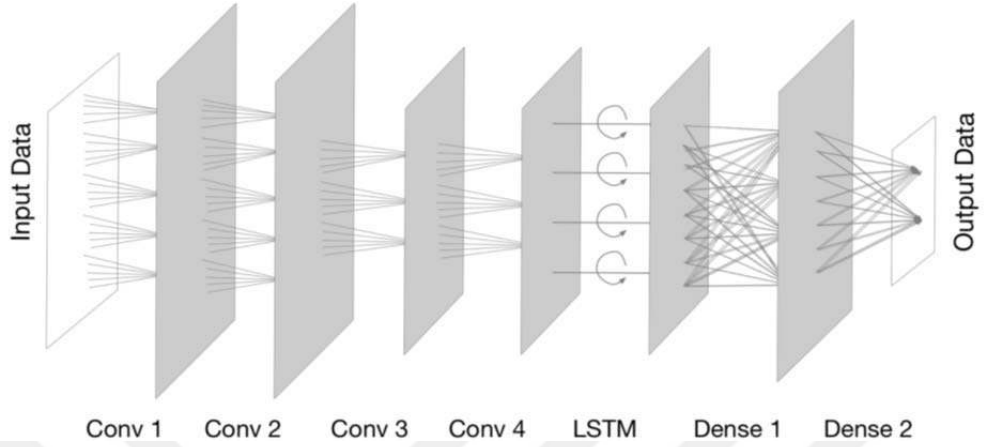

Figure 3.10. CLDNN architecture

One layer of LSTM is used with 200 cells the output of the LSTM is passed to two FC layers with 100 hidden units. For activation functions, the DNN layers uses ReLU and the output layer exploits softmax to output probability scores.

This study was conducted to find the best classification model in deep neural networks for 2 different types of listeners. For this purpose, 16 different models were created using different hyperparameters such as optimization method, learning rate, epoch number and by comparing the performances of the created models, the best model for classification wanted to be determined.

4. EXPERIMENTAL SETUP

The success of the music emotion recognition system depends on the database used in modeling a system. It is crucial for music signals to give an acceptable level of emotion. In this study, a new dataset has been constructed for music emotion recognition. Our dataset consists of 124 Turkish music excerpts drawn from Youtube videos and each music annotated by 2 experts and 21 native listeners. The music excerpts are trimmed to 30 seconds manually by using the Audacity program and converted to same format as 44,100 Hz, 16 bits, and mono channel PCM WAV. Music excerpts were selected to cover a variety of styles and genres. Dataset was constructed by expert control. Creating a dataset in this way was a labor-intensive and time-consuming process that limited the feasibility of increasing the number of music excerpts and reflected our decision to emphasize quality more than quantity in constructing the dataset.

In the literature, two different approaches in determining the emotional content of the music signal are considered. The most common and classic one is, the content of the music signal is the classification including different categories of emotions (sad, angry, happy, etc.). The second method is using an evaluation of the multidimensional space of emotions. Multidimensional approaches, unlike the categorical classification approach have further generalization. Although fewer studies focused on using multi-dimensional approaches, there has been an increase regarding these studies at a remarkable level work recently.

AnnoEmo, was used to evaluate dimensionally for subjective testing. Annotators assigned values to the music between -5 and 5 for valence and arousal but then we've normalized these values between -1 and 1. These annotation results were used in 5 different ways in our study. These are the average and median annotation results of native listeners, the average annotation results of the experts and finally the expert1 and expert2 annotation results individually. Average can be defined here as the sum of annotations divided by the number of annotators.

Median can also be defined as the annotations that are left in the middle when the annotations are sorted from large to small.

Another important problem in the design of a music emotion recognition system is the extraction of appropriate features that efficiently characterize different emotions. It is believed that the suitable feature selection significantly affects the classification performance. Several audio features are extracted from our dataset to better explain the relationship between musical emotions and features. Three audio frameworks - OpenSmile (6553 features), MIRToolbox (348 features) and JAudio (468 features), were used to extract these features from the music excerpts as mentioned before. Additionally, programming languages such as perl, matlab, python, and java were used to obtain these features. MER can be formulated both as a classification and a regression problem, depending on the underlying emotional model, making it possible to apply almost any machine learning algorithm to MER. Some approaches will be explained below and results obtained using using these approaches will be reported. It must be noted that different classification, regression, approaches use different evaluation metrics, which makes it difficult to compare the achieved accuracy. However, regression was used here to test the success of standard features. Regression analysis is an analysis method that allows finding the cause-effect relationship between variables. The idea behind regression is to predict an actual value, based on a previous training examples. Since, the Russell's model is a dimensional representation of emotion, a regression algorithm is used to train two different dimensions as valence and arousal.

7369 standard features and 124 music excerpts were used to train the following three popular regression algorithms: K-Nearest Neighbors (K-NN), Random Forest (RF), Sequential Minimal Optimization Regression (SMOreg). The feature selection method is applied called correlation-based feature selection (CFS) to bring better results using less features. These algorithms and selection method were run on Weka.

Table 4.1.- 4.5. shows the results of each regression method in terms of valence and arousal according to different evaluation values (median, average). The performance of the regression methods was evaluated by 10 fold cross-validation. To ensure the reliability of the result, no instance of a test subject is allowed to be in the train dataset of each fold. For each experiment, the feature selection will perform by appliying CFS to the training set of each fold. In order to measure the performance of the regression models, correlation coefficient (r) (Edwards, 1976) and, mean absolute error (MAE) (Willmott and Matsuura, 2005) were used. MAE measures the average magnitude of the errors in a set of predictions, regardless of their direction. It's the average over the test sample of the absolute differences between actual and prediction observation where all individual differences have equal weight. The mean absolute error, MAE is defined as,

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right|, \tag{4.1}$$

where n is the number of music excerpts, y_j is the value predicted and $\hat{y}_j$ is the value predicted by the model.

The correlation coefficient (r), displays the degree of linear relationship between two variables. So, given pairs of values for variables Y and X, designated (x, y), r is given by the following formula:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 + \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4.2}$$

where n is the number of music excerpts, x_i is the value predicted, $\overline{x}$ is the mean of the value predicted, y_i is the value predicted by the model and $\overline{y}$ is the mean of the value predicted by the model.

When the results are examined, it is seen that the outperform results were yielded by using only standard audio features based on arousal and valence. Also, it was found that better results could be obtained by finding new features because there is still much room for improvement. For this purpose, novel audio features that best represent emotion are began to research. Additionally, these results showed that the best results for use in the next stage were obtained when average values were chosen for both experts and native listeners. Considering that the results could be improved, new features were obtained by using CNN later.

Table 4.1. Native listeners average regression results in terms of valence and arousal according to different methods.

Native Listener Average									
Arousal	Standard Features		CFS		Valence	Standard Features		CFS	
	CC	MAE	CC	MAE		CC	MAE	CC	MAE
SMOreg	0,854	0,249	0,9	0,17	SMOreg	0,829	0,283	0,842	0,248
KNN	0,907	0,173	0,916	0,167	KNN	0,784	0,244	0,838	0,206
RF	0,927	0,168	0,95	0,148	RF	0,868	0,229	0,921	0,203

Table 4.2. Native listeners median regression results in terms of valence and arousal according to different methods.

Native Listener Median									
Arousal	Standard Features		CFS		Valence	Standard Features		CFS	
	CC	MAE	CC	MAE		CC	MAE	CC	MAE
SMOreg	0,880	0,23	0,887	0,192	SMOreg	0,831	0,276	0,842	0,258
KNN	0,907	0,173	0,916	0,158	KNN	0,776	0,277	0,882	0,208
RF	0,923	0,17	0,934	0,168	RF	0,867	0,24	0,922	0,203

Table 4.3. Experts average regression results in terms of valence and arousal according to different methods.

Experts Average									
Arousal	Standard Features		CFS		Valence	Standard Features		CFS	
	CC	MAE	CC	MAE		CC	MAE	CC	MAE
SMOreg	0,773	0,293	0,892	0,211	SMOreg	0,708	0,289	0,770	0,26
KNN	0,828	0,228	0,854	0,221	KNN	0,662	0,297	0,83	0,219
RF	0,862	0,221	0,907	0,197	RF	0,778	0,254	0,854	0,221

Table 4.4. Expert 1 average regression results in terms of valence and arousal according to different methods.

Expert 1									
Arousal	Standard Features		CFS		Valence	Standard Features		CFS	
	CC	MAE	CC	MAE		CC	MAE	CC	MAE
SMOreg	0,78	0,3	0,846	0,267	SMOreg	0,721	0,293	0,732	0,329
KNN	0,853	0,203	0,894	0,171	KNN	0,659	0,306	0,761	0,269
RF	0,88	0,212	0,919	0,193	RF	0,767	0,269	0,86	0,230

Table 4.5. Expert 2 average regression results in terms of valence and arousal according to different methods.

Expert 2									
Arousal	Standard Features		CFS		Valence	Standard Features		CFS	
	CC	MAE	CC	MAE		CC	MAE	CC	MAE
SMOreg	0,730	0,323	0,753	0,334	SMOreg	0,649	0,317	0,758	0,264
KNN	0,746	0,282	0,822	0,252	KNN	0,622	0,326	0,672	0,302
RF	0,827	0,249	0,864	0,235	RF	0,752	0,259	0,838	0,234

When the selected average values were analyzed, it was seen that there were concentrations in 3 different quarters in the 2-dimensional plane, and therefore, instead of regression, it was decided to make classification by adapting the values to 3 classes.

4.1. Development Environments and Platforms

There are many open-source platforms that can be used on artificial intelligence, machine learning and deep learning. In studies on deep learning, a large number of studies may be required to find out which data in the data set will be used and which data will produce the best result. Many criteria can be effective when deciding which of these platforms to correct such as fast working, visualizing the result in the best way, statistically informing with many criteria, processing on common file formats and ease of use. Therefore, platforms can be superior to each other in many ways. In some platforms, even coding and programming experience is not required. Therefore, choosing the best platform suitable for the desired work; will give researchers an advantage in terms of time, correct results and cost. Matlab, Keras, Weka, Python, C++ and Java can be said as the most preferred tools and languages in these areas.

4.1.1. Weka

Weka (Waikato Environment for Knowledge Analysis) is a data mining and machine learning software. It has been developed at Waikato University, New Zealand in 1993. Experimental studies were carried out by the WEKA software. Open source Weka software is developed with Java, which is an object-oriented programming language because the use of Java is a platform where many different learning algorithms can be used regularly and effectively. There are many libraries on machine learning and statistics available on WEKA. You can perform data preprocessing, regression, classification, clustering, or feature selection with help of user-friendly graphical user interface (GUI). It is also a platform commonly used for academic purposes in many studies.

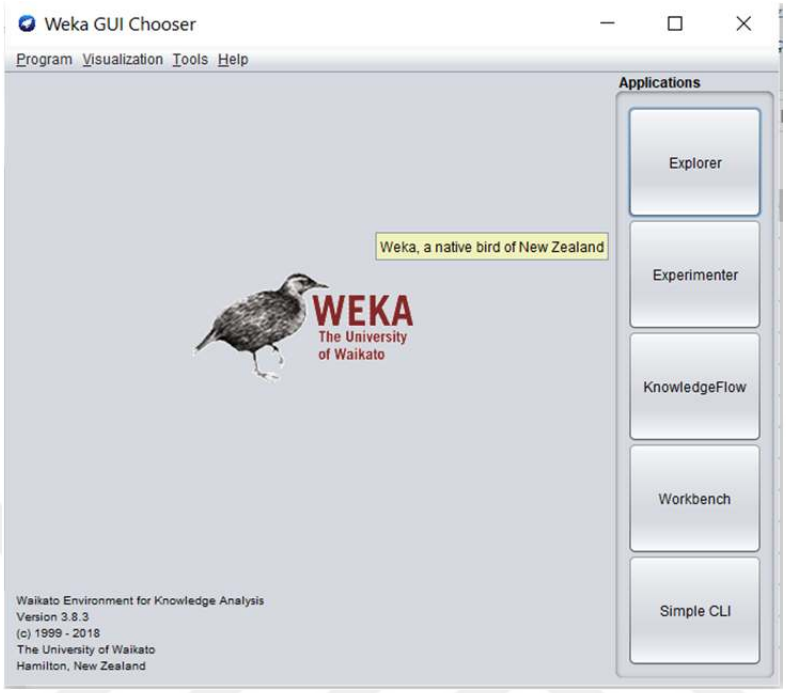

Figure 4.1. The interface of Weka

Arff which is a kind of data file type, is specially designed for Weka and it is an abbreviation of the Attribute Relationship File Format. In other words, if it will work with a data set other than the data sets provided by the Weka library, it is necessary to convert them to the arff file format first. There are two parts in the arff format, the first one is the header part, which consists of the relation name, attributes and attributes value types, the second one is the data part where each line contains the features of the samples.

All the features obtained with the help of different tools and found in different file types were merged and converted into arff format thanks to a code we wrote in java. Different feature sets in arff format are also combined using the Simple CLI (Command Line Interface) command line interface, which is part of Weka. On the other hand, files such as c45, libsvm, csv (Comma Separated Values), which are frequently used in data mining, are also used in Weka.

With Weka, several machine learning methods can be applied on the dataset to be used and relevant parameters can be set for training and test datasets to be used in the classification process easily. Various feature selection algorithms can be applied to select the best features used on the dataset and data can be visualized. In our study, various methods such as feature selection algorithms and classification algorithms were applied to the data set in the ARFF file with the Weka and the results were examined in detail in Section 5.

WEKA's current program does not have deep learning algorithms inside the original set of classifiers. However, there is an external package known by name WekaDeeplearning4j (Lang et al., 2019) which providing access to modern techniques of deep learning in Weka. This deep learning package allows users to create their own architectures using different layers and algorithms. The method named Dl4jMlpClassifier brought with this package can be used for both regression and classification. It enables deep learning architectures such as recurrent neural networks (e.q. long short-term memory networks), convolutional neural networks, and it supports users the ability to test and train deep neural networks in Weka.

4.1.2. Keras

There are many ready-made libraries with different features that have been developed by various universities and companies for deep learning, which make these studies practical and easy. Each of these libraries has different functions. Appropriate libraries should be selected according to the subject to be studied.

Anaconda is a free and open-source distribution that is used with Python and R programming languages in areas such as machine learning, deep learning and data mining, which aims to facilitate package management and distribution for researchers. Installing the libraries needed in Python programming language one by one causes a waste of time. Since Anaconda contains many libraries such as numpy, pandas, mathplotlib, required for deep learning architectures, when you

install anaconda there is no need for additional downloads for libraries and Python programming language.

Python is an open source programming language that is growing in popularity around the world, written by a Dutch programmer named Guido Van Rossum and Jelke de Boer (1991). It is effectively used in many areas of engineering. Python is also a interpreted, high-level, and general-purpose dynamic programming language that focuses on code readability. Python and its modules have been used in this thesis from the neural network generation to the data processing. All of the Python libraries and the modules had been used extensively and freely.

Keras is a high-level Application Programming Interface (API) written in Python that can use Tensorflow, Microsoft Cognitive Toolkit (CNTK) or Theano libraries as a backend. It frees the user from the complexity of these lower-level libraries, making it easier and faster for researchers to define deep learning models, to determine their stages and training processes. It is also user-friendly, modular and extensible. Keras is one of the APIs that support a wide variety of neural network layers such as recurrent layers, dense layers, or convolutional layers by allowing to create sequential models and it can be easily run on the Graphics Processing Unit (GPU) or Central Processing Unit (CPU). It was developed by a Google engineer, Francois Chollet (2015), in order to facilitate rapid experimentation. It can be used well in areas such as translation, image recognition, speech recognition, emotion recognition.

Throughout this thesis, using an effective development environment for coding in the Python programming language will be of great benefit in the development process. JetBrains Pycharm is an integrated development environment (IDE) program written for Python programming language. With this IDE program, projects can be developed very quickly and easily because this IDE used to have the best control possible on code development is PyCharm. It has been developed to work on all operating systems. CNN was developed by using

Python language in the JetBrains PyCharm IDE environment and Keras was run on CPU as the platform to design a model for feature extraction.

Microsoft Windows 10 Home Single Language was used as the operating system. The computer processor used for software development is Intel (R) Core (TM) i7-6700HQ CPU @ 2.60GHz. The system has an x64-based processor and has 16.00GB of RAM.

4.2. Cross Validation

Cross-validation is a statistical method used to evaluate and compare the results of the learning algorithm. Weka enables cross-validation, which is a very useful process for evaluation, on the used learning algorithm. In n-fold cross-validation, the dataset is randomly divided into n equal parts. After that, data is stratified to get approximately the same class distribution for all the training sets. Then, n-1 parts are used as training data, the remaining part is used for testing to calculate an error rate. This process is repeated n times until each subset is used as both training data and test data.

The error rate of the model is found by taking the arithmetic mean of the n error rates calculated separately to get the most realistic result. The success rate of the model depends on two factors while performing N-fold cross validation. One of them is a training set, the other is the dividing parts. In our study, 10-fold cross-validation was used for the regression and classification process. The 10-fold cross-validation process is shown in Figure 16 in order to understand easily.

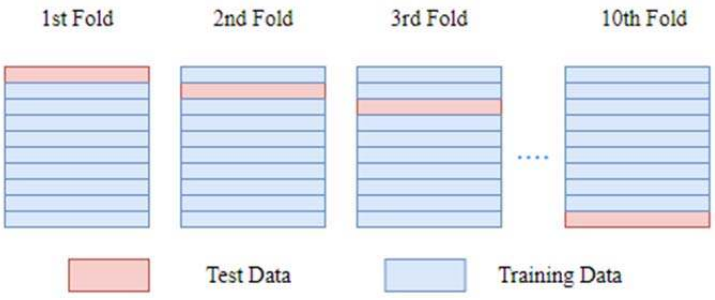

Figure 4.2. Ten-fold cross-validation

4.3. Classification Performance Evaluation Metrics

Experimental studies have been done with the WEKA program developed on JAVA language with open source code and a 10-fold cross-validation method was used to evaluate and compare the results as mentioned before. There are many metrics such as accuracy, confusion matrix, error rate, recall, precision and f-measure used to evaluate the performance and success of a classifier method applied to the dataset. Metric selection greatly affects the measurement and comparison of the classification algorithms performance.

4.3.1. Accuracy

The simplest and the most popular metric among these is the accuracy rate of the model. The success of the model in music emotion recognition is related to the number of correctly classified samples and the number of incorrectly classified samples. During the test on the model, in order to measure the accuracy rate, the training data should not be processed. Data other than the training data is called test data. After the model is created on the training data, the model created is tested in the test data. The success of the results achieved by the test can be found through the confusion matrix. The confusion matrix shows not only how the model is predicted, but also where the problematic processes are located. As seen in Figure 17, in the confusion matrix, the rows represent the real numbers of the music excerpts in the test set, and the columns represent the predictions of the model.

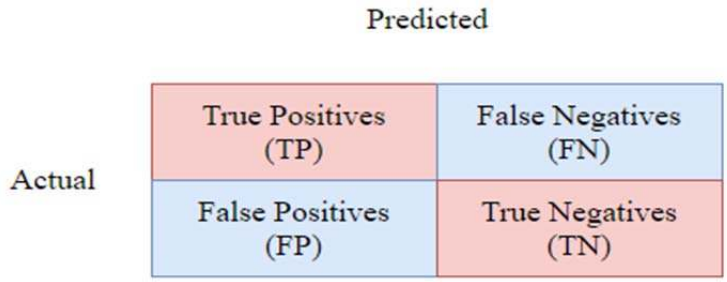

Figure 4.3. Confusion matrix

True Positive (TP): It shows the number of music excerpts whose actual class is the same as the predicted class.

True Negative (TN): It shows the number of correctly classified music excerpts belonging to the other class.

False Negative (FN): It shows the number of music excerpts whose actual class is predicted as another class.

False Positive (FP): It shows the number of music excerpts classified as another class.

$$Average\ Accuracy = \frac{\sum_{i=1}^{l} \frac{TP_i + TN_i}{TP_i + FN_i + FP_i + TN_i}}{l} \times 100 \quad (4.3)$$

The results are presented in terms of average accuracy (Equation. 4.3), where l, i, FN_i, FP_i, TN_i, and TP_i refer to number of class, number of music excerpt, false negatives, false positives, true negatives and true positives. It is calculated by dividing the number of true classified music excerpts $(TN + TP)$ by the total number of music excerpts $(TN + TP + FN + FP)$. Average classification accuracy (in percent) is found by averaging each class accuracy.

When the number of classes is more than two, this 2 x 2 confusion matrix will take the form of an expanded confusion matrix in $l \times l$. In a confusion matrix of $l \times l$ dimensions, the main diagonal shows correctly estimated music excerpt numbers while the matrix elements outside the main diagonal represent the incorrectly estimated results. Accuracy is a metric that is widely used to measure the success of a model but appears to be insufficient alone.

4.3.2. Precision

It is calculated by dividing the number of true predicted positive class (TP), by the total number of predicted positive class ($TP + FP$). The precision is shown as an equation as follows:

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{4.4}$$

4.3.3. Recall

It is calculated by dividing the number of true predicted positive class (TP), by the total number of actual positive class ($TP + FN$). The recall is shown as an equation as follows:

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{4.5}$$

4.3.4. F-measure

Precision and recall metrics are not sufficient to achieve a meaningful comparison result alone. Evaluating both metric together gives more accurate results. For this, F-measure has been defined. The F-measure is the harmonic mean of precision and recall. This value can be a maximum of 1 and it indicates how accurately the class is classified by the classifier algorithm. As the value approaches 1, the classification of the relevant class by the algorithm means that it is so successful. The F-measure is always closer to the smaller than the precision and recall rates. It is more successful as a performance metric since it contains both false positive and false negative values.

$$F_i = \frac{2 \times P_i \times R_i}{P_i + R_i} \tag{4.6}$$

4.4. Datasets

In this thesis, Bi-Modal (Malheiro et al. 2016) and Soundtrack (Eerola and Vuoskoski, 2011) were also used to compare the success of the study in music emotion recognition with other studies and show the effect of the dataset on this success compared to other datasets. These datasets were chosen because they are similar to our dataset.

These music excerpts have been converted to uniform format by using the Audacity program and they have standardized as 44,100 Hz, 16 bits, and mono channel PCM WAV before the feature extraction process.

4.4.1. Bi-Modal Dataset

The Bi-Modal dataset, published by Malheiro et al. (2016), consists of 162 songs containing various music genres from different eras. Each song is of 30 seconds duration and songs are distributed to 4 quadrants of the Russell emotional model, with the number of songs 45, 52, 34 and 31, respectively. It can be said that these quadrants correspond to anger, happy, tender, and sad respectively.

The songs were annotated manually by 39 people from different backgrounds. Unlike our dataset, the songs in this dataset were contained lyrics and consisted of songs in different languages. Therefore, annotators made classification by both listening to the audio and reading the lyrics. Annotators were introduced to assign values between -4 and 4 to arousal and valence for both audio and lyrics. If two different emotions were obtained as a result of this evaluation, they were asked to choose the dominant one. The arousal and valence of each song were obtained by the averaging the annotations of all the annotators for both, audio and lyrics. Then, the standard deviation of the annotations made by different annotators for the same song was calculated.

In this study, the audio signal part is considered and lyrics are ignored. This dataset is similar to our dataset in many aspects such as its size, preparation,

annotation, etc. Therefore, this data set was used to test whether our study will be successful in similar data sets.

4.4.2. Soundtrack

The Soundtrack (Stimulus Set 2) dataset (Eerola and Vuoskoski, 2011), consists of 110 music excerpts from movie soundtracks with different film genres. Each music excerpt is annotated with five different emotion categories (anger, sad, happy, fear, tender), and three-dimensional model (valence, energy, tension) by annotators. The dataset was constructed into two blocks: block 1 with the 50 discrete emotion excerpts and block 2 with the 60 dimensional model excerpts. While the discrete emotions were determined in the first 50 music excerpts, the emotions in the remaining 60 music excerpts were considered to represent the highest-matching emotion according to the mean ratings given by the annotators. Looking at the assigned values, it was noticed that there was a high correlation between anger and fear so these two classes were combined and considered as one class. As a result, music excerpts are distributed to 4 discrete emotion: happy, angry, sad and tender with the number of songs 23, 39, 27 and 21, respectively.

The annotators were 116 university students aged 18-42 years (%68 females and %32 males). Each music excerpt is of 15 seconds duration and does not contain lyrics, dialogue or sound effects. The reason for choosing this data set was that it was decided to be used because the soundtracks of the movie, although produced for a different purpose, were composed with the intention of expressing emotions strongly.

4.5. Hyperparameters

Parameters whose choices are left to the person designing the model and vary according to the problem and data set are called hyper-parameters. The multi-layer architectural structures that came with deep learning brought along a series of hyperparameter groups waiting to be decided by the researcher. As with other

algorithms, these hyperparameters should be adjusted ideally in order for deep learning algorithms to give the best results. Because the ideal hyperparameter selection provides the realization of artificial neural network training with less cost and high performance. Some of these parameters consist of a certain number of options such as optimization algorithm and activation function and are not difficult to choose. However, there are also hyperparameter types such as the number of layers, number of neurons, learning rates that wait for us to choose from a cluster that is not within certain limits and extends to infinity.

In this study, parameter optimization was carried out intuitively and it has been tried to optimize 3 different parameters of the deep learning algorithm. Various values of these parameters were used for classification, and the performance and results were compared and interpreted. Details of each hyperparameter will be explained below.

Let's see what these basic hyperparameters which are related to neural network structure:

- **Number of hidden layers and number of hidden units**: The width of the network refers to the amount of hidden units in a hidden layer and the depth of the network refers to the number of hidden layers found in the network. Increasing the number of layers and units generally increases success while at the same time increasing the computation time proportionally. Therefore, the number of layers and units are special hyperparameters that need to be considered when building the model.
- **Number of epochs**: Epoch indicates the number of times the training process will be repeated. The number of times that all samples in the dataset will be trained is adjusted by making changes over this value. In this study, 2 different epoch numbers, 50 and 100, were used.

- **Neural network learning rate:** Learning rate is a parameter used in optimization algorithms. The learning rate ensures the convergence of these algorithms and plays a critical role in training the model. Updating of parameters in deep learning is done by "backpropagation" process. In the backpropagation process, first of all, for this update process the difference is found by taking the backward derivative called "chain rule". The difference value is then multiplied by the "learning rate" parameter. Finally the result is subtracted from the weight values and new weight values are calculated. If this rate is too high, it may not reach the target; it can circle or diverge around its global minimum point. If the coefficient is chosen too small, the convergence will take too long as the algorithm will proceed with very small steps in each loop. For different The "learning rate" parameters (0.0001, 0.00001, 0.000001, 0.0000001) were employed in this study.
- **Optimization algorithms**: When a neural network is trained, it uses an algorithm called an optimizer to determine the optimal weights for the model. Optimizers are used to make the difference smallest between the output value produced by the network and the actual value. The basic option is Stochastic Gradient Descent (SGD) (Bottou, 1991). If the mini-batch value is chosen as "1", the stochastic gradient descent algorithm will be applied; in this case, the error is calculated for one sample at a time. Accordingly, the weights are updated, hence the speed brought about by vectorization is lost. Another common algorithm is Adaptive Moment Estimation (ADAM) (Kingma and Ba, 2014), which computes adaptive learning rates for each parameter dynamically. Adam is a more efficient, adaptive optimization algorithm that can be used instead of the classic SGD method. It is computationally efficient and requires low memory requirements. Adam and SGD algorithms were used within the scope of

the study. Other algorithms are Adagrad, RMSprop, AdaDelta and Adamax.

- **Batch size:** "Batch size" value is used to get small groups of data to be trained. During model training, not all of the data is simultaneously inserted into the model. Because, when all data is added to the training at once, it requires very large data processing capacity, therefore the training is carried out in this way by taking as many parts as desired from the data set with this value. Since the large batch size caused to poor generalization and deterioration in the quality of the model, batch size 10 was used in this study.
- **Activation function:** The values taken as input into the neural network are calculated and value is created as an output. Whether or not these created values will be used is determined by the activation function. The activation function can affect the network's ability to converge and learn for different ranges of input values, as well as the speed of training. The most used activation functions are softmax, sigmoid, rectified linear unit (Relu), and hyperbolic tangent (tanh). Sigmoid, which is the activation function frequently used in classification problems in artificial neural networks, converts incoming inputs into outputs between 0 and 1. Generally, the sigmoid function is used in studies with two classes, while the softmax function is used in studies with more classes. The hyperbolic tangent activation function converts incoming inputs into outputs between -1 and 1. The relu is a function that converts negative values to 0, and keeps other values unchanged. The final layer will reduce the vector of size to three since there are three classes to be predicted. Softmax forces all three outputs of the neural network to sum up to one. The output value will represent the probability for each of the three classes in the output layer. The class with the highest value will be our predicted value. All layers that require an activation function in the model used ReLU activation. The only

exception was the output layer, which used a softmax activation to categorize the result of the network.

- **Dropout:** One of the problems that can occur while doing deep learning applications is overfitting. The dropout layer should be added to avoid overfitting, which is more common in cases where the data set consists of a small number of samples. This parameter ensures that the number of neurons determined in fully connected layers is randomly killed during the training phase. A value between 0 and 1 can be entered. Since dropout rate is chosen as 0.05, %5 of the neurons will take a zero weight. When this process is applied, the network becomes less sensitive to react to smaller variations in the data, and more successful results are obtained in training processes.
- **Loss Function:** This value is the error rate that occurs between the actual value to be estimated and the estimated value. This value is low during training indicates that the success of training has increased. Cross entropy and mean square error are the two main loss functions used when training neural network models. Since MER is a multi-class classification problem, cross-entropy loss function is used.

To summarize briefly, the models was trained using ADAM and SGD optimizers with 4 different learning rates. A batch size of 10 samples was chosen for training the model and trained up to 50 and 100 epochs with early stopping. Also, the model compiled using categorical cross-entropy as, its loss function to optimize neural networks. RELU and Softmax is used as activation function in the model. In addition, dropout rate is chosen as 0.05, to reduce overfitting (Lemaire and Holzapfel, 2019).

5. RESULTS AND DISCUSSIONS

5.1. Results

In this study 3-class music emotion classification is carried out since music excerpts are distributed into three quadrants of arousal-valence plane based on the distribution of the average of the annotator's ratings. A labeled database which is consists of 124 music with a lot of features extracted from each music was created. Results were presented for three quadrants HAPV, HANV, and LANV. Different information sources are used for feature extraction. In addition to standard audio features, CNN is used to extract features using MFCCs and log-mel filterbank energies as inputs. After extracting all useful features and selecting the optimum feature set, the training and testing phase was initiated to measure the performance of the classifiers. For this purpose, Weka pattern recognition tool is used. Comprehensive experiments have been conducted with each classification method to compare the performance of our LDNN method. K-NN, SMO, Naïve Bayes, and RF are considered for benchmarking. Ten-fold cross-validation method is employed for training the classifiers in every experiment. For each experiment, the feature selection is performed using the CFS algorithm in the Weka pattern recognition tool to the training set of each fold and finding best result. Several performance metrics are considered to evaluate the performance of the classifiers. The results are presented in terms of accuracy, recall, precision and f-measure. The all evaluation metrics were then calculated by averaging results from dataset.

During the development, data enhancement was done, and training design are shown of great improvements. A number of experiments were carried out with different settings and hyperparameters. In this section, created models and hyperparameter adjustments, comparison of algorithms according to datasets, comparison of our results with similar studies, and findings are presented in detail.

5.1.1. Results based on Native Listeners Evaluation

This study was carried out to find the best classification model in deep neural networks and compare this model with the most popular methods. For this purpose, 16 models were created using 2 different optimization methods (Sgd and Adam), 4 different learning rates (from 0.0001 to 0.0000001), and 2 different epoch numbers (50,100). The performances of the created models were compared and the best model was determined for classification according to native listener annotations. According to the results in Table 5.1. It has been observed that the performance of the models varies depending on the parameters. In the most successful model, ADAM was used as the optimization method, 0.0001 as the learning rate and 100 as the number of epochs. When CFS is applied to this model in which all features are used, it has been observed that %99,19 recognition success is achieved in our dataset.

Table 5.1. The classification performance of the proposed LDNN architecture using different hyperparameters.

Accuracy Results		ADAM		SGD	
Learning Rate	Epoch Number	Full Feature Set	CFS Applied	Full Feature Set	CFS Applied
0,0001	50	89.51	97.58	88.70	95.96
	100	**91.93**	**99.19**	89.51	**98.38**
0,00001	50	86.29	94.35	87.09	93.54
	100	89.51	95.96	89.51	94.35
0,000001	50	86.29	93.54	87.90	93.54
	100	89.51	93.54	**90.32**	93.54
0,0000001	50	87.09	93.54	86.29	94.35
	100	91.12	94.35	87.09	95.96

Then, the classification performance of the proposed LDNN architecture was evaluated using various feature sets. Features are obtained by feeding different information sources into CNN layer. These information sources are log-mel filterbank energies and MFCCs that are labelled as fea_set1 and fea_set2, respectively. The classification performances of proposed feature sets are compared to that of standard audio features (fea_set3) extracted from music excerpts. Feature level combinations of feature sets and the effect of feature selection on classification performances are also evaluated. The performances are shown in Table 5.2. in terms of classification accuracy. The best result is achieved with combined feature set. Specifically, by combining new feature sets (fea_set1, and fea_set2) and standard features set (fea_set3), 6.45 points improvements over using only standard feature set is achieved as shown in Table 12. This result suggest that new features provides additional discriminative information for the music emotion recognition.

Table 5.2. The classification performance of the proposed LDNN architecture using different feature sets.

Feature Type	Full Feature Set	CFS applied
Fea_set1	87.09	93.54
Fea_set2	90.32	96.77
Fea_set3	87.09	92.74
Fea_set1 + Fea_set3	88.70	98.38
Fea_set2 + Fea_set3	88.70	96.77
Fea_set1 + Fea_set2 + Fea_set3	**91.93**	**99.19**

Table 5.3. shows the number of features before and after feature selection process for each feature set. Results indicated that the sizes of feature sets are reduced substantially by using correlation based feature selection. Also as can be seen from Table 5.2. that improved performances are achieved by employing

feature selection. The numbers of selected features from each feature sets is given in Figure 5.1. It can also be observed from the figure that there are features selected from each feature sets.

Table 5.3. The number of features and the number of features selected by CFS for each feature set.

Feature Type	# of features	# of selected features
Fea_set1	768	61
Fea_set2	768	50
Fea_set3	7369	80
Fea_set1 + Fea_set3	8137	103
Fea_set2 + Fea_set3	8137	112
Fea_set1 + Fea_set2 + Fea_set3	8905	110

Table 5.4. Comparison of performance of proposed methods with other classifiers before and after feature selection in terms of accuracy using standard features.

		Accuracy	F-Measure	Precision	Recall
LDNN	Full Feature Set	87.09	0.864	0.861	0.871
	CFS	**92.74**	**0.926**	**0.925**	**0.927**
K-NN	Full Feature Set	83.87	0.839	0.840	0.839
	CFS	89.51	0.904	0.919	0.895
RF	Full Feature Set	87.09	0.843	0.844	0.871
	CFS	91.93	0.918	0.917	0.919
Naïve Bayes	Full Feature Set	87.09	0.855	0.848	0.871
	CFS	**92.74**	**0.925**	**0.923**	**0.927**
SMO	Full Feature Set	87.09	0.866	0.862	0.871
	CFS	90.32	0.905	0.907	0.903

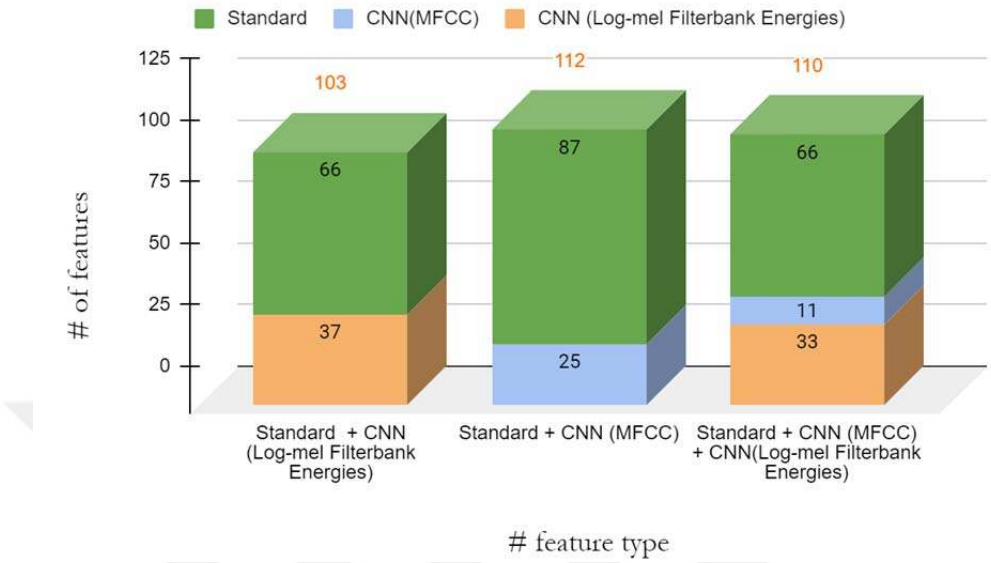

Figure 5.1. The number of features selected by CFS from each feature types in combined feature set.

These results show that proposed features based on CNN provides additional information to music emotion classification. Performance of the LDNN classifier is compared with the k-NN, SMO, Naive Bayse and Random Forest classifiers. The results are presented in detail in Table 5.4. and Table 5.5.. The standard feature set is considered in Table 5.4. while combined feature set (fea_set1 + fea_set2 + fea_set3) is considered in Table 5.5.

Results show that, the LDNN yields 1.61, 1.61, 2.42 and 3.23 points improvements in music emotion recognition accuracies compared to that of k-NN, SMO, Naïve Bayes and Random Forest classifiers respectively. The detailed classification results in terms of recall, precision and f-measure for each emotion class (HAPV = high arousal positive valence, HANV = high arousal negative valence, LANV = low arousal negative valence) for each classifier are given in Table 5.6.

Table 5.5. Comparison of performance of proposed methods with other classifiers before and after feature selection in terms of accuracy using all features.

		Accuracy	F-Measure	Precision	Recall
LDNN	Full Feature Set	91.93	0.915	0.915	0.992
	CFS	**99.19**	**0.992**	**0.992**	**0.992**
K-NN	Full Feature Set	88.70	0.878	0.874	0.887
	CFS	97.58	0.976	0.978	0.976
RF	Full Feature Set	87.90	0.852	0.854	0.879
	CFS	95.96	0.957	0.961	0.960
Naïve Bayes	Full Feature Set	86.29	0.834	0.879	0.863
	CFS	96.77	0.967	0.969	0.968
SMO	Full Feature Set	89.51	0.904	0.919	0.895
	CFS	97.58	0.974	0.977	0.976

Table 5.6. Performance of classifiers in terms of precision, recall, and f-measure for each class before and after feature selection.

		Full Feature Set			CFS applied		
		Recall	Precision	F-Meas.	Recall	Precision	F-Meas.
LDNN	HAPV	0.973	0.924	0.948	1.000	0.987	0.993
	HANV	0.545	0.750	0.632	0.909	1.000	0.952
	LANV	0.921	0.946	0.933	1.000	1.000	1.000
K-NN	HAPV	0.960	0.889	0.923	0.973	1.000	0.986
	HANV	0.273	0.429	0.333	0.909	1.000	0.952
	LANV	0.921	0.972	0.946	1.000	0.927	0.962
RF	HAPV	0.973	0.859	0.913	0.987	0.949	0.967
	HANV	0.091	0.500	0.154	0.636	1.000	0.778
	LANV	0.921	0.946	0.933	1.000	0.974	0.987
Naïve Bayes	HAPV	0.868	0.943	0.904	0.947	1.000	0.973
	HANV	0.091	1.000	0.167	0.818	1.000	0.900
	LANV	0.973	0.830	0.896	1.000	0.949	0.974
SMO	HAPV	0.974	0.925	0.949	1.000	1.000	1.000
	HANV	0.091	0.589	0.455	0.727	1.000	0.842
	LANV	0.987	0.881	0.931	1.000	0.962	0.980

5.1.2. Results based on Experts Evaluation

The performances of the created models were compared and the best model was determined for classification according to expert annotations. According to the results in Table 5.7. It has been observed that the performance of the models varies depending on the parameters. In the most successful model, as in native listeners, ADAM was used as the optimization method, 0.0001 as the learning rate and 100 as the number of epochs again. When CFS is applied to this model in which all features are used, it has been observed that %99,19 recognition success is achieved again in our dataset.

Table 5.7. The classification performance of the proposed LDNN architecture using different hyperparameters.

Accuracy Results		ADAM		SGD	
Learning Rate	Epoch Number	Full Feature Set	CFS Applied	Full Feature Set	CFS Applied
0,0001	50	87.90	97.58	87.09	94.35
	100	**89.51**	**99.19**	87.09	**95.16**
0,00001	50	87.90	92.74	83.87	92.74
	100	87.90	94.35	**90.32**	92.74
0,000001	50	83.87	91.12	83.87	90.32
	100	84.67	91.93	84.67	91.93
0,0000001	50	86.29	92.74	85.48	90.32
	100	87.90	92.74	87.09	91.93

Then, the classification performance of the proposed LDNN architecture is evaluated using various feature sets. Features obtained by feeding log-mel filterbank energies and MFCCs are labelled as fea_set1 and fea_set2, respectively. The classification performances of proposed feature sets are compared to that of standard audio features (fea_set3) extracted from music excerpts. Feature level

combinations of feature sets and the effect of feature selection on classification performances are also evaluated. The performances are shown in Table 5.8. in terms of classification accuracy. The best result is achieved with combined feature set. Specifically, by combining new feature sets (fea_set1, and fea_set2) and standard features set (fea_set3), 5.65 points improvements over using only standard feature set is achieved as shown in Table 5.8. This result suggest that new features provides additional discriminative information for the music emotion recognition.

Table 5.8. The classification performance of the proposed LDNN architecture using different feature sets.

Feature Type	Full Feature Set	CFS applied
Fea_set1	87.90	90.32
Fea_set2	**91.93**	95.96
Fea_set3	86.29	93.54
Fea_set1 + Fea_set3	88.70	95.96
Fea_set2 + Fea_set3	87.90	95.96
Fea_set1 + Fea_set2 + Fea_set3	89.51	**99.19**

Table 5.9. shows the number of features before and after feature selection process for each feature set. Results indicated that the sizes of feature sets are reduced substantially by using correlation based feature selection. Also as can be seen from Table 5.8. that improved performances are achieved by employing feature selection. The numbers of selected features from each feature sets is given in Figure 5.2. It can also be observed from the figure that there are features selected from each feature sets.

Table 5.9. The number of features and the number of features selected by CFS for each feature set.

Feature Type	# of features	# of selected features
Fea_set1	768	50
Fea_set2	768	69
Fea_set3	7369	96
Fea_set1 + Fea_set3	8137	112
Fea_set2 + Fea_set3	8137	113
Fea_set1 + Fea_set2 + Fea_set3	8905	139

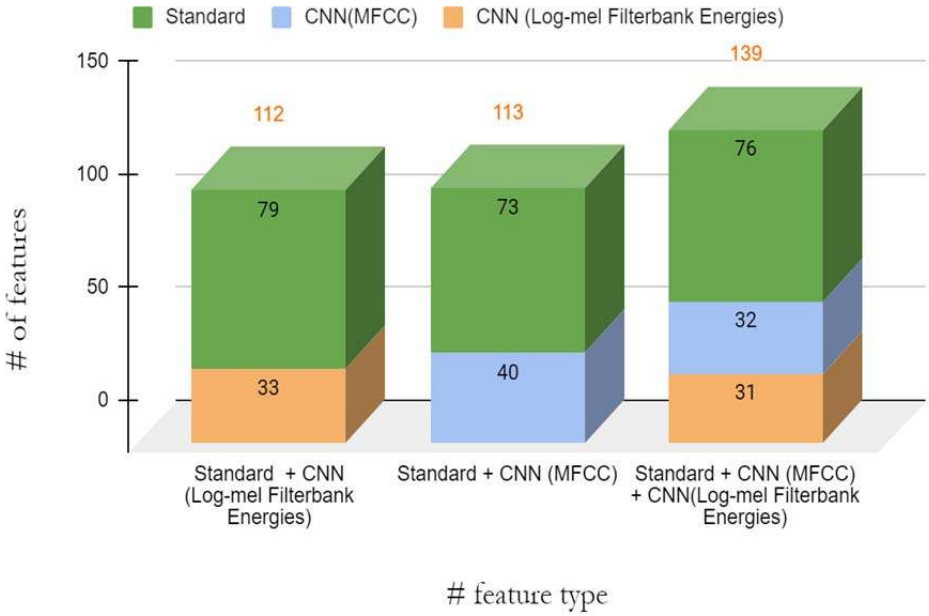

Figure 5.2. The number of features selected by CFS from each feature types in combined feature set.

Table 5.10. Comparison of performance of proposed methods with other classifiers before and after feature selection in terms of accuracy using standard features.

		Accuracy	F-Measure	Precision	Recall
LDNN	Full Feature Set	86.29	0.860	0.861	0.863
	CFS	**93.54**	**0.936**	**0.936**	**0.935**
K-NN	Full Feature Set	82.25	0.833	0.846	0.823
	CFS	84.67	0.849	0.851	0.847
RF	Full Feature Set	86.29	0.834	0.876	0.863
	CFS	90.32	0.886	0.910	0.903
Naïve Bayes	Full Feature Set	84.67	0.832	0.825	0.847
	CFS	91.12	0.909	0.909	0.911
SMO	Full Feature Set	88.70	0.873	0.875	0.887
	CFS	91.12	0.907	0.906	0.911

These results show that proposed features based on CNN provides additional information to music emotion classification. Performance of the LDNN classifier is compared with the k-NN, SMO, Naive Bayse and Random Forest classifiers. The results are presented in detail in Table 5.10. and Table 5.11.. The standard feature set is considered in Table 5.10. while combined feature set (fea_set1 + fea_set2 + fea_set3) is considered in Table 5.11.

Results show that, the LDNN yields 5.65, 0.81, 4.84, and 6.45 points improvements in music emotion recognition accuracies compared to that of k-NN, SMO, Naïve Bayes and Random Forest classifiers respectively. The detailed classification results in terms of recall, precision and f-measure for each emotion class for each classifier are given in Table 5.12.

Table 5.11. Comparison of performance of proposed methods with other classifiers before and after feature selection in terms of accuracy using all features.

		Accuracy	F-Measure	Precision	Recall
LDNN	Full Feature Set	89.51	0.890	0.896	0.895
	CFS	99.19	0.992	0.993	0.992
K-NN	Full Feature Set	86.29	0.867	0.873	0.863
	CFS	93.54	0.935	0.936	0.935
RF	Full Feature Set	85.48	0.827	0.868	0.855
	CFS	92.74	0.922	0.931	0.927
Naïve Bayes	Full Feature Set	86.29	0.826	0.879	0.863
	CFS	94.35	0.937	0.948	0.944
SMO	Full Feature Set	90.32	0.892	0.903	0.892
	CFS	98.38	0.984	0.984	0.984

Table 5.12. Performance of classifiers in terms of precision, recall, and f-measure for each class before and after feature selection.

		Full Feature Set			CFS applied		
		Recall	Precision	F-Meas.	Recall	Precision	F-Meas.
LDNN	HAPV	0.943	0.846	0.892	0.971	1.000	0.986
	HANV	0.538	0.875	0.667	1.000	0.929	0.963
	LANV	0.934	0.922	0.928	1.000	1.000	1.000
K-NN	HAPV	0.857	0.909	0.882	0.943	0.892	0.917
	HANV	0.615	0.500	0.552	0.769	0.909	0.833
	LANV	0.908	0.920	0.914	0.961	0.961	0.961
RF	HAPV	0.886	0.861	0.873	0.971	0.895	0.932
	HANV	0.154	1.000	0.267	0.538	1.000	0.700
	LANV	0.961	0.849	0.901	0.974	0.937	0.955
Naïve Bayes	HAPV	0.886	0.912	0.899	0.971	1.000	0.986
	HANV	0.077	1.000	0.143	0.538	1.000	0.700
	LANV	0.987	0.843	0.909	1.000	0.916	0.956
SMO	HAPV	0.971	0.872	0.919	0.971	0.971	0.971
	HANV	0.385	0.833	0.526	0.923	1.000	0.960
	LANV	0.903	0.900	0.892	1.000	0.987	0.993

5.1.3. Results of the Proposed System with Other Datasets

In this study, Bi-Modal and Soundtrack datasets were used to compare the success of the proposed method in music emotion recognition with other methods, to show the effect of the dataset on this achievement, and to compare with our dataset. Because these datasets are widely used in many studies (Sarkar et al., 2019). In the previous stage, all parameters and conditions used in the study on our dataset were used in these datasets. Therefore, the proposed method, which shows the highest recognition accuracy, was created in the same way, and was tested on these datasets.

5.1.3.1. Results of the Proposed System with Soundtrack Dataset

The Soundtrack dataset consists of 110 music excerpts that are distributed to 4 quadrants with the number of excerpts 23, 39, 27, and 21, respectively. It can be said that these quadrants correspond to happy, anger, sad and tender. The dataset contains music excerpts from movie soundtracks with different film genres, and do not contain lyrics, dialogue or sound effects. This dataset differs from other datasets used in this study in that it consists of soundtracks. Therefore, this data set was used to test whether the proposed approach will be successful on a dataset that consists of music excerpts that are created for a different purpose.

Table 5.13. The classification performance of the methods using standard feature set.

Classification Methods	Full Feature Set	CFS applied
LDNN	65.45	**79.09**
K-NN	59.09	71.81
RF	61.81	77.27
Naïve Bayes	58.18	72.72
SMO	64.54	78.18

Table 5.13. shows a comparison of the performance of the proposed method with other classifiers in terms of accuracy using standard features (fea_set3) before and after feature selection. When looking at the results, it is seen that the most successful method is LDNN and it is followed by SMO, RF, Naïve bayes and K-nn, respectively.

Table 5.14. The classification performance of the methods using combined feature set.

Classification Methods	Full Feature Set	CFS applied
LDNN	66.36	**87.27**
K-NN	60	71.81
RF	63.63	86.36
Naïve Bayes	60.90	85.45
SMO	67.27	85.45

On the other hand, Table 5.14. shows the accuracy results when using the combined features (fea_set1 + fea_set2 + fea_set3). The best result is achieved with combined feature set. The results show that the highest recognition rate with %87.27 belongs to LDNN again. It is followed by RF, SMO, Naïve bayes and K-nn, respectively. Specifically, by combining new feature sets and standard features set, 8.18 points improvements over using only standard feature set is achieved as shown in Table 5.13.

5.1.3.2. Results of the Proposed System with Bi-Modal Dataset

The Bi-Modal dataset consists of 162 songs that are distributed to 4 quadrants with the number of songs 52, 45, 31, and 34, respectively. It can be said that these quadrants correspond to happy, anger, sad and tender. Unlike our dataset, this dataset consists of more data and the classes are almost evenly distributed. Also, the dataset contains songs from a different language and culture, and there are lyrics within the song. Therefore, the proposed approach was applied to test for

a dataset with different characteristics. The results were obtained using only the features obtained from the songs, the lyrics were not used.

Table 5.15. shows a comparison of the performance of the proposed method with other classifiers in terms of accuracy using standard features (fea_set3) before and after feature selection. When looking at the results, it is seen that the most successful method is Naïve bayes and it cannot be said that LDNN is successful for this dataset when standard features are used.

Table 5.15. The classification performance of the methods using standard feature set.

Classification Methods	Full Feature Set	CFS applied
LDNN	63.58	70.37
K-NN	54.32	65.43
RF	63.58	74.69
Naïve Bayes	56.79	**76.54**
SMO	69.13	71.60

On the other hand, Table 5.16 shows the accuracy results when using the combined features (fea_set1 + fea_set2 + fea_set3). The best result is achieved with combined feature set. The results show that the highest recognition rate with %96,91 belongs to LDNN. It is followed by SMO, RF, Naïve bayes and K-nn, respectively.

Table 5.16. The classification performance of the methods using combined feature set.

Classification Methods	Full Feature Set	CFS applied
LDNN	72.22	**96.91**
K-NN	61.11	85.18
RF	64.19	91.97
Naïve Bayes	69.13	88.88
SMO	79.62	95.06

5.2. Discussions

In this thesis, general problems have been identified for the recognition of emotion from music, and it has been aimed to develop approaches to overcome these problems and increase the success of classification. For this purpose, a new dataset consisting of Turkish music has been created in order to obtain a good music emotion recognition system. This dataset was annotated by two different groups (native listeners, experts), the classes were determined according to these annotations, and classification was made according to these classes. Looking at the results, it has a recognition success of %99,19 in the classification made according to both evaluations. Looking at all the results, it can be said that the evaluations made by the native listeners make it more accurate than the experts.

In this study, a new method is proposed to obtain features via CNN. Log mel filterbank energies and MFCCs are used to obtain new features as input. When the effects of these two different inputs on the results are examined, it is seen that the features obtained using MFCC increase the recognition performance more than the log-mel filterbank energies. Also, when these features are combined with standard features, better results are obtained compared to the standard features. For example, we achieved between 8.07 and 4.03 points improvements on the Turkish Emotional Music Database for native listeners, as shown in Table 5.4 and Table 5.5. Also, as can be seen from Table 5.10 and Table 5.11. that the combined features yielded between 8.87 and 3.32 points improvement on the Turkish Emotional Music Database for experts. On the other hand, according to the results in Table 5.13. and Table 5.14, improvements were obtained between 0 and 12.73 points on the Soundtrack Database for all classifiers. Finally, as can be seen from Table 5.15. and Table 5.16. that performances improved between 12.34 and 26.64 points on the Bi-Modal database. As a result, the new features are significantly effective in improving performance.

As a classifier, LDNN, which is a classification method based on deep learning that complements CNN in the CLDNN architecture. The CLDNN and has not been applied for music emotion recognition before. This method includes

various hyperparameters as mentioned before. When the effect of these parameters on the result is examined, it is seen that the results obtained by using ADAM, one of the optimization methods, are better than SGD in almost all cases. In the learning rate, which is one of these parameters, it is seen that the best results are obtained when 0.0001 is used.

Sequential minimal optimization, K nearest neighbor, Random forests and Naïve Bayes methods, which are frequently encountered and widely applied in many studies, were used to compare the success of the system. After applying CFS and decreasing the number of features, it is seen that the LDNN gives the best results among all methods in almost all datasets. For example, the LDNN classifier yielded 1.61, 1.61, 2.42, and 3.23 points improvements on the Turkish Emotional Music Database for native listeners and 5.65, 0.81, 4.84, and 6.45 points improvements for experts compared to K nearest neighbors (k-NN), Sequential Minimal Optimization (SMO), Naïve Bayes and Random Forest (RF) classifiers, respectively. On the other hand, as can be seen from Table 5.14. and Table 5.16. the LDNN classifier was yielded 15.46, 0.91, 1.82, and 1.82 points improvements on the Soundtrack Database and 11.73, 5.94, 8.03, and 1.85 points improvements on Bi-Modal Database compared to k-NN, RF, Naïve Bayes and SMO classifiers, respectively. In terms of recognition rate, LDNN is usually followed by SMO, RF, Naïve Bayes and K-nn, respectively.

6. CONCLUSION AND FUTURE WORK

Recognizing emotion from music is still quite a difficult task today. People can feel different emotions from the same music because the emotion perception of people is inherently subjective. The perceived emotion can vary based on age, gender, character, lyrics, or environmental factors. Therefore, these factors cause listeners to evaluate music differently.

Music emotion recognition systems are used for many kinds of research such as music therapy and treatment of emotional disorders. In addition, many music suggestion applications that service according to the mood of people such as Spotify have been developed recently. These applications automatically analyze, recognize and classify the emotional content of music using signal processing and machine learning techniques.

In order to create a music emotion recognition system, firstly, a labeled emotional music database is needed. Labeling quality plays an important role in the success of the methods used on the data. It was seen in the literature review that there was no Turkish music dataset among the studies publicly available. Therefore, new Turkish Emotional Music Database composed of 124 Turkish traditional music excerpts with a duration of 30 seconds is constructed to evaluate the performance of the approaches in this thesis. The music excerpts were annotated on valence and arousal dimensions. 3-class was used for music emotion recognition since music excerpts are distributed into three quadrants based on the distribution of the average of the annotator's assessments.

In this thesis, we propose an approach for music emotion recognition based on convolutional long short term memory deep neural network (CLDNN) architecture. This architecture was previously applied to speech recognition and music detection and it outperformed the other methods. Therefore, we used CLDNN architecture for music emotion recognition and better results were achieved than the other methods.

CLDNN architecture consists of two basic parts. The first part is the Convolutional Neural Network for feature extraction, while the other part is the classifier consisting of Long Short-Term Memory with Deep Neural Networks. MFCCs and log-mel filterbank energies are the most widely utilized features for speech recognition since these features convey the most relevant information. Same could also be true for music. Therefore, in this study, we proposed MFCCs and log-mel filterbank energies are fed to 1D CNN as input to extract new features. When new features obtained using MFCC and log mel filterbank energies are added to standard features better results are achieved.

In this thesis, the LDNN architecture was compared with other classifiers and better performances were achieved than other classification methods. The results show that the LDNN classifier yields 1.61, 1.61, 2.42 and 3.23 points improvements for native listeners and 5.65, 0.81, 4.84, and 6.45 points improvements for experts in music emotion recognition accuracies compared to that of K nearest neighbors (k-NN), Sequential Minimal Optimization (SMO), Naïve Bayes and Random Forest (RF) classifiers, respectively. Additionally, the overall accuracy of 99.19% was obtained using the proposed system with 10 fold cross-validation. In addition when new features are added to the standard features, 6.45 and 5.65 points improvements are achieved for native listeners and experts, respectively. As a result, the experimental results showed the quality of the database created, the effect of new features on the performance and the proposed classifier showed better results than other classification methods.

In future, the database size will be increased to include samples from the low arousal positive valence (LAPV) quadrant. In addition, the system will be evaluated with extra databases and the cross database performance of the approach will be explored. The approaches developed and presented in this thesis can also be applied to other classification problems. As an example, we consider to apply them for music genre classification.

REFERENCES

Aljanaki, A., 2016. Emotion in music: representation and computational modeling. Utrecht University, Ultrecht, Netherlands.

Aljanaki, A., Yang, Y. H. and Soleymani, M., 2017. Developing a benchmark for emotional analysis of music. PLoS ONE 12(3), 1–22, doi: 10.1371-journal.pone.0173392

Benito-Gorron, D. de., Lozano-Diez, A., Toledano, D. T. and Gonzalez-Rodriguez, J., 2019. Exploring convolutional, recurrent, and hybrid deep neural networks for speech and music detection in a large audio dataset. Eurasip Journal on Audio, Speech, and Music Processing, (1), 1–18, doi: 10.1186/s13636-019-0152-1.

Bottou, L., 1991. Stochastic gradient learning in neural networks. Proceedings of Neuro-Nimes, 91 (8).

Breiman, L., 2001. Random Forests. Machine Learning, 45, 1, 5–32, doi: 10.1023-A:1010933404324.

Cabrera, D., 1999. PsySound: A Computer program for psychoacoustical analysis. Australian Acoustical Society Conference, 47-54, doi: 10.1002/asi.

Chen, Y. A., Yang, Y. H., Wang, J. C. and Chen, H., 2015. The AMG1608 dataset for music emotion recognition. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 693-697, doi: 10.1109/ICASSP.2015.7178058.

Cheveigne, A. and Kawahara, H., 2002. YIN, A fundamental frequency estimator for speech and music. The Journal of the Acoustical Society of America 2002, 111 (1): 1917-30, doi: 10.1121/1.1458024.

Chollet, F. and et al., 2015. Keras. GitHub. Retrieved from https://github.com/fchollet/keras.

Dai, W., Dai, C., Qu, S., Li, J. and Das, S., 2017. Very deep convolutional neural networks for raw waveforms. In ICASSP. IEEE, 421–425.

Delbouys, R., Hennequin, R., Piccoli, F., Royo-Letelier, J. and Moussallam, M., 2018. Music mood detection based on audio and lyrics with deep neural net. CoRR, abs/1809.07276.

Dieleman, S. and Schrauwen B, 2014. End-to end learning for music audio. In ICASSP. IEEE, 6964–6968.

Edwards, A. L., 1976. The Correlation Coefficient. Ch. 4 in An Introduction to Linear Regression and Correlation. San Francisco, CA: W. H. Freeman, 33-46, 1976.

Eerola, T. and Vuoskoski, J., 2011. A comparison of the discrete and dimensional models of emotion in music. Psychology of Music, 10.1177/0305735610362821.

Ekman, P., 2005. Basic Emotions. In Handbook of Cognition and Emotion, (3), 45-60, doi: 10.1002/0470013494.ch3.

Eyben, F. and Schuller, B., 2015. OpenSMILE – The Munich Versatile and fast open-source audio feature extractor. ACM SIGMultimedia Records, 6(4), 4–13, doi: 10.1145/2729095.2729097.

Feng, Y., Zhuang, Y. and Pan, Y., 2003. Popular music retrieval by detecting mood. in: SIGIR Forum (ACM Spec. Interes. Gr. Inf. Retrieval), 375–376.

Fleiss, J. L., 1971. Measuring nominal scale agreement among many raters. Psychological Bulletin, 76(5), 378–382.

Gemmeke, J. F. and et al., 2017. Audio set: An ontology and human-labeled dataset for audio events. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing Proceedings, 776-780, doi: 10.1109-ICASSP.2017.7952261.

Grekow, J., 2015. Audio features dedicated to the detection of four basic emotion. Computer Information Systems and Industrial Management: CISIM'2015: 14th IFIP TC8 International Conference, September 24-26, Warszawa, Poland.

Hall, M. and Smith, L., 1997. Feature subset selection: A correlation-based filter approach. Proceedings of the 4th International Conference on Neural Information Processing and Intelligent Information Systems, New Zealand, 855–858.

Hall, M., and et al. 2009. The WEKA data mining software: An update. SIGKDD Explorations, 11, 10-18, doi: 10.1145/1656274.1656278.

Han, B. J., Rho, S., Dannenberg, R. B. and Hwang, E., 2009. SMERS: Music emotion recognition using support vector regression. Proceedings of the 10th International Society for Music Information Retrieval Conference, 651-656.

Harte, C., Sandler, M. and Gasser, M., 2006. Detecting harmonic change in musical audio. Proceedings of the ACM International Multimedia Conference and Exhibition, 21-26, doi: 10.1145/1178723.1178727.

Hevner, K., 1936. Experimental studies of the elements of expression in music. The American Journal of Psychology, 48, 2: 246-268.

Hochreiter, S. and Schmidhuber, J., 1997. Long Short-Term Memory. Neural Computation 9, 8, 1735–1780. doi: 10.1162/neco.1997.9.8.1735.

Hu, X., Downie, J. S., Laurier, C., Bay, M. and Ehmann, A. F., 2008. The 2007 MIREX Audio Mood Classification Task: Lessons Learned. In Proceedings of International Society for Music Information Retrieval, 462–467.

Huang C, W. and Narayanan S, S., 2017. Characterizing types of convolution in deep convolutional recurrent neural networks for robust speech emotion recognition. Arxiv 2017; abs/1706.02901.

Kingma, D. P. and Ba, J., 2014. Adam: A method for stochastic optimization. International Conference on Learning Representations, abs/1412. 6980: 1-15.

Koch, G. G., 1982. Intraclass correlation coefficient. In Samuel Kotz and Norman L. Johnson (ed.). Encyclopedia of Statistical Sciences 4, New York: John Wiley & Sons. 213–217.

Krippendorff, K., 2007. Computing Krippendorff's alpha reliability. Departmental Papers (ASC), 43.

Landis, J. R. and Koch, G. G., 1977. The measurement of observer agreement for categorical data, in Biometrics. (33), 159–174.

Lang, S., Bravo-Marquez, F., Beckham, C., Hall, M. and Frank, E., 2019. WekaDeeplearning4j: A deep learning package for Weka based on Deeplearning4j. Knowledge-Based Systems, 178(April): 48-50. doi: 10.1016/j.knosys.2019.04.013

Lartillot, O. and Toiviainen, P., 2007. Mir in matlab (II): A toolbox for musical feature extraction from audio. Proceedings of the 8th International Conference on Music Information Retrieval, September 23-27, Vienna, Austria, 127–130.

Lemaire, Q., and Holzapfel, A., 2019. Temporal convolutional networks for speech and music detection in radio broadcast. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), 2019, pp. 229–236.

Li, T. and Ogihara, M., 2003. Detecting emotion in music. Proceedings of the International Symposium on Music Information Retrieval, 2003 (3), 239-240.

Lin, Y. C., Yang, Y. H. and Chen, H. H., 2011. Exploiting online music tags for music emotion classification. ACM Transactions on Multimedia Computing, Communications and Applications, 7 S(1), doi: 10.1145-2037676.2037683.

Liu, H., Fang, Y. and Huang, Q., 2019. Music emotion recognition using a variant of recurrent neural network. International Conference on Mathematics, Modeling, Simulation and Statistics Application, 164: 15-18, doi: 10.2991-mmssa-18.2019.4.

Liu, T., Han, L., Ma, L. and Guo, D., 2018. Audio-based deep music emotion recognition. AIP Conference Proceedings 1967, doi: 10.1063/1.5039095.

Liu, X., Chen, Q., Wu, X. and Liu, Y., 2017. CNN based music emotion classification. ARXiv 2017, abs/1704.05665.

Lu, L., Liu, D. and Zhang, H. J., 2006. Automatic mood detection and tracking of music audio signals. IEEE Transactions on Audio, Speech and Language Processing, 14 (1): 5-18, doi: 10.1109/TSA.2005.860344.

Maas, A. L. and et al., 2017. Building DNN acoustic models for large vocabulary speech recognition. Computer Speech and Language, 41: 195-213, doi: 10.1016/j.csl.2016.06.007.

Malheiro, R., Panda, R., Gomes, P. and Paiva, R. P., 2016. Bi-Modal Music Emotion Recognition: Novel Lyrical Features and Dataset. 9th International Workshop on Music and Machine Learning – MML'2016 – in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases – ECML/PKDD, Riva del Garda, Italy.

Malheiro, R., Panda, R., Gomes, P. and Paiva, R. P., 2018. Emotionally-relevant features for classification and regression of music lyrics. IEEE Transactions on Affective Computing, 9(2): 240-254, doi: 10.1109-TAFFC.2016.2598569

Mathieu, B., Essid, S., Fillon, T., Prado, J. and Richard, G., 2010. Yaafe, an easy to use and efficient audio feature extraction software. Proceedings of the 11th International Society for Music Information Retrieval Conference August 9-13, Utrecht, Netherlands, 441–446.

McKay, C., 2009. JAudio: Towards a standardized extensible audio music feature extraction system. Course Paper, McGill University, Canada.

Mo, S. and Niu, J., 2017. A novel method based on OMPGW method for feature extraction in automatic music mood classification. IEEE Transactions on Affective Computing, 3045 (c): doi: 10.1109/TAFFC.2017.2724515

Pampalk, E., Rauber, A. and Merkl, D., 2002. Content-based organization and visualization of music archives. Proceedings of the ACM International Multimedia Conference and Exhibition, 570-579, doi: 10.1145-641118.641121

Panda, R. and Paiva, R. P., 2012. Music emotion classification: Dataset acquisition and comparative analysis. Proc. of the 15th Int. Conference on Digital Audio Effects, September 17-21, York, UK.

Panda, R., 2010. Automatic Mood Tracking in Audio Music. MSc. Thesis. Universidade de Coimbra, Portugal.

Panda, R., Rocha, B. and Paiva, R. P., 2015. Music emotion recognition with standard and melodic audio features. Applied Artificial Intelligence, 29 (4): 313-334, doi: 10.1080/08839514.2015.1016389

Platt, J. 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. Microsoft Research Technical Report: MSRTR, 98-14.

Plutchik, R., 1980. A general psychoevolutionary theory of emotion. In R. Plutchik & H. Kellerman (Eds.), Emotion: Theory, research and experience, Theories of emotion, (1), 3–33, New York: Academic Press.

Rocha, B., Panda, R. and Paiva, R. P., 2013. Music emotion recognition: The importance of melodic features. 6th International Workshop on Music and Machine Learning in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, September, Prague, Czech Republic.

Russell, J. A., 1980. A circumplex model of affect. Journal of Personality and Social Psychology, 39, 1161–1178.

Sainath, T. N., Vinyals, O., Senior, A. and Sak, H., 2015. Convolutional, long short-term memory, fully connected deep neural networks. ICASSP, 4580-4584, doi: 10.1109/ICASSP.2015.7178838.

Sainath, T. N., Weiss, R. J., Senior, A., Wilson, K. W. and Vinyals, O., 2015. Learning the speech front-end with raw waveform CLDNNs. Proceedings of the Annual Conference of the International Speech Communication Association, 1-5.

Santamaria-Granados, L., Munoz-Organero, M., Ramirez-Gonzalez, G., Abdulhay, E. and Arunkumar, N., 2019. Using deep convolutional neural network for emotion detection on a physiological signal dataset (AMIGOS). IEEE Access, 57-67, doi: 10.1109/ACCESS.2018.2883213.

Sarkar, R., Choudhury, S., Dutta, S., Roy, A. and Saha, S. K., 2019. Recognition of emotion in music based on deep convolutional neural network. Multimedia Tools and Applications, 79:765–783, doi: 10.1007/s11042-019-08192-x.

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. CoRR 2014, abs/1409.1556.

Soleymani, M., Caro, M. N., Schmidt, E. M., Sha, C. Y. and Yang, Y. H., 2013. 1000 songs for emotional analysis of music. CrowdMM - Proceedings of the 2nd ACM International Workshop on Crowdsourcing for Multimedia; 1-6, doi:10.1145/2506364.2506365.

Song, Y., Dixon, S. and Pearce, M., 2012. Evaluation of musical features for emotion classification. Proceedings of the 13th International Society for Music Information Retrieval Conference, October, Porto, Portugal.

T., Cover. and P., Hart., 1967. Nearest neighbor pattern classification. In IEEE Transactions on Information Theory, 13(1), 21-27, doi: 10.1109-TIT.1967.1053964.

Thayer, R. E., 1989. The Biopsychology of Mood and Arousal. New York: Oxford University Press.

Tzanetakis, G. and Cook, P., 1999. MARSYAS: A Framework for audio analysis. Organised Sound, 4(3), 169-175.

Tzanetakis, G., 2002. Manipulation, analysis and retrieval systems for audio signals. PhD, Princeton University, Princeton, New Jersey, USA.

Van Rossum, G. and de Boer, J., 1991. Interactively Testing Remote Servers Using the Python Programming Language, CWI Quarterly, 4, 4, 283–303.

Willmott, C. and Matsuura, K., 2005. Advantages of the Mean Absolute Error over the Root Mean Square Error in Assessing Average Model Performance, Climate Research. 30. 79. 10.3354/cr030079.

Yang, Y. H. and Chen, H. H., 2011. Music emotion recognition. CRC Press, Boca Raton, Florida, USA.

Yang, Y. H., Su, Y. F., Lin, Y. C. and Chen, H. H., 2007. Music emotion recognition: The role of individuality. In Proceedings of the ACM International Workshop on Human-Centered Multimedia, 13-21.

Yang, Y. H., Su, Y. F., Lin, Y. C. and Chen, H. H., 2008. A regression approach to music emotion recognition. IEEE Transactions on Audio, Speech and Language Processing, 16 (2): 448-457, doi: 10.1109/TASL.2007.911513.

Zhang, F., Meng, H. and Li, M., 2016. Emotion extraction and recognition from music. 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, 1728-1733, doi: 10.1109-FSKD.2016.7603438.

CURRICULUM VITAE

Serhat HIZLISOY was born in Kayseri in 1987. He received his BSc. from Department of Computer Engineering of Erciyes University in 2010 and MSc. from Department of Computer Engineering Çukurova University in 2015, respectively. He has been working as a research assistant in Department of Computer Engineering of Çukurova University since 2012.