ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES

AUTOMATIC CLASSIFICATION OF ETHIOPIAN TRADITIONAL MUSIC USING AUDIO-VISUAL FEATURES AND DEEP LEARNING

Selam Mulugeta Alemseged

A Thesis Submitted to the Department of Computer Science in Partial Fulfilment of the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

June 2020

ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES

Selam Mulugeta Alemseged

Advisor: Fekade Getahun (PhD)

This is to clarify that the thesis prepared by *Selam Mulugeta Alemseged*, titled: *Automatic Classification of Ethiopian Traditional Music using Audio-Visual Features and Deep Learning* and submitted in partial fulfilment of the requirements for the Degree of Masters of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:	_____	_____	_____
Examiner:	_____	_____	_____
Examiner:	_____	_____	_____

Abstract

Music bridges the gap between linguistic and cultural gap and helps connect people. Ethiopia is a country with more than 80 tribes each having their own unique musical sound and style of dance. Distinguishing one from another is not an easy task especially in the era of streaming where lots of music are recorder and released each day through the Internet.

Machine learning and recently deep learning is a subfield of machine learning that came to tackle the problem of automating tedious classification tasks previously done by programmers manually crafting the classification rules. Deep learning algorithms automatically learn the classification rules by just looking at the data.

In this work, we address the automatic classification of Ethiopian traditional music to their respective locality using audio-visual features. To achieve that we use a deep neural network architecture composed of both convolutional neural network (CNN) and recurrent neural network (RNN). This architecture has an audio feature extracting component, that is composed of a parallel deep CNN and RNN which takes mel-spectrogram of an audio signal as an input and a video feature extracting component. The video feature extracting component uses transfer learning to extract visual information from a pre-trained network (VGG-16) then passes these features to a Long Short-Term Memory (LSTM) recurrent network so that sequential information will be extracted. Features from both modules will then be merged and the class of the music video will be predicted.

We did an experiment to know the performance of the proposed system. We collected music data that represent Ethiopian traditional music from Internet-based music archive such as YouTube and personal music collections. After passing the collected data through a pre-processing step, we trained the proposed system, which uses both audio-visual feature and a system that only uses visual feature or audio feature. The performance of the video data only classifier was 78% while the audio data only classifier was 85% and by adding audio feature to the video data only classifier we were able to increase the accuracy of the proposed system by 7 units making its performance 85%.

Key Words: Deep learning, CNN, RNN, Transfer Learning, Music Information Retrieval, Music Processing, Dance Recognition

Dedication

I dedicated this work to my parents. You gave me all the support I needed throughout the research and more. Dedicating this work is the least I can do and you deserve more.

Acknowledgments

Doing this research was the most challenging task I have to deal with and It would not be complete without those people who helped me throughout the whole time.

First of all, I would like to thank my advisor Fekade Getahun (PhD). He helped me from the initiation of the idea till the end of the research. Next, I would like to thank Addis Ababa University for sponsoring my study for Masters degree.

Finally, I want to mention the names of the people who helped me through the data collection, information gathering and classification step. Without their help and contribution, this research would have not reached to its final stage: Meron Sheferaw, Milki Feyissa, Samson Hailu, Pertos Adanew, Yeselam Araya, Amanuel Negash, Kibrom Haftu, Mulugeta Kejela, Addis Yehasab, Birhanu Tarekegn, Rekik Kassahun, Anduamlak Birhanu, Kiros Arefayne and Gebremichael Kidu. All of you were behind the success of this research.

Table of Contents

List of Figures

List of Tables

Acronyms

ANN	Artificial Neural Network
AI	Artificial Intelligence
AUC P/R	Area Under the Precision Recall Curve
AUC-ROC	Area Under the Receiver Operating Characteristic Curve
CD	Compact Disc
CNN/ConvNet	Convolutional Neural Network
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
DWT	Discrete Wavelet Transform
EBC	Ethiopian Broadcast Corporate
ELU	Exponential Linear Unit
ETMC	Ethiopian Music Classifier
FBC	Fana Broadcast Corporate
FN	False Negative
FP	False Positive
GPU	Graphical Processing Unit
GRU	Gated Recurrent Unit
HOF	Histogram of Optical Flow
HOG	Histogram of Oriented Gradients
IndRNN	Independent Recurrent Network
ISA	Independent Space Analysis
KNN	K Nearest Neighbor
LBP	Local Binary Patterns
LSTM	Long Short-Term Memory
MFCCs	Mel-frequency Cepstral Coefficient
MP3	Motion Picture Expert Group Audio Layer-3
MSE	Mean Squared Error
NB	Naïve Bayes
ReLu	Rectified Linear Unit
RGB	Red Green Blue

RMS	Root Mean Squared
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SIFT	Scale Invariant Fourier Transform
STFT	Short-Time Fourier Transform
STIP	Space-Time Interest Point
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

Chapter 1: Introduction

1.1 Background

1.1.1 History of Music, Its Format and the Current Trend

Music is a highly valued feature of all known living cultures that shows many aspects of daily life and it is very ancient [1]. It is not known when humans first made music or what inspires them to do so [2]. However, the oldest known musical instruments appear in archeological record from 40,000 years ago [1]. Some say that the human voice was the first instrument. But most cultures have developed other distinctive ways of creating musical sound, from something as simple as two sticks struck together to the most complex pipe organ [3].

Sounds is made up of complex harmonics or overtones and the harmonic series is a mathematical reality, a physical truth or law of the universe. Pythagoras (570-495 BC) had described the mathematical relationship between the length of stretched string and the period of its vibration when plucked [2] . Aristotle (384 – 322 BC) understood that sound is a particular movement of air [4]. Chrysippus (280 – 207 BC), Romans Virtuvius (1st century BC) and Boethius (480-524 AC), speculated that sound is a wave phenomenon [2]. Galileo Galilei (1564 - 1642), originator of the study of modern acoustic, identified the relationship between the frequency and pitch of a sound [2]. Although it would not be applied for music for more than a century and half, French military scientist Jean-Baptiste Joseph Fourier (1768 - 1830) gave a lecture in 1807 describing a method to approximate any signal through a combination of trigonometric function that led to several mathematical processes such as Fourier Series, Analysis, Transform and Synthesis [2]. But it was not until Thomas Alva Edison invented the Phonograph in 1877 the preservation of quality of music was possible [2].

In post-world war II era, the most popular medium in recording music production was polyvinyl chloride, also known as vinly [5]. The late sixties brought about the significant popularity of the cassette as a means of record production. While storing the same amount of information as vinly, a blank cassette was sold for half the price. Moreover, with the introduction of the Sony Walkman in 1979 and the increase in sound quality of the cassette, it quickly became the most widely used music listening format in 1983 [6]. The late 1980s introduced a new format called Compact Disc (CD) that revolutionized the music industry

to the next level. Again, the CD was replaced by the technology of the digital era and CD along with all the physical media continued to decline by a compressed audio file format called MP3 (Motion Picture Experts Group Layer-3) [6]. The late 1990s were a revolutionary period for the music industry as it brought about the relevance of music in its digital form. It was from a combination of the Internet and digital audio recording that the MP3 was born. Because of their availability, consumers were able to download music immediately from their computers. Following this iTunes [7] was created in 2003 and this legal purchase of music allowed MP3 to quickly grow into one of the mainstream formats of the music industry in 2007 [6].

While it seemed that the MP3 was as attractive to the average consumer as music could legally be, the mid-2000s introduced an interesting concept of music consumption. With Internet speeds and technological familiarity increasing rapidly, developers saw the opportunity to deliver music in yet another form, i.e. Streaming of music. Similar to its digital counterpart, streaming allowed consumers to listen to music without having to download the songs to a computer or device. The music was simply delivered to consumers as a continuous stream of data. By early 2010s, streaming became the fastest growing music format in the recorded music industry [6]. Spotify [8], Tidal [9], Apple Music [10], YouTube Music [11], Pandora [12] and Amazon Music [13] are some of the service providers of music streaming.

With this new development of digital music, the dynamics of the music industry have been affected dramatically both in terms of record sales and the way music listeners are consuming music [6]. A study released by Recording Industry Association of America (RIAA), Streaming grew 13% in 2019, from $9.8 billion to $11.1 billion in retail value, accounting for 79% of the whole music industry revenue [14]. Physical sales now account for just 10% of the market place. But some of the biggest shrinks continue to be in digital downloads [14].

When it comes to the case of Ethiopia, even though cassettes have been vanished out from the market, many Musicians still release their album through CDs and all the music archives found in the oldest broadcast station such as Ethiopian Broadcast Corporate (EBC), have been digitized and stored in their digital format. Moreover, it has become so common for Ethiopian artists to release music videos through YouTube and they were able to get millions of views. For example, Zebiba Girma, Ephrem Amare, Selamawit Yohannes, Rahel Getu, Wendi Mac are among the many artists with millions of views on

YouTube. From this we can infer that Ethiopian songs are becoming popular with many people watching and listening their music using streaming service.

This revolution in the music distribution and storage brought by digital technology has fueled tremendous interest in and attention to the way information is retrieved to this kind of content. It resulted in the rapid growth of digitally available music data that requires novel technologies to allow users to browse personal collections or discover new music on the Internet.

1.1.2 Importance of Music Industry and the Need for Classification

Music helps in connecting people by bridging the linguistic and cultural divides. It is a representation of once identity and expression. The music industry plays an important role to one country when viewed from two angels. First, through music the culture of one country can be represented. In a traditional music, things such as how a society lives, dresses, beautifies, dances and perform day to day activities is show. Secondly, music helps job creation, economic growth, tourism development and artistic growth [15]. But this will not be possible if the recorded music is not properly delivered to the audience. As the produced and stored digital music is increasing each day, delivering the right kind of music to the right type of audience is important. Moreover, searching of any type of music should be possible by all mean including text based and content-based music retrieval.

1.1.3 Classification Techniques

Classification is a supervised way of machine learning where objects are assigned certain categories (or classes) based on the feature information. Classification techniques can be divided into two broad areas: statistical and structural (or nonmetric) techniques with a third area that borrows from both, sometimes called cognitive methods [16]. Bayesian Networks could be an example of statistical approach whereas decision tree and rule-based classifiers are examples of structural approach. Neural Networks and Support Vector Machines (SVM) are grouped in the cognitive approach category [16].

Music classification have been investigated to solve many problems in areas such as automatic music tagging, music recommendation and music information retrieval. Automatic music classification includes classifying music based on genre, mood or instrument.

There are many classification algorithms used for music classification where most of them fall under the traditional machine learning by using algorithms such as SVM, Artificial Neural Network (ANN) or Naïve Bayes (NB). But more recently, the traditional machine learning algorithms are being replaced by the use of deep learning, which is a sub-filed of machine learning that uses neural networks as its building block. Some of the researches under this music classification only consider the audio data while some investigate only the dance performed in the music.

1.1.4 Ethiopian Traditional Music

Ethiopia, an East African country, has over 80 ethnic groups. It is a country that has rich and diversified cultural heritage. As a consequence of the co-existence of different people in the country, "a living dance museum" has come into existence [17]. Each ethnic group has its own musical style that represents its own culture. As a result, it is quite difficult to distinguish which song belongs to which tribe even for a person born and raised in Ethiopia.

Ethiopian folk music and dance has existed over 3,000 years [18, 19]and because of the trade relationship the country had with near and far countries, the music has different style compared to that of other African countries [18]. In addition, each nation and nationalities have their own distinct music style [20]. However, most of them use a unique modal system that is pentatonic with long intervals between some nots [18, 20, 21]. Ethiopian music has a very steady tempo [21]. There is a lot of vocal music, accompanied with strong melodies and powerful rhythm [21].

The dance style in Ethiopia is extraordinarily diverse and it is related to the culture of peoples and not of the multiplicity of the dance stock belonging to an individual people, therefore it is horizontally grouped i.e. in space [17]. A type of dance could be performed in group, such as *Eskista* in Amhara and *Tilhit* in Tigray. A dance could also be performed only by a specific gender type male or female. Furthermore, there are dance types that are performed by couples, for example, *Shagoye* in eastern Oromia and *Awris* in Tigray [17].

Dance is integrated part of Ethiopians [19] and unlike the western world, Ethiopian traditional music and dancing are strongly connected and one is not separable form the other [22]. Each tribe has a unique dance step and rhythm [19]. To mention some, the dance of the Tigray region is characterized by two-beat drum rhymes and accordingly the dancers move in circles. In addition, it has different type of steps like neck motion,

rhythmical shoulder movement and so on. The dance of Amara people which is called *Eskista* has a unique movement of neck, shoulder and the chest though it has serval variation in motion and steps based on locality like Gondar, Gojjam and Wollo. In Somali, men dance to drum rhythm with masculine and elegant steps while women dance spreading their long dress like a butterfly. In Oromia, the dance differs according to the locality: Shoa Oromo, Harrer Oromo and so on. The dance of women of the Shoa Oromo is very popular by its fast and sharp neck movement while the men use stick in the dance. The dance of the Gurage people is fast tempo dance. Both men and women keep steps first without rest. In Wolayta, the people dance with distinctive movement of waist. The dance of the Gambella people involves the swirl of the waist. The women dance using their body while at the same time playing an instrument called fringe [19].

The advancement in digital technology has made the generation and storage of digital music data possible. Machine learning plays a significant role in looking for hidden patter in the data and classify them to proper classes. The fusion of audio and visual information is expected to increase classification accuracy [23]. For example, in speech recognition, using visual information such as lip movement along with the audio signal are used to disambiguate those acoustic signals that has similar sound such as the consonant /b/ and /d/ to increase the accuracy [23]. The fusion of audio visual information not only plays an important role in speech recognition but also in areas like person identification and event detection and recognition [23].

1.2 Motivation

Few years ago, most of the digital content was textual but it has now expanded to include audio, image, video and other types of multimedia documents. This is true for the music domain as well where listeners enjoy the unlimited access to huge music collections. Such huge amount of music data needs a retrieval strategy that allows users to explore the large music collection in a convenient and enjoyable way.

Text-based retrieval systems are very powerful. However, they require the music data to be rich with a significant amount of meta data which is not available most of the time. Furthermore, not all retrieval can be handled by purely text-based approach. To handle the scenarios that requires music to be retrieved based on its content, it requires content-based retrieval system that only makes use of the raw music data, i.e. audio and video, rather than relying on manually generated metadata.

Considering the web as huge collection of multimedia data, users listen and watch video clip. However, it is not easy to spell the specific origin/locality of the music based on the language used alone. Traditional Ethiopian music has distinct features associated to the modal system, rhythm, tempo etc. and the dance is quite distinct on the bases of the focus given to the part of the body. If one is interested to study the intangible cultural heritage of a society at large or a locality in particular, one of the starting point shall be studying the music.

When classifying an Ethiopian traditional music, depending on the language to know the origin might not always give us the right information because of many reasons: different style of music sung using the same langue, a song being sung by a different language form the origin of the song and use of two or more language to sing the song. In addition, the diversified nature of the type of songs sung in the country makes it hard to distinguish one type from another even for a person born and raised in Ethiopia. Therefore, trying to classify origin of a song using only language leads to the wrong conclusion and if knowing the exact origin of the song is needed we have to extract features related to the song and dance movements.

A survey made to two big radio and TV station (Ethiopian Broadcast Corporate – EBC and Fana Broadcast Corporate - FBC) revealed that even though they have huge Ethiopian music archive, the music collection is not categorized based on Ethiopian traditional music genre that made it very hard for a person in need of retrieving music data using that information. Even though FBC tried to categorized some of the songs, the tagging is made using manually leading to tedious and inaccurate categorization.

1.3 Statement of the Problem

In Ethiopia there are many types of music along with their own way of dance. Knowing the origin of a music either by listening the song or observing the dance movement might be difficult even for a person born and raised in Ethiopia. This is due to the diversified nature of Ethiopian traditional music that arises from the existence of over 80 ethnic group and the similarity between songs. For example, if we consider dance and music style in Amhara region, there is a slight difference between the song style of Gondar, Gojjam and Wollo people. This difference might be hard to distinguish for a person from outside that region.

There have been several researches done on music categorization based on genre [24, 25], mood or emotion [26, 27, 28] and the study of human body gesture for a specific purpose such as sign language recognition [29, 30], human computer interaction [31] and suspicious action detection [32].

Moreover, many researches are conducted on using video processing approach for classification and recognition of dance movements of other countries such as India [33] and Greek [34]. However, none of them integrated audio features of the video.

Even though there have been some researches done on the classification of Ethiopian traditional dance using frames taken from video clips such as the work done by Andargie Mekonnen *et al.* [35, 36] and Fitsum Tamene [37], their focus was mainly on regions found in the country, such as Oromia, Tigray, and not the specific locality in the identified region.

For example, in all works we see Oromia as one category but the dance and music style of the Oromo people found in different location is not the same and considering it as one category is wrong. The other wrong classification is in Andargie Mekonnen *et al.* work. The authors consider eskista as one category which should further be classifier into Gondar, Gojjam and Wollo. In Fitsum Tamene's classification, the one that was considered as Eskista was only the dance style of the Gondar people. In addition to that, none of them integrate the audio of the video when extracting features or there was no music processing in their research. In other words, their research was only on video processing. But the research that we are proposing here is based on both audio processing (music processing) and video processing making this research different from the others and we believe the fusion of both features can classify an Ethiopian music further down the hierarchy with a better accuracy.

Fistum Tamene [37] takes features related to the local shape of the dancers related to exterior body parts such as head, limb and joint in the video and we believe for the type of dance found in Ethiopian traditional music, if the feature related to the movement or motion of the dance performers are extracted rather than shape of the dancers, the accuracy could increase because the traditional Ethiopian dances are distinguishable by the focus given to the movement of particular body part and not the shape they produce. Extracting shape features from the dance movements for recognition of dance movement such as ballet dance could be the best method since the pose the dancers make could distinguish one dance pose from the other. As contrast to ballet dance, the Ethiopian traditional dance

is full of movement of particular body part. Therefore, we believe extracting features related to local movement of the dancers could give greater accuracy.

Although the work of Andargie Mekonnen *et al.* [35, 36] extracted features related to the motion of the performers using the magnitude of optical flow, they did not include the scale and direction of the optical flow that could increase the classification accuracy. Moreover, their classification is wrong as explained above and their researches did not consider audio feature of the video which can give a great clue to the origin of the music.

More recently, human action recognition problems are being investigated using a deep learning architecture [38, 39, 40, 41]. But their architecture was made to be trained on a powerful computer with large memory, high processing power and GPU support which we do not have. In addition, some use action bank features [41] and some use skeleton-based dataset [40] which we do not have for the Ethiopian traditional music data. Finally, as the problem they were investigating was action recognition, all of them did not incorporate audio features of the video.

Therefore, the problem that is investigated in this thesis is going to be the classification of traditional Ethiopian music in hieratical style (i.e. region and specific locality in the region) using both audio and visual features. To the best of our knowledge this research is the first that combines both audio and video for the purpose of Ethiopian music classification.

This research shall answer the following questions:

- What techniques shall be used to extract features from the audio and video so that the classification of an Ethiopian music is possible?
- What learning algorithm should be used for the purpose of the classification?
- How to improve the accuracy of the music classification using features that are extracted from both the audio and video?

1.4 Objective

The General Objective of this research is to automatically classify Ethiopian traditional music using audio-visual features and deep learning.

The specific objectives are:

- Study both the Ethiopian song and the dance movements.
- Prepare music corpus that represent traditional Ethiopian music style.

- Select appropriate deep learning algorithm that can construct a system to classify traditional Ethiopian music.
- Develop prototype using the selected deep learning algorithm.
- Test the performance of the system.
- Compare the performance of the system.

1.5 Methods

In order to achieve the objectives stated above the following methods were followed.

Literature Review

A detailed literature review on related topics such as Ethiopian traditional music, audio and video signal processing, machine learning and deep learning was conducted.

Corpus Preparation

The traditional Ethiopian music videos were collected from Internet based music archives such as YouTube. In addition, personal music collection was used.

Design and Development of Ethiopian Traditional Music Classifier

Deep neural network architecture that has both Convolutional and Recurrent layer was developed.

Implementation of the Ethiopian Traditional Classifier

The model was implemented using python programming language and Keras deep learning framework [42] that uses TenserFlow as a backend.

Experimentation

An experimentation was conducted to know the performance of the developed system.

1.6 Scope and Limitations

The scope of the research is limited to classification of only Ethiopian traditional music. It will not consider those songs that are sung using one of the native languages of Ethiopia but are not considered as being traditional, i.e. the modern Ethiopian songs. In addition, songs related to religion are also not included.

Moreover, the classification will not consider any language feature and language model. In other words, it will not put into consideration the language of the song and the terms

that are indicated or used in the song. For example, the repeated utterance of words entailing language such as *Oromia* may indicate that the origin of the music is Oromia region but the proposed research is only limited to features related to the audio and music features and will not do any kind of speech recognition or language processing. Furthermore, any kind of textual information either from the audio or video will not be considered.

The other thing that should be noted in the proposed work is that all music types of every ethnic group is not included. The ones that are selected in this research are those that are easier to get abundance of data and information on them. In addition, music representations from different location but similar tones or languages were included to evaluate how the classification model would perform in those kinds of scenarios.

Factors that could limit the result of the research include limitation of time or unavailability of some of the data in abundance and, processing power and memory of the machines we use to train our models. The time could limit us on the amount of data that can be collected and preprocessed since we can only collect data using the available time we have to do the research. We have found out that the most time-consuming task in doing this research was the collection and preprocessing of the data. In addition, processing power of the machines we use and availability of memory we have will determine the amount of data that can be processed or trained by and complexity of the model leading to not attaining the highest accuracy possible.

1.7 Applications of Results

After the finalization of the research it could be applicable in the following areas:

- **Content Base Music Information Retrieval**: If we want to do a search engine that can retrieve an Ethiopian traditional music by using the content found in the audio and video and not only by a textual meta-data tied to the music, this research can be a foundation.
- **Music Recommendation**: music recommendation systems can benefit from this research by studying the category preference of a person listening to a music and recommend similar songs in that category for future times.
- **Automatic Music Tagging**: Music tagging such as by artist, year, genre, category and so on is done in our country manually. People need to insert the tag of a music

and this research will help in the automatic tagging of the traditional music collection to their respective categories.

1.8 Organization of the Thesis

The organization of the thesis is as follows. In Chapter 2 we present background information regarding the area of study such as, Ethiopian traditional music, Digital Signal Processing (DSP), audio signal, image/video signal, machine learning and deep learning. Chapter 3 is about previous researches done on the area of music genre classification and dance or human action recognition. We review researches that uses both traditional machine learning and deep learning approaches. Chapter 4 is about the architecture proposed to solve the problem. In Chapter 5, we discuss about data collection, tools used and performance of the system. Finally, in Chapter 6, we conclude our research and present future work.

Chapter 2: Literature Review

2.1 Introduction

This Chapter is dedicated to give a general background regarding the area of study. The area of study of this thesis includes Ethiopian traditional music, which is discussed in Section 2.2, digital signal processing particularly about audio and video, which is discussed in Section 2.3, 2.4 and 2.5 and finally, deep learning that is discussed in Section 2.6.

2.2 Ethiopian Traditional Music

Ethiopia is a country found in East Africa with a population estimated around 109 million (in 2018) [43] making it the second most populous nation in Africa. It is the land of many nation, nationalities and people. It is believed that more than 80 ethnic group exists in the country. Ethiopia has a very rich and diverse musical sound. The various tribes and ethnic groups of Ethiopia have their own distinct music, culture and tradition.

Ezra [18] classified the different ethnic groups into three regions: North and Central, Eastern and South Western areas of the country. In the northern and central region, even though the music has rhythmic variation, the pentatonic scale is widely used. This could be listened in the music of Amhara, Tigray, Oromo and Gurage songs. The second, which is the Eastern, includes Somali, Harer and Kotu. As a result of the earlier contact with the Arab world especially with the Turk, the music in this region has the influence of Islam and the Arab culture. In this region, the pentatonic scale is also the popular one. The last one, which is the south western uses a different scale than the previous two regions. The type of scales used in this area start from two-tone scale up to the complicated type of diatonic scale and its rhythmic structure is rich and complicated and have homophonic factor. This can easily be listen from the Dokko songs or Derashe music. Some scholars who did a study on this area suggested that their scale to twelve-ton scale [18].

The culture of a society is not something unchangeable, it grows, changes, mixes up and also dies. Moreover, culture of once society or tribe could intermingle with another society. Therefore, it is common to see some similarities in musical sound and dance style of neighboring tribes in Ethiopia. Because of this mixing up of musical style, sometimes there is no clear-cutting point to determine which belongs to which society. For instance, songs

around the Wollo area could be an example. In Wollo a mixture of many tribes has lived there making their song very rich and diversified but also hard to put clear boundary.

Music is a way of expressing culture, physiological makeup of a society, love, sadness and so on. In most cultural (traditional) songs its common to hear songs that glorify their heroes or kings, tell their history, sing about their country, express their feeling to the loved once in a cultural way and so on. For example, in the song of the Gondar people, it is common to hear repetitive words such as Tewodros (Emperor of Ethiopia from 1855-1868), Kassa (another name for Emperor Tewodros). In Gojjam, Belay Zeleke (patriot who participated in the resistance against Italian during the occupation of Italy), Abay (the largest river in Ethiopia - Nile) and in songs of Wolayta, Kawo Tona (their last king before Menelik).

We have selected 10 traditional musical styles for this research. We selected those categories we though we can get an abundance of musical data and information and nothing else. The selected categories are: music of Tigray, Temben, Agew Wag Himra, Awi Agew, Gondar, Gojjam, Shoa Oromo, Harrer Oromo, Gurage and Wolayta.

2.2.1 Music of Tigray

a) Tigray

Tigray is the name of the northern most regions in Ethiopia. It is the homeland of Tigrayan, Irob and Kunama people. Tigray is bordered by Eritrea to the north, Sudan to the west, Amhara region to the south and Afar region to the east and south east. As a result, their culture, music and dance movement have similarities with some of the places bordering the region. The Tigrayans have smooth, circular dance routine which is concluded with shoulder and neck movement [44]. This circular dance is commonly known as *kuda* by the people of the land.

In their music, it is very common to hear traditional Ethiopian instruments such as the *masinqo*, a one-stringed violin like instrument that is played with a bow, *kerar*, a six-stringed lyre, played with fingers or a plectrum drum like structure and *kebero*, a drum like instrument, played using hand. In fact, they dance while playing the *kebero*. Music of Tigray people can be seen in Figure 1.

Figure 1 Picture Taken from Tigrigna Songs. Top: People Dancing in Circle (Kuda). Below: People Dancing while Playing Kebero

b) Temben

Temben is one of the woredas in Tigray region of Ethiopia but the people in Temben have different music and dance style from *kuda*. The dance style of Temben is called *Awris* and it has been described as imitating the movements of a hen and rooster. The dance includes two people, usually men and women. The woman crosses her arms and makes small movements while turning her back on the other person (usually a man). Meanwhile, the man, makes a huge movement with his arms and legs – basically showing off, this can be seen in Figure 2. After sometimes, either person can be replaced by a new participant, who pushes the old dancer away and takes turn to impress their new dance partner.

Figure 2 Awris, Traditional Dance of Temben People

2.2.2 Music of Agew

a) Agew Wag Himra

The Agew are ethnic group found both in Ethiopia and Eritrea. The Agew people that live in Seqota, Lasta and Lalibela share few similarities in their music with Tigray, *Ashenda/Shadey* being one example. *Ashenda/Shadey* is a festival celebrated in northern Ethiopia between 16-26 August by girls and young women's. In this paper the term Agew regers to the Agew Wag Himra people.

Their dance mostly includes strong movement of neck, shoulder and hand according to the rhythm of the song which is very strong. Figure 3 shows some of their dance movements.

Figure 3 Picture Taken from Traditional Music of Agew People

b) Awi Agew

Agew Awi is one of the 11 zones in the Amhara Region of Ethiopia. It is named from the Awi sub-group of the Agew people, some of whom live in this zone. Agew Awi zone is bordered on the west by Benshangul-Gumuz Region, on the north by Semien Gondar Zone and on the east by Mirab Gojjam.

The women usually dance holding an umbrella looking object made of bamboo while the men dance holding tool called *chira* that is used to remove fly. In their music, they often use an instrument called *tirumba*, an air(pipe) instrument that has a pleasant sound. Their dance can be seen from Figure 4.

Figure 4 Pictures Taken from Traditional Songs of Agew Awi People

2.2.3 Music of Amhara

a) Gonder

Gondar is a city and a separate woreda located in Semen Gondar zone of the Amhara region. The city previously served as the capital of both the Ethiopian Empire and the subsequent Begemder Province. The dance of Gondar people, along with other Amhara people is called *Eskista*. It involves the movement of neck, shoulder and chest.

The Gondar dancing style is characterized by leaping in to the air, wriggling the shoulder and the neck accordingly, as the first dancer perform his part in this manner the second dancer will join him so as to excel the first dancer. This action will continue for the next few minutes accompanied by women's singing and clapping of hands [37]. Figure 5 shows the traditional dance of the people.

Figure 5 Picture Showing Traditional Dance of Gondar People

b) Gojjam

Gojjam is in the north west part of the country located in Amhara region. Like the other Amhara people, their dance involves the movement of neck, shoulder and chest but has several variations of movements and steps. Their dance is well known by the shacking of the upper body or chest. Figure 6 shows their traditional dance.

Figure 6 Pictures Taken from Traditional Song of Gojjam People

2.2.4 Music of Oromo

a) Shoa Oromo

Shoa Oromos are among the variant Oromos located in the central part of Ethiopia. They have a very unique dance movement that involves the fast movement of the head by the women. In addition, they have a unique costume, the women wear leather-made wild two-piece costume decorated with shells while the men wear fur skin lion's mane on the head and use stick for dance. This can be seen from Figure 7. The song of Oromo which is sung during the dance performance is very different from the traditional African type as the emphasis is given more to string instruments than drum beats. The vocals are also quite unique and a flavor of Afro-Asian influence can be easily heard [45].

Figure 7 Pictures Taken from Traditional Songs of Shoa Oromo

b) Harrer Oromo

Harrer Oromos or Hararghe is located in eastern Ethiopia. The Oromos that live in this area have different dance and music style than the Shoa Oromo. Their famous dance is

known as *Shagoye* which is performed by men and women as a couple. In this dance, persons of different sex never form independent couples drawn from the opposite rows during the course of the dance, but the two sexes dance with each other as two impersonal collective units [17]. *Shagoye* is seen in Figure 8.

Figure 8 Shagoye - Traditional Dance of the Harrer Oromo People

2.2.5 Music of Gurage

Gurage is a zone in the Ethiopian Southern Nations, Nationalities and Peoples' region (SNNPR). This zone is named for the Gurage people, whose homeland lies in this zone.

The Gurage people dance style is characterized by foot motifs which goes with the whole-body movement. This dance is performed in two ways, the first part is short and slow motion and it lasts between approximately ten to twenty seconds. In this stage the dancer will accumulate his energy for the second and fast part of the dance. The second part of the dance might be performed repeatedly for three or four times with short intervals depending on the dancer's interest and energy [37]. Figure 9 shows how they dance.

Figure 9 Pictures Taken from Gurage People Songs

2.2.6 Music of Wolayta

Wolayta is one the ethnic groups found in southern Ethiopia. The Wolaytas are known for their patriotism, rich culture and modern music. Their dance has a distinct movement of the waist [37] Figure 10 and their music is playing a prominent role in national

entertainment in Ethiopia today. The unique and fast-paced Wolayta tunes have influenced several styles and rhythm as it continues to shape the identity of Ethiopian musical diversity.

Figure 10 Pictures Taken from Wolayta Songs

2.3 Digital Signal processing

Everything in the universe, be it natural or man-made can be represented using signal. Signal is a description of how one parameter is related to another parameter [46]. Signals could be continuous (analog) as occurred naturally or digital as occurred in digital devices such as computers. Computers can only store and process signal in digital form. Therefore, image, audio and video signals need to be converted to a digital form before they are stored and processed by computers.

Digital Signal Processing (DSP) is the mathematics, algorithms, and techniques used to manipulate signals after they have been converted into a digital form. This includes enhancement of visual images, recognition and generation of speech, compression of data for storage and transmission, etc. DSP is distinguished from other areas in computer science by the unique type of data it uses, that is, signals. These signals originate as sensory data from the real world such as visual images and sound waves, etc [46].

A signal is a description of how one parameter is related to another parameter or it can also be defined as how one parameter varies with another parameter [46]. It is a function that tells information about the state or behavior of a physical system. For example, a signal may describe the time-varying sound pressure at some place or the distribution of light on a screen representing an image. In addition, as in the case of video signal, it can also describe the sequence of images [47]. The most common type of signal in analog electronics is a voltage that varies with time or brightness that varies with distance in an image [46].

We can model signals using two mathematical notations: analog (continuous) signals and digital signals [47]. In continuous signals the parameters can assume a continuous range

of values [46] which generally leads to an infinite number of values. Since a computer can only store and process a finite number of values, one has to convert the waveform into some discrete representation by a process commonly referred to as digitization [47]. Passing this signal through an analog-to-digital converter forces each of the two parameters to be sampled and quantized. Signal formed from parameters that are quantized in this manner are said to be discrete signals or digitized signals. For the most part, continuous signals exist in nature, while the discrete signals exist inside computers [46].

A sinusoid is a periodic function *f* defined by:

$$f(t) := A\sin(2\pi(\omega t - \phi)),\ t \in \mathrm{R}. \quad (1)$$

Where A describes the amplitude, ω the frequency, t the time, and the parameter ϕ the phase.

Sinusoid has a compact description of a signal wave. It also has an explicit physical meaning with a perceptual correspondence. For example, in an audio signal, the amplitude corresponds to the loudness and the frequency to the pitch of the sinusoidal sound. Only the phase, which indicates the relative position of an oscillation within its cycle, does not have a direct perceptual correspondence [47].

The signals that are abundantly available in the environment (before being sensed) are naturally analog. By analog, we mean two things: that the signal exists on a continuous (space/time) domain, and that also takes values that comes from a continuum of possibilities [48]. However, computers only process digital image and video signals, which means that once the signal is sensed, it must be converted into a computer-readable digital format. By digital, we also mean two things: that the signal is defined on a discrete (space/time) domain, and that it takes value from a discrete set of possibilities [48].

The most common approach for digitizing of signals consists of two steps called sampling and quantization. Sampling refers to the process of reducing a continuous-time signal to a discrete-time signal, which is defined only on a discrete subset of the time axis. Quantization is the step that replaces the continuous range of possible amplitudes by a discrete range of possible values [47].

2.4 Audio Signal

The term audio is used to refer to the transmission, reception, or reproduction of sounds that lie within the limits of human hearing [47], i.e. 20 Hz and 20,000 Hz (20kHz).As

mentioned above a signal is a description of how one parameter varies with another parameter and sound/audio signal represents the amplitude of the air pressure over time [47].

A sound is generated by a vibrating object such as the vocal cords of a singer. These vibrations cause displacement and oscillations of air molecules, resulting in local regions of compression and refraction. As shown in Figure 11 below, The alternating pressure travels through the air as a wave, from its source to a listener or a microphone. At its destination, it can then be perceived as sound by the human or converted into an electrical signal by microphone.

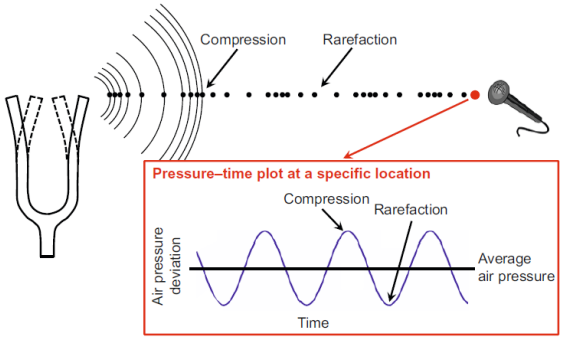


Figure 11 Vibrating Tuning Fork Resulting in Back and Forth Vibration of the Air Particles [47]

Graphically, the change in air pressure at a certain location can be represented by a pressure-time plot, also referred to as the waveform of the sound. The waveform shows the deviation of the air pressure from the average air pressure. The following diagram, Figure 12, shows the waveform representation of a recording.

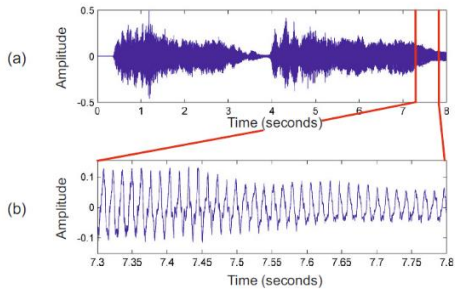


Figure 12 Pressure-Time Graph of a Sound Wave [47]

2.4.1 Signal Terminologies

The perception of a continuous sound, such as a note from a musical instrument, is often divided into three parts: pitch, loudness and timbre. Pitch is the frequency of the

fundamental component in the sound, that is, the frequency with which the waveform repeats itself. Loudness is a measure of sound wave intensity and Timber is more complicated, being determined by the harmonic content of the signal [46].

Frequency and Pitch

We have seen above that sound wave can be visually represented by a waveform. If the points of high and low air pressure repeat in an alternating and regular fashion, the resulting waveform is called periodic. In this case, the period of the wave is defined as the time required to complete a cycle. The frequency, measured in Herts (Hz), is the reciprocal of the period. The following diagram, Figure 13, shows a sinusoid, which is the simplest type of periodic waveform. The sinusoid is completely specified by its frequency, amplitude (the peak deviation of the sinusoid from its mean), and its phase (determining where in its cycle the sinusoid is at time zero) [47].

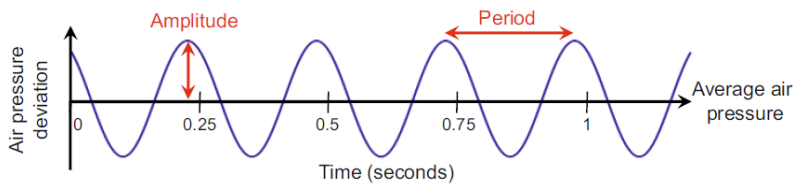


Figure 13 Waveform of Sinusoid with a Frequency of 4 Hz [47]

The sinusoid can be considered the prototype of an acoustic realization of a musical note. The notion of frequency is closely related to what determines the pitch of a sound. In general pitch is subjective attribute of sound. The higher the frequency of a sinusoidal wave, the higher its sound [47].

Dynamics, Intensity, and Loudness

Dynamics is a general term that is used to refer to the volume of a sound as well as to the musical symbols that indicate the volume. On the audio side, dynamics correlate with a perceptual property called loudness, by which sounds can be ordered on scale extending from quiet to loud. Similarly, to the relation between pith and frequency, loudness is a subjective measure which correlates to objective measure of sound intensity and sound power. However, loudness also depends on other sound characteristics such as duration or frequency [47].

Power is the rate at which energy is transferred, used, or transformed, whereas, sound power expresses how much energy per unit time is emitted by a sound source passing in

all direction through the air. The term sound intensity is then used to denote the sound power per unit area [47].

Timber

Timber or tone color is what allows a listener to distinguish the musical tone of a violin, guitar, or a trumpet even if the tone is played at the same pitch and with the same loudness. As with pitch and loudness, timber is a perceptual property of sound. However, timber is very hard to grasp, and because of its vagueness, it is often described in an indirect way: timber is the attribute whereby a listener can judge two sounds as dissimilar using any criterion other than pitch, loudness, and duration [47].

2.4.2 Music Processing

The Fourier Transform

Music signals are generally complex sound mixtures that consist of multitude of different sound components. Because of this complexity, the extraction of musically relevant information from a waveform constitutes a difficult problem. A first step in better understanding a given signal is to decompose it into building blocks that are more accessible for the subsequent processing steps. In the case that these building blocks consist of sinusoidal functions, such a process is also called Fourier analysis. Sinusoidal functions are special in the sense that they possess an explicit physical meaning in terms of frequency. As a consequence, the resulting decomposition unfolds the frequency spectrum of the signal. The Fourier transform converts a signal that depends on time into a representation that depends on frequency [47].

The pitch of a musical tone is closely related to its fundamental frequency, the frequency of the lowest partial of the sound. Therefore, we need to determine the frequency content, the main periodic oscillations of the signal.

The original signal and the Fourier transform contains the same amount of information. This information, however, is represented in different ways. While the signal displays the information across time, the Fourier transform displays the information across frequency. The signal tells us when certain notes are played in time but hides the information about frequencies. In contrast, the Fourier transform of music displays which notes (frequencies) are played but hides the information about when the notes are played [47].

Even though a signal and it's superpositions are different in nature, the resulting magnitude Fourier transform is more or less the same. This demonstrates the drawbacks of Fourier transform when analyzing signals with changing characteristics over time. Therefore, a short-time version of the Fourier transform, where time information is recovered at least to some degree is usually used.

Short-Time Fourier Transform (STFT)

The Fourier transform yields frequency information that is averaged over the entire time domain. However, the information on when these frequencies occur is hidden in the transform. To recover the hidden time information, Dennis Gabor introduced in the year 1946 the short-time Fourier Transform (STFT). Instead of considering the entire signal, the main idea of the STFT is to consider only a small section of the signal [47].

Spectrogram

The spectrogram reveals the frequency information of the played note over time. Figure 14 shows the spectrogram of a music played over time. The horizontal lines that are stacked on top of each other on equally spaced lines correspond to the partials, the integer multiples of the fundamental frequency of a note. The higher partials contain less and less of the signal's energy.

The composition of sound in terms of its partials can be visualized by a so-called spectrogram, which shows the intensity of the occurring frequencies over time [47].

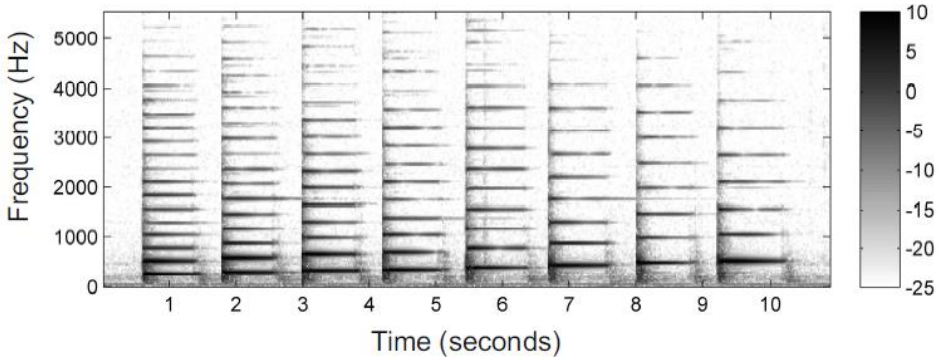


Figure 14 Spectrogram of a Music Signal [47]

Musical Dimensions – Feature representation

Since the sampled waveform of an audio signal is relatively uninformative by itself, the first step in music processing is to transform the given music recording into a suitable feature representation that correlates to the various musical aspects.

As a first representation, we consider chroma features. The normalized chroma vector describes the signal's local energy distribution over an analysis window (frame) across the twelve pitch classes of the equal-tempered scale. Capturing pitched content, a chroma-based feature sequence relates to harmonic and melodic properties of music recording.

Besides melody and harmonic, the instrumentation and timbral characteristics are of great importance for the human perception of music structure. In the context of timber-based structure analysis, one often uses mel-frequency cepstral coefficient (MFCCs), which were originally developed for automated speech recognition. Parametrizing the rough shape of the spectral envelop, MFCC-based features capture timbral properties of the signal.

At a third musical dimension, we consider properties that are related to beat, tempo, and rhythmic information. Tempogram encodes local tempo information.

Temporal and structural regularities are perhaps the most important incentives for people to get involved and to interact with music. It is the beat that drives music forward and provides the temporal framework of a piece of music. Intuitively, the beat corresponds to the pulse a human tap along when listening to music. The beat is often described as a sequence of perceived pulse positions, which are typically equally spaced in time and specified by two parameters: the phase and the period. The term tempo refers to the rate of the pulse and is given by the reciprocal of the beat period.

Finally, we want to mention that the extraction of onset, beat, and tempo information is of fundamental importance for the determination of higher-level musical structures such as rhythm and meter. Generally, the term rhythm is used to refer to a temporal patterning of event durations, which are specified by a regular succession of strong and weak stimuli.

2.5 Image/Video Signal

Images are signals that describe how a parameter varies over surface. They have their information encoded in the spatial domain, the image equivalent of the time domain. In other words, features in image are represented by edges, not sinusoids. This means that the spacing and number of pixels are determined by how small of features need to be seen, rather than by the formal constraints of the sampling theorem [46].

Images are signals with special characteristics. First, they are a measure of a parameter over space (distance), while most signals are a measure of a parameter over time, such as audio. Second, they contain a great deal of information. For example, more than 10 megabytes can be required to store one second of television video. This is more than a thousand times greater than for a similar length voice signal. Third, the final quality is often a subjective human evaluation, rather than an objective criterion. These special characteristics have made image processing a distinct subgroup within DSP [46].

Video is the dynamic form of static images. In other words, the video is composed of a series of static images in a certain order, and each image is called a 'frame' [49]. Video refers to pictorial (visual) information, including still images and time-varying images. A time-varying image is such that the spatial intensity pattern changes with time. Hence, a time-varying image is a spatio-temporal intensity pattern, denoted by $sc(x_1, x_2, t)$, where x_1 and x_2 are the spatial variables and t is a temporal variable. It is formed by projecting a time-varying three-dimensional spatial scene into the two-dimensional image plane. The temporal variations in the 3-D scene are usually due to the movement of objects in the scene. Thus, time-varying images reflect a projection of 3-D moving object into the 2-D image plane as a function of time. Digital video corresponds to a spatio-temporally sampled version of this time-varying image [50].

An important feature of digital images and videos is that they are multidimensional signals, meaning that they are functions of more than a single variable. In the classic study of digital signal processing, the signals are usually one-dimensional function of time. Images, however, are function of two, and perhaps three space dimensions, where digital video as a function includes a third (or fourth) time dimension as well. The dimension of a signal is the number of coordinates that are required to index a given point in the image [48].

People start the research in image processing simulating the human vision to see, to understand, and even to explain the real world using three techniques: image segmentation, image analysis and image understanding [49]. While image processing refers to the technique of performing a series of operations on an image to achieve some desired purposes, Image Analysis mainly aims to detect and measure the object of interest in the digital image, in order to create specific descriptions. It is a process from an image to values or symbols, which generates some non-image descriptions or representations by extracting useful data or information first. Its content can be divided into several parts, such as feature extraction, object description, target detection, scene matching and

recognition. They describe the characteristics and features of the target in the image. Similar to image analysis, video analysis is also a broad concept which includes the task of visual object tracking, human action analysis, abnormal behavior detection, and so on.

In real world environment, due to the changes in illumination, the target movement complexity, occlusion, color similarity between targets and background, and background clutter, it is difficult to design a robust algorithm for target detection and tracking.

2.5.1 Image and video scene segmentation

We are only interested in regions of inputs which are called target or foreground, and it generally corresponds to certain specific or unique properties regions in the image or video frame. On the other side, a region in which we are not interested is called background.

Segmentation means dividing an image or video frame into a number of specific and unique property parts or subsets according to the principle and extracting the target of interest for a higher level of analysis and understanding [49]. Therefore, it is the basis of feature extraction, recognition and tracking. There is neither a general method for image segmentation so far, nor an objective criterion for judging whether the segmentation is successful.

The purpose of segmentation is to isolate a meaningful entity from image or video sequence. These targets typically correspond to some specific, unique areas of image or video frames. The uniqueness here can be the grayscale value of pixel, object contour curve, color, texture, movement information, etc. Such uniqueness can be used to represent an object as the characteristics between regions of different objects. The target can correspond to a single region or multiple regions. To identify and analyze the targets, it is necessary to isolate and extract them, such that further identification and understanding can be carried out.

Image segmentation can be built on two basic concepts: similarity and discontinuity. The similarity of pixels means that pixels of the image in a certain region have some similar characteristics, such as pixel grayscale, or texture formed by the arrangement of pixels. The discontinuity refers to the discontinuity of some features of pixels, such as the mutation of grayscale value, the mutation of color and the mutation of texture structure.

Image segmentation is generally achieved by considering the image color, grayscale, edge, texture, and other spatial information. Currently, image segmentation algorithms can be divided into two categories: structural segmentation and non-structural segmentation.

Structural segmentation methods are based on the characteristics of the local area of image pixels, including threshold segmentation, region growing, edge detection, texture analysis, etc. These methods assure that the features of these areas are known in advance and they are obtained during processing.

Non-structural segmentation methods include statistical pattern recognition, neural network methods, and the methods using the prior knowledge of relationships between objects. For example, the snake method which uses active contour model to segment objects, is a framework in computer vision for delineating an object outline from a possibly noisy 2D image.

Because there is no temporal information in image segmentation, it cannot be used to get satisfactory segmentation results on video sequence. The efficiency of segmentation algorithm can be improved by considering the time correlation of video frames. Therefore, video segmentation jointly uses spatial and temporal information to achieve this goal.

2.5.2 Image and video feature description

Another step to image/video segmentation is to describe the characteristics of the scene with a series of symbols or rules and then identify, analyze and categorize the descriptions.

Image features refer to their original properties or attributes. It is one of the most basic attributes used to distinguish different images. Some are natural and can be perceived by the vision, such as the brightness of the region, edges, textures or color, etc. There are also some artificial (man-made) ones that need the be transformed or measured, such as transform spectrum, histogram, moment, etc.

Feature extraction is such a process of measuring the intrinsic, essential and important features or attributes of the object and quantizing the result or decomposing and symbolizing the object to form the feature vector or symbol string and relational map [49].

In general **Feature descriptors** refer to a series of symbols that are used for describing the characteristics of an image or video object. A good descriptor should be insensitive to the target's scale, rotation, translation, etc. Feature descriptors generally fall into two main groups: global and local.

A **global feature** is calculated from all the pixels of an image, and it describes the image as a whole. Commonly used features include color, texture, shape features, etc.

Compared to global features, **local features** describe the local regions in an image with better uniqueness, invariability and robustness and they have the better robustness to background clutter, local occlusion and illumination change. Local features may be points, edges, or blobs in an image, which have the advantages of describing the characteristics of pixels or colors in local regions. Due to its excellent performance, local features have attracted more and more research attention. In particular, some local features with strong robustness to illumination and occlusion have been proposed in recent years, such as: Moravec corner detector, Harris corner detector, Smallest univalue segment assimilating nucleus corner detector (SUSAN), Scale Invariant Feature Transform (SIFT), Difference of Gaussian (DOG) operator, Gradient Location and Orientation Histogram (GLOH), Speed Up Robust Features (SURF), Maximally Stable Extreme Regions (MSER), Local Binary Pattern (LBP).

2.5.3 Object Recognition in image/video

The recognition ability of human kind is more powerful. Even with dramatic scale changes, large displacement, and heavy occlusion, people can still identify the objects.

In computer vision, image recognition mainly refers to the task of recognizing objects in an image or video sequence. We employ some computational models to extract features from a two-dimensional image to form the digital description and then establish a classifier for classification and recognition.

The classifier can be divided into three categories: **Generative/Productive Model** (including probability density model), **Discriminative Model** (decision boundary learning model). Recently, the **Deep leaning-based models** have been widely applied in object recognition tasks, which can be viewed as another category.

Generative Model is also called productive model, which tries to estimate the joint probability distribution of training samples (observations) and their labels. Popular generative methods include: Gaussian Mixture Model (GMM), Naïve Bayes Model (NBM), Mixture of Multi-nominal Model (MMM), Mixture of Experts System (MES), Hidden Markov Models (HMM), Latent Dirichlet Allocation (LDA), Sigmoidal Belief Networks (SBN), Bayesian Networks (BN), Markov Random Fields (MRF) etc.

Discriminative Model is also called Conditional Model, or Conditional Probability Model. The objective of this model is to look for the optimal classification surface between different categories, which reflects the difference between heterogeneous data. The commonly used methods include Linear Discriminant Analysis, Logistic Regression, Artificial Neural Networks, Support Vector Machine, Nearest Neighbor, Boosting trees, Conditional Random Fields, etc.

Different from traditional object recognition pipeline of "feature extraction-classification", object recognition can be achieved in an "end-to-end" way through deep learning. Almost all object recognition tasks have been shined by **deep learning** nowadays, and the performance is greatly improved [49].

2.5.4 Scene Description and Understanding

Scene description and understanding is the high-level tasks of image understanding. The main objective is to automatically assign labels to the image scene via a set of semantic categories, in order to provide contextual information for other jobs like object recognition. It is the task of finding out some specific regions in an image based on the organization principle of visual perception, and then automatically labeling them based on a given set of semantic categories.

Scene classification provides an effective contextual semantic information for higher-level image understanding (e.g., object recognition).

2.6 Machine Learning and Deep learning

Artificial Intelligence (AI) is a field that was born in the 1950s having the idea of making computers to "think" like humans. The formal definition of the field is the effort to automate intellectual tasks normally performed by humans [51]. As such, AI is a general field that includes machine learning and deep learning but also includes other approaches that do not involve any learning. For instance, early chess programs only involve hardcoded rules crafted by programmers and did not qualify as machine learning. For a long time, it was believed by many experts that human-level artificial intelligence could be achieved by having programmers handcraft a sufficiently large set of rules for knowledge. This approach is known as symbolic AI. Although this approach was suitable to solve well-defined logical problems such as playing chess, it turned out to be difficult to craft rules for solving more complex problems, such as image classification, speech

recognition and language translation. As a result, a new approach arose to take symbolic AI's place: machine learning.

Machine learning arose as an answer to the following question: rather than programmers craft data-processing rules by hand, could a computer automatically learn these rules by looking at data? This question opens a door to a new programming paradigm. With machine learning, humans input data as well as the expected answer and the rules come as an output as opposed to classical programming (symbolic AI) where human input rules (a program) and data to be processed according to these rules and answer comes as an output as shown in Figure 15.

Machine learning models are all about finding appropriate representations for the input data. By representation, it means transforming the data that makes it more appropriate to the task at hand. This transformation of data has been the central problem for machine learning tasks, i.e. to learn meaningful representation of the input data at hand. Deep learning came to tackle this problem. Deep learning is a new take on learning representation from data that puts an emphasis on learning successive layers of increasingly meaningful representations [51].

Figure 15 Classical Programming and Machine Learning [51]

Machine learning is a subfield of computer science where machines learn to perform tasks for which they were not explicitly programmed. In short, machines observe a pattern and attempt to imitate it in some way that can be either direct (supervised) or indirect (unsupervised) [52].

2.6.1 Machine Learning

Recognizing objects and patters is easy for humans. Distinguishing the sound of a dog from a cat or apple form orange is easy for us. We are so good at this that we take our ability for granted. On the other hand, writing a computer program to do the same is difficult and we need computers to do the same, i.e., recognizing patterns. Pattern recognition is when a computer program is able to capture the patterns specific to the object in recognition and be able to identify it by analyzing sample examples [16].

Machine Learning algorithm is an algorithm that is able to learn from data. A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in T, as measured by *P*, improves with experience *E* [53]. If we want to automatically classify data to specific class then the classification process is the task. To evaluate the ability of a machine learning algorithm, we must design a qualitative measure of its performance. Usually, this performance measure *P* is specific to the task *T* being carried out by the system. E.g. for tasks such as classification, we often measure the accuracy of the model.

Machine learning algorithms can be broadly categorized as unsupervised or supervised by what kind of experience they are allowed to have during the learning process [54].To make a machine learning algorithm, we need to design an algorithm that will improve the weight *w* in a way that reduces MSE_{test} (Mean Squared Error) when the algorithm is allowed to gain experience by observing a training set $(X^{(train)}, y^{(train)})$ [54].

The ability to perform well on previously unobserved inputs is called generalization. The factor determining how well a machine learning algorithm will perform is its ability to make the training error small and make the gap between training and test error small. These two factors correspond to the two central challenges in machine learning: underfitting and overfitting [54].

Underfitting occurs when the model is not able to obtain a sufficiently low error value on the training set. Overfitting occurs when the gap between the training error and test error is too large. We can control whether a model is more likely to overfit or underfit by altering its capacity. Informally, a model's capacity is its ability to fit a wide variety of functions. Models with low capacity may struggle to fit the training set. Models with high capacity can overfit by memorizing properties of the training set that do not serve them well on the test set. One way to control the capacity of a learning algorithm is by choosing its hypothesis space, the set of functions that the learning algorithm is allowed to select as being the solution [54]. Machine learning algorithms will generally perform best when their capacity is appropriate for the true complexity of the task they need to perform and the amount of training data they are provided with.

2.6.2 Deep Learning

Deep learning has made enormous success in different areas, such as computer vision, speech recognition and natural language processing. Inspired by this success, many

researchers are moving from using traditional (shallow) machine learning algorithms to using deep learning to solve problems in music information retrieval and human activity recognition. Before the popularity of deep learning algorithms, audio and video classification problems were used to be investigated using the traditional (shallow) machine learning algorithms such as SVM, ANN, NB. When designing a model using those algorithms, the performance of the model heavily depends on the feature engineering task rather than on the classifier itself [55]. However, deep neural network automatically learns advanced feature layer by layer automatically and effectively instead of time consuming feature engineering [56] which led to the success of many problems in different areas. The idea of training a data using neural network has been there for a long time but it has become popular due to several reasons such as: availability of large amount of data and advancement of hardware technologies such as Graphical Processing Unit (GPU).

Deep learning is a subset of machine learning, which is a field dedicated to the study and development of machines that can learn (sometimes with the goal of eventually attaining genera artificial intelligence). In industry, deep learning is used to solve practical tasks in a variety of fields such as computer vision (image), natural language processing (text), and automatic speech recognition (audio). In short, deep learning is a subset of methods in the machine learning toolbox, primarily using artificial neural networks, which are a class of algorithm loosely inspired by the human brain [52]. Furthermore, Deep Learning is a mathematical framework for learning representations from data [51].

Representation at its core is a different way to look at data, i.e. to represent or encode data. For example, a color image can be encoded in the RGB (Red Green Blue) format or in the HSV (Hue Saturation Value): these are two different representations of the same data. Some tasks that maybe difficult with one representation can become easy with another. Machine-learning models are all about finding appropriate representations for their input data – transformations of the data that make it more amendable to the task at hand, such as a classification task [51].

Modern deep learning involves tens or even hundreds of successive layers of representations and they are all learned automatically from exposure to training data. Meanwhile, other approaches to machine learning tend to focus on learning only one or two layers of representations of the data. Hence, they are sometimes called shallow learning.

In deep learning, these layered representations are (almost always) learned via models called neural networks. The term neural network is referred to neurobiology, but although some of the central concepts in deep learning were developed in part by drawing inspiration from our understanding of the brain, deep-learning models are not models of the brain [51]. A simple definition of a neural network is one or more weigh that is multiplied by an input data to make prediction [52]. Steps involved in deep learning are Predict, Compare and Learn.

Prediction is what the neural network will tell you, given an input data, such as given temperature, how likely it is going to rain. But the prediction is not always right. Therefore, it will compare with the truth value and adjust the weight. Comparing gives a measurement of how much prediction missed by. A good way to measure error is one of the most important and complicated subjects of deep learning. One simple way of measuring error is mean squared error (MSE). Learning tells each weight how it can change to reduce the error. It is all about error attribution, or the art of figuring out how each weight played its part in creating error. It is the blame game of deep learning: gradient descent [52].

At the end of the day, learning is really about one this: adjusting the weights either up or down so the error is reduced. Therefore, we can say learning in neural network is a search problem. We are searching for the best possible configuration of weights so the network's error falls to 0 and predict perfectly.

As we have said before, deep learning is a multistage way to learn data representation. What deep neural networks do the input-to-target mapping via deep sequence of simple data transformations (layers) and that these data transformations are learned by exposure to examples.

The specification of what layer does to its input data is stored in the layer's weights. In technical terms, we would say that the transformation implemented by layers is parameterized by its weight. In this context, learning means finding a set of values for the weights of all layers in a network, such that the network will correctly map example inputs to their associated targets. Deep neural network can contain tens of millions of parameters and we have to find the correct value for all of them.

To control the output of neural network, how far the output is from what is expected needs to be measured. This is the job of the loss function. The loss function takes the predictions of the network and the true target (what we want the network to output) and computes a

distance score (loss score), capturing how well the network has done on this specific example. This score is used as a feedback signal to adjust the value of the weights a little, in a direction that will lower the loss score. This adjustment is the job of the optimizer, which is implemented by what is called the Backpropagation algorithm.

In the learning process, what the neural network does is search for correlation between the input layer and output layer of the training data. If the data have no correlation between the input and output we create an intermediate layer (dataset) that does have correlation with the output layer. The goal is to train the network so that even though there is no correlation between input and output dataset, the layer (dataset) inserted in the middle (which is called hidden layer) will have correlation with the output layer.

If neural network is trained by adding hidden layers as it is, it would not converge. The solution is to make the hidden layers to sometimes correlate and sometimes not correlate. This is called conditional correlation or sometimes correlation. This gives it correlation of its own.

The sometimes correlation is achieved by turning off nodes in the hidden layers based on a condition, for example, whenever it is negative (Relu or Rectified Linear Unit). This logic is called nonlinearity. Without the tweak, the neural network is linear.

Activation function is a function applied to the neurons in a layer during prediction such as ReLu (ReLu is a function that have the effect of turning all negative numbers to 0) [52]. Other activation functions are:

- **Sigmoid**: it converts the infinity amount of input to an output between 0 and 1. It is used both in hidden and output layers.
- **Tanh**: it is better than sigmoid for hidden layers. It is different than sigmoid because it's output lies between -1 and 1. That means it can throw in some negative correlation.
- **Softmax**: it raises each input value exponentially and then divides by the layer's sum.

A **gradient** is the derivative of tensor operation. It is the generalization of the concept of derivations to function of multidimensional inputs (to functions that take tensors an input). Given a differentiable function, it is theoretically possible to find its minimum analytically: it is known that a function's minimum is a point where the derivative is 0, so all we have

to do is find all the points where the derivative goes to 0 and check for which of these points the function has the lowest value. Applied to neural network, that means finding analytically the combination of weight values that yields the smallest possible loss function. In practice, a neural network function consists of many tensor operations chained together, each of which has simple known derivative. Chain of functions can be derived using the chain rule: f(g(x)) = f' (g(x)) * g' (x). Applying the chain rule to the computation of gradient values of neural network gives rise to an algorithm called Backpropagation. **Backpropagation** starts with the final loss value and works backward from the top layer to the bottom layers, applying the chain rule to compute the contribution that each parameter had in the loss value.

There are 3 ways in which we can apply gradient descent to the network [52]: Full, batch and stochastic gradient descent.

Stochastic gradient descent updates weights one example at a time (the idea of learning one example at a time). It performs a prediction and weight update for each training example separately. It iterates through the entire dataset many times until it can find a weight configuration that works well for all the training examples.

(Full) gradient descent updates weights one dataset at a time. Instead of updating the weights once foreach training example, the network calculates the average weight delta over the entire dataset, changing the weight only each time it computes a full average.

Batch gradient descent updates weights after n examples. Instead of updating the weight after just one example or after the entire dataset of examples, a batch size of examples is chosen (typically between 8 and 256), after which the weights are updated. We can use this method to increase the speed of training and the rete of convergence.

Deep learning offered better performance on many problems. In addition, deep learning makes problem-solving much easier by completely automating what used to be the most crucial step in a machine-learning workflow: feature engineering. Humans has to go to great lengths to make the initial input data more amendable to processing by these methods: they had to manually engineer good layers of representations for their data. This is called feature engineering. Deep learning on the other hand, completely automate this step.

2.6.3 Anatomy of Neural Networks

The core building block of neural networks is layer. Layer is the data-processing module that is used as a filter for data. Some data goes in, and it comes out in a more useful form. Specifically, layers extract representations out of the data fed into them. These representations are hopefully more meaningful for the problem at hand.

Current machine learning systems use tensors as their basic data structure. Tensor is a container for data. Metrics are 2D tensors and tensors are a generalization of metrics to an arbitrary number of dimensions. A tensor that contains one number is called scalar whereas tensor that contains an array of numbers (data) is called a vector or 1D tensor. In real world, timeseries or sequential data is represented using 3D tensor (sample, timesteps, feature), images are represented in 4D tensors (sample, height, width, channels) and video is represented using 5D tensor (sample, frame, height, width, channels).

In summary, training neural networks involves the following objects:

- **Layers**, which are combined into a network (or model)
- The **input data** and corresponding targets
- The **loss function**, which defines the feedback signal used for learning
- The **optimizer**, which determines how learning proceeds

Layer is the data-processing module that takes as input one or more tensors and outputs one or more tensors. Some layers are stateless but more frequently layers have a state: the layer's weight. Different layers are appropriate for different tensor formats and different types of data processing. For example, simple vector data, stored in 2D tensors of shape (samples, features), is often processed by densely connected layers, also called fully connected or dense layers. Sequence data, stored in 3D tensor of share (samples, timesteps, features) is typically processed by recurrent layers such as an LSTM layer. Image data, stored in 4D tensors, is usually processed by 2D convolution layers.

Loss function (Objective function) – is the quality that will be minimized during training. It represents a measure of success for the task at hand. The loss function we use depends on the type of classification problem at hand. For binary classification we usually use binary cross entropy or mean squared error, and for multiclass classification problem we use categorical cross entropy (it measures the distance between two probability distribution: here, between the probability distribution output by the network and the true distribution of the label. By minimizing the distance between these two distribution, we

can train the network to output something as close as possible to the true label). Cross entropy is a quantity from the field of Information Theory that measures the distance between probability distribution or, in our case, between the ground-truth distribution and our prediction.

Optimizer – Determines how the network will be updated based on the loss function. It implements a specific variant of gradient descent.

2.6.4 Evaluating Deep Neural Network Models

Evaluating a deep neural network model needs the splitting of the available data into three sets: training, validation and test. We train on the training data and evaluate the model on the validation data. Once the model is ready, it is tested one final time on the test data.

3 sets of data than 2 is needed because developing a deep learning model always involves tuning its configuration: for example, choosing the number of layers or the size of the layers. We do this tuning by using as a feedback signal to measure the performance of the model on the validation data.

There are some methods we use to split the data into training, validation and test sets, such as: Simple hold-out validation and K-fold validation.

Some terminologies are necessary to know before discussing the evaluation metrics. True positive (TP) refers to the number of positive data points that were correctly labeled by the classifier. True negative (TN) is the negative data points that were correctly labeled by the classifier. False positive (FP) is the negative data points that were incorrectly labeled as positive and False negative (FN) are positive data points that were mislabeled as negative.

To evaluate our models, we chose the following measure of success or evaluation metrics:

a) **Accuracy** (recognition rate): The accuracy of a classifier on a given test set is the percentage of test set that are correctly classified by the classifier [57]. That is,

$$accuracy = \frac{TP + TN}{P + N} \tag{2}$$

b) **Precision**: can be thought of as a measure of exactness. In other words, it is the percentage of test data labeled as positive are actually positive [57].

$$precision = \frac{TP}{TP + FP} \tag{3}$$

c) **Recall** (sensitivity): is a measure of completeness, i.e. what percentage of test data are labeled as such [57].

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{4}$$

d) **F1 Score** (harmonic mean of precision and recall): is a measurement that combines precision and recall in to one. By doing that, it gives equal weight to precision and recall [57]. It is defined as the following:

$$F = \frac{2\ x\ precision\ x\ recall}{precision + recall} \tag{5}$$

e) **Confusion Matrix**: is a table of at least size m by m. An entry, CM_{ij} in the first m rows and m columns indicates the number of test rows of class i that were labeled by the classifier as class j. For the classifier to have good accuracy, ideally most of the tuples would be represented along the diagonal of the confusion matrix, from $CM_{1,1}$ to $CM_{m,m}$ with the rest of the entry being zero. That is, ideally FP and FN are around zero [57].

f) **P/R AUC** (area under precision recall curve): Since measure of accuracy is not enough to truly evaluate a classifier [58] we put an alternative measurement called area under precision-recall curve. Precision-recall curve summarizes the trade-off between the TP rate and the positive predictive value for the prediction. Area under the curve summarized the integral or an approximation of the area under the precision-recall curve.

g) **Macro Average**: This will compute the average by giving each class the same weight and divide by the total number of classes. It is calculated as follows:

$$Macro\ Average = \frac{\sum_{i=1}^{n} ci}{n} \tag{6}$$

h) **Weighted Average**: This will compute the average by multiplying each class by its weight (samples in each class) and divide by total number of samples. It is calculated as follows:

$$Weighted\ Average = \frac{\sum_{i=1}^{n}(ci\ x\ wi)}{\sum_{i=1}^{n} wi} \tag{7}$$

2.6.5 Overfitting

Overfitting is extremely common in neural networks. The more powerful the neural network's expressive power (more layers and weights), the more prone the network is to overfit.

Regularization is a key to combatting overfitting in neural network. It is a method for getting a model to generalize to new datapoints instead of just memorizing the training data. It is a subset of methods that help the neural network learn the signal and ignore the noise. It is used to encourage generalization in learned model, often by increasing the difficulty for a model to learn the fine-grained details of training data.

Memorization is when the neural network learned to transform input data to output data for only specific input configurations. It is guaranteed to work well if the new data is nearly identical to inputs in the training data. But neural networks are useful if they work on data we do not already know the answer. Generalization is when the neural network works well on previously unseen data. Memorizing the data is easier than generalizing to all data.

Neural networks can get worse if we train them too much. A more official definition of neural network that overfits is a neural network that has learned the noise in the dataset instead of making decisions based on only on the true signal. Noise is a detailed information (or staffs that are irrelevant to classification). For example, in an image containing a dog, the background could be a noise if the object that we are interested in is a dog. In contrast, the things that tell us the essence of a dog are the true signals, such as edges.

The following are the most common ways to fight overfitting [51, 52]

- **Reducing Network's size**: By network size we mean the number of learnable parameters in the model, which is determined by the number of layers and the number of units per layers.
- **Adding weight regularization**: forcing the weights of the layer to take only small values, which makes the distribution of weight values more regular (weight regularization). It is done by adding loss function of the network a cost associated with having large weights. E.g. L1 and L2 regularization.
- **Early Stopping**: One way to train neural network to only learn the signal and ignore the noise is early stopping. It turns out a large amount of noise comes in the

fine-grained details of an image and most of the signal is found in the general shape and perhaps color of the image. Therefore, the simplest and cheapest regularization method is to stop the network when it starts getting worse, i.e. early stopping. (if we want to ignore the fine-grained details and capture only the general information present in the data.)

- **Adding Dropout**: It is to randomly dropping out or turns off neurons (set them to 0) a number of output features of the layer during training. Dropout makes a big network act like a little one by randomly training little subsections of the network at a time, and little networks do not overfit. This is because small neural networks do not have much expressive power. The cannot learn the more granular details (noise) that tend to be the source of overfitting. They have room to capture only the big, obvious, high-level features.
- **Data Augmentation:** It is a powerful technique for mitigating overfitting in computer vision when there is little data to train.

2.6.6 Convolutional Neural Network

Overfitting is often caused by having more parameters than necessary to learn a specific dataset. When neural networks have lots of parameters but not many training examples, overfitting is difficult to avoid [52]. Regularization is a means of countering overfitting. But it is not the only technique to prevent overfitting.

As mentioned before, overfitting is concerned with the ratio between the number of weights in the model and the number of data points it has to learn those weights. Thus, there is a better method to counter overfitting. When possible, it is preferable to use something loosely defined as structure. Structure is selectively choosing to reuse weights for multiple purposes in a neural network because the same pattern needs to be detected in multiple places [52].

Normally removing parameters make the model less expressive (less able to learn patterns), but if we are clever in where we reuse weights, the model can be equally expressive but more robust to overfitting. Perhaps surprisingly, this technique also tends to make the model smaller (because there are fewer actual parameters to store). The most famous and widely used structure in neural networks is called convolution, and when used as a layer it is called convolutional layer [52]. Convolutional networks are simply neural

networks that use convolution in place of general matrix multiplication in at least one of their layers [54].

In convolution layer, lots of very small linear layers are reused in every position, instead of a single big one. The core idea behind a convolution layer is that instead of having a large, dense linear layer with a connection from every input to every output, lots of very small linear layers are used, usually with fewer than 25 inputs and a single output, which are used in every input position. Every mini-layer is called a convolutional kernel, but it is really nothing more than a baby linear layer with a small number of inputs and a single output.

Shown here in *Figure 16*, is a single 3x3 convolutional kernel. It will predict in its current location, move one pixel to the right, then predict again, move another pixel to the right, and so on. Once it has scanned all the way across the image, it will move down a single pixel and scan back to the left, repeating until it has made prediction in every possible position within the image. The result will be smaller square of kernel predictions, which are used as input to the next layer.

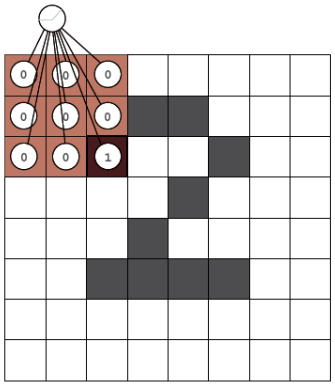


Figure 16 Convolution Kernel [52]

A typical layer of convolutional network consists of three stages [54]. In the first stage, the layer performs several convolutions in parallel to produce a set of linear activations. In the second stage, each linear activation is run through a nonlinear activation function, such as the rectified linear activation function. In the third stage, a pooling function is used to modify the output of the layer further. The pooling function replaces the output of the network at a certain location with a summary statistic of the nearby output. For example, the max pooling operation reports the maximum output within a rectangular neighborhood.

The fundamental difference between a densely connected layer and a convolution layer is the dense layer learn global pattern (patterns involving all pixels) in their input features, whereas convolution layers learn local patterns (patterns found in the kernel). This key characteristic gives convolutional networks two interesting properties [51]:

- The pattern they learn are translation invariant.

- They can learn spatial hierarchies of patterns. The first convolution layer will learn small local patterns such as edges, a second convolution layer will learn larger patterns made of the features of the first layers and so on.

2.6.7 Recurrent Neural Network

Recurrent neural networks (RNNs) are family of neural networks for processing sequential data [54]. Densely connected networks and convolutional neural networks have no memory. Each input is processed independently, with no state kept in between inputs. RNNs processes sequences by iterating through the sequence elements and maintaining a state containing information relative to what it has seen so far. An RNN is a type of neural network that has an internal loop. In RNN data point is no longer processed in a single step, rather the network internally loops over sequence elements. Easy enough, an RNN is a for loop that reuses quantities computed during the previous iteration of the loop [51].

Simple RNN has major issue: in practice long-term dependencies are impossible to learn. This is due to the vanishing gradient problem, as we keep adding layers to a network, the network eventually becomes untrainable [51]. The LSTM (Long Short-Term Memory) layer, GRU (Gated Recurrent Unit) layers and Independent Recurrent Network (IndRNN) are designed to solve this problem [51]. LSTM layer adds a way to carry information across many timesteps. It saves information for later, thus preventing older signals from gradually vanishing during processing. GRU layer works using the same principle as LSTM, but it is somewhat streamlined and thus cheaper to run, although they may not have as much representational power as LSTM.

2.6.8 Action Classification

Action classification in video needs to solve two problems. The first one is extracting the appearance information of moving human body and the second one is simulating the temporal dynamics of action [59]. CNN can capture the complex spatial appearance information from an image. However, using CNN for video classification will cause the loss of temporal context information in the video sequence. Therefore, to model the temporal context information, the deep learning models have been categorized into 4 [59]. Action recognition based on 3D CNN [38], action recognition based on spatio-temporal two stream networks [39], action recognition based on RNN and action recognition based on skeleton sequence [40].

Chapter 3: Related Work

3.1 Introduction

This chapter is about the review of the papers that have been published regarding both music and video classification using the two approaches, traditional machine learning and deep learning. It has 3 sections. The first Section, 3.1, is about music classification while the second Section, which is 3.2 is about video classification. In Section 3.3, we summarize things that have discussed in this chapter.

3.2 Music Classification

In this section we are going to review works related to music genre classification. The first section is the review of works that use traditional machine learning approach and the second section is about works that use deep learning to solve the problem of music genre classification.

3.2.1 Music Classification using Machine Learning

Tzanetakis and Cook [24] proposed automatic classification of audio signals into hierarchy of music genres. At the top of the hierarchy is an audio signal which will be classified as either speech or music. The speech is again classified as male, female and sports while the music will be classified to 10 genres namely: Classical, Country, Disco, HipHop, Jazz, Rock, Blues, Reggae, Pop and Metal. Moreover, the classical category will again be classified in to 4 sub-categories (Choir, Orchestra, Piano, String Quartet) and the jazz will be classified to 6 sub-categories (BigBang, Cool, Fusion, Piano, Quarter, Swing). Even though there is no clear boundary and strict definition for a music genre, according to the authors, members of particular genre share some common characteristics related to instrumentation, rhythmic structure and pitch content of the music. The authors proposed 30 set of features for representing timbral texture (19 features), rhythmic content (6 features) and pitch content (5 features). Out of the 19 features that were proposed to represent the timbral texture 9 of them were calculated based on Short Time Fourier Transform (STFT) and the rest using Mel-Frequency Cepstral Coefficient (MFCC). For the purpose of classification, they used a number of statistical pattern recognition (SPR) classifiers to estimate the probability density function (PDF) for the feature vectors of each class. For each speech types and music genres, the authors used 100 representative

excerpts each 30s long. To ensure that the training set contains variety of recording qualities, the excerpts were taken from radio, compact disc and MP3 compressed audio files. To evaluate the proposed features performance and importance, they used ten-fold cross-validation evaluation technique with 100 iterations and achieved an accuracy of 61% which was comparable with the human genre classification. Even though this work was a pioneer on music classification and led the foundation for the following researches, it did not include the temporal features found in the music data. In addition, better classification algorithms have come recently, such as the use of deep learning that took the research on music classification to the next level.

Ren *et al.* [25] tried to include the sequential temporal information when classifying music genre which was not included in the previous approaches. The authors believed music can be viewed as a sequence of sound events. Therefore, they proposed a song tokenization method, which is transforming the music in to sequence of units. In this method they converted the music into a sequence of tokens using acoustic segment models (ASMs). The tokens have variable length which is determined using Hidden Markov Model (HMM). Then they integrated the temporal information of the music signal to the HMM. By using the music tokenization method, they converted the problem of music genre classification to text classification in which common text information retrieval technique can be used to tackle the problem. Although the technique that is usually used in text classification uses statistics of unigrams and bigrams, the technique cannot capture all temporal structures in music. For example, the repeatedly performed similar patterns of piano in classical music indicate the importance of long-term structure which cannot be captured using the unigram/bigram. Therefore, the authors suggested a data mining technique named AprioriAll to find the repeated sequential patterns (SPs). In this technique, the transcript of ASM indices of the same genre that were acquired from each song in the training corpus are utilized to discover long-term structure represented by sequential patterns (SPs). Given a collection of music transcripts of the same genre, and minimum support threshold which decides whether a mined pattern is frequent or not, AprioriAll algorithm was used to find SPs. Next, how many times an SP appears in the transcript of ASM indices of a song is counted. Finally, for the SPs mined from each genre a corresponding SP-song matrix can be created which will be used to design the classifier. Combining the SP-song matrix with the unigram-song and bigram-song matrix a vector modeling of the song will be created which will be used to represent the song. Next, Latent

Semantic Analysis (LSA) was performed to get matrix with entropy weighted counts. Finally, nearest neighbor classified is used to evaluate the performance. The authors used 2,035 30-second clips for training and 926 30-second clips for testing. The dataset used by the authors was Magnatunes. The music will be classified into 5 genre namely Classical, Electronic, Jazz/Blues, Metal/Punk and Rock. The observation the authors got from their experiment was the SPs mined from the class "Classical" has the most discriminative power and the class "Metal/Punk" and "Rock" are similar, indicating these two classes are highly confusable. The result of the experiment showed that their proposed method, which is introduction of sequential patterns outperformed the previous approaches that only considered local temporal feature. They got an accuracy of 74.19%. Although this work includes the sequential information found in the music and was able to acquire a better result than the previous research [24], again its accuracy could be increased by the use of deep learning.

3.2.2 Music Classification using Deep Learning

Wu *et al.* [55] did a research with the objective of using independent RNN (IndRNN) to the music genre classification problem. They claimed that IndRNN can learn long-term relationships better than LSTM and GRU. This is because LSTM and GRU use sigmoid and hyperbolic tangent function that may result in gradient decay in the deep network. As a result, the network cannot work in long time scale like long music clip. To solve this problem the authors suggested 5-layer IndRNN and the use of scattering transform for data preprocessing in order to keep information loss as less as possible. Additionally, they use ReLU function and softmax classifier. Furthermore, dropout of 0.5 and learning rate of 1e-5 was chosen. The dataset used by the authors was the GTZAN dataset that was collected by Tzanetakis and Cook [24]. The authors used 10-fold cross-validation for evaluation. Their experimentation showed that classification accuracy of 96% was achieved with the IndRNN and 97% with LSTM but the training time for LSTM was much worse (0.68s per iteration) than IndRNN (0.23s per iteration). This research showed that the using deep learning approach for music classification gives a greater accuracy but it only focuses on using the audio feature of the music and did not consider adding the visual data.

Panwar *et al.* [60] proposed a hybrid approach to solve the music genre classification. They use CRNN architecture which is the combination of CNN and RNN. CNN was used to

extract the local features while RNN was used to discover global features to understand the temporal context. The dataset used by the authors is the MagnaTagATune. The raw music in mp3 format was transferred to mel-spectrogram signal in the pre-processing stage. This is then fed to the proposed CRNN architecture which was designed to understand the local and temporal features. This architecture has 4 convolutional and 2 recurrent neural network layers. The RNN layers with GRU are used for temporal pattern summation over features learned in CNNs. The model was trained for more than 60,000 iterations with a batch size of 32, completing a total of 100 epochs. To reduce the internal covariant shift, which is a phenomenon that slows down the learning rate, the authors use batch normalization which is performed after each convolution layer and before activation function. The activation function that was used after each convolution layer was the Exponential Linear Unit (ELU). In addition, ADAM (Adaptive Moment Estimation) optimizer was used for learning control and binary cross entropy was used for loss calculation. To determine the accuracy of the proposed architecture, the authors use Area Under and ROC (Receiver Operating Characteristic) Curves. The AUC-ROC for the proposed model was 0.893. The same as the previous research [55], this work also did not consider the visual feature of the music. Its only focus was on the audio data.

Song *et al.* [56] investigated the use of transfer learning to music genre classification. Transfer learning solves the problem of having small dataset by pre-training the network in a similar task which has enough data, then fine-tuning the parameters of the pre-trained network using the target dataset. The authors used the Magnatagtune dataset for pre-training the proposed five-layer RNN with GRU using softmax as a classifier. The overall process of their proposed architecture consists of two parts. One part is a deep RNN training on large music dataset (Magnatagtune) and the other part is genre classification process after fine-tuning the previous trained deep RNN using GTZAN dataset. Scattering transform is applied at the beginning of each part in order to reduce the raw music data and to extract features. It can provide invariants over large time scale without too much information loss. The dropout of their model was set to 0.7 and the learning rate was 0.00001. They use AUC-ROC score to evaluate the performance of their model with a 10-fold cross validation. The transfer learning model became stable after 100 epochs while the random initialed model needed more. The authors have achieved an accuracy of 95.8% when using transfer learning and 93.5% when using random initialization. This research makes use of fine-tuning of hyper parameters of a pre-trained network which would

require a machine with high performance processing power and GPU that is not easily available for the research we are conducting. In addition, this research did not make use of the visual feature of the music.

3.3 Video Classification

Since dance movements can be considered as human action, in this section we are going to consider dance classification problem the same as human action recognition (action classification).

3.3.1 Video Classification using Machine Learning

The work done by Kapsouras *et al.* [34] deal with the recognition and classification of Greek folk dance. The aim of the research was to apply a recognition framework in general activity recognition on the specific sub-task of folk dance recognition. The authors specifically checked whether that approach can operate sufficiently on folk dance videos. To represent the video data, feature vectors are used and K-means was applied on the feature vectors to create a codebook consisting of the centroids V_c, c=1, …, C where C is the number of the K-means clusters. Each centroid represents a code word of the codebook. Then, the feature vectors are mapped to its closest code word using Euclidian distance. Next, for each training sequence the frequency of appearance of every codeword is computed and thus, a histogram for each sequence that characterizes is formed. For the unlabeled dance video sequence, the same procedure is used, thus, the feature vectors of the test sequence are mapped to the closest codewords of the codebook and the histogram of codewords that characterizes the test sequence is formed. Then, SVM classifier trained using the histogram of the training set is used for the classification. They used a non-linear SVM with X^2-kernel. To learn the unsupervised features from the video the authors used two approaches: Independent Space Analysis (ISA) and Space-Time Interest Points (STIPS). The ISA network is a network that can be described as a two-layered network with square and square root nonlinearities in the firsts and second layer respectively. The other method, which is STIPS is based on the evaluation from each action video and their description by a set of Histogram of Oriented Gradients/ Histogram of Optical Flow (HOG/HOF) descriptors, which refers to local shape and motion. In this method, they used the Harries3D detector, in order to detect video locations where the image intensity values undergo significant spatio-temporal changes. The dataset used by the authors contains 4 videos of 2 Greek folk dances namely Lotizia and Capetan Loukas. 1 video from each

were selected for training and the other for testing. Some of the videos were recording indoor while some were recorded outdoor which makes the recognition setup more realistic but also challenging. The authors did an experiment to know about the classification rate of the two methods and they get 89.47% classification rate using the ISA method and 87.08% using the STIP method. The down side of this research is it uses traditional machine learning approach that extracts features manually which is a tedious task.

Kumar *et al.* [33] tried to classify the Indian classical dance with an objective of extracting Indian classical dance poses from both online and offline videos given a specific dance pose sequence as input. Among the constraints the authors have stated are video resolution, frame rate, background lighting, scene change rate and blurring. Automatic dance motion extraction is complicated due to complex poses and actions performed at different speeds in sink to music or vocal sound and in the case of Indian classical dance the features representing the dancer should focus on the entire human body shapes. The Indian dance forms are a set of complex body signatures produced from rotation, bending and twisting of fingers, hands and body along with their motion trajectory and spatial location. As the authors stated there are 8 different classical Indian dance forms and extracting these complex movements from online videos and classification requires a complex set of algorithms working in sequence. To solve the problem, the authors proposed to use silhouette detection and background elimination, human object extraction, local texture with shape reference model, and 2D point cloud to represent the dancer pose. Five features are calculated that represent the exact shape of the dancer in the video sequence. For recognition, a multiclass multilabel Adaboost (MCMLA) algorithm is proposed to classify query dance video based on the dance dataset. Wavelet reconstructed local binary patterns were used for feature representation preserving local shape content of hands and legs. Two fusion models were proposed for feature fusion. Early fusion at segmentation stage with PCA based Haar wavelet and LBP is used and late fusion using the Zernike moments, Hu moments and shape signature is used with Haar and LBP is proposed. The Adaboost can effectively recover the query video frames from the dataset, by shape-texture observation model defined by discrete wavelet transform (DWT) and local binary patterns (LBP). The early and late features and classifiers performance tests show that the proposed late fusion features and multiclass multilabel Adaboost classifier give better classification accuracy and speed compared to AGM and SVM. The authors conducted a total of 8 experiments.

The first 4 experiments were for the early fusion and the last 4 experiments were for late fusion. From the 4 experiments two of the were for online video and the rest two were for the offline videos. For the offline videos the accuracy ranges from 0.74 (early fusion) – 0.99 (late fusion) and for the online videos the accuracy ranges from 0.66 (early fusion) – 0.99 (late fusion). In this research, the authors used only shape features which cannot be used directly by Ethiopian music classifier. Adding additional information to the classifier such as movement of dancers would give more clue when it comes to Ethiopian dance movements. In addition, extracting features manually is not the ideal solution now a day in image processing problems.

Fitsum Tamene [37] did a research on the classification of Ethiopian traditional dance using visual features. In his research, he considered 10 traditional style of songs namely Eskista (only the Gondar type), Gurage, Asosa, Somali, Oromo, Afar, Hararghe, Tigrinya, Wolaita and Gambella. He applied HOG on each frame to detect region of interest i.e. dance performer. Then he extracted local features related to body shape. In order to achieve that he used Log Gabor filter. As feature selection technique the author have used Principal Component Analysis and finally the selected features were used directly by the classifier. The learning algorithms used to build the classifier model were SVM and Naïve Bayes. The performance of the classifier when using SVM was 76% and 64.5% when using Naïve Bayes. In this research, the selected feature was shape and we believe, if other features such as movement of the dancers were integrated, the performance of the classifier would have been increased. In addition, the author was very selective when choosing video clips that were included in the dataset, such as background and we believe the performance of the model would be very low on real time data that includes different types of background.

The work done by Andargie Mekonen *et al.* [35, 36] aims at classifying ten Ethiopian traditional dance namely: Afar, Benshangul, Gambella, Eskista, Gurage, Hararghe, Oromo, Somali, Tigrinya and Wolayta. They used optical estimation to estimate the motion of the dancers as a way of feature extraction and HOG to detect region of interest. As optical flow estimation method, the authors use Horn-Schunck and Lucas-Kanade optical flow algorithm. Both works are basically the same but their difference lies on the learning algorithm used. While the learning algorithm used in [36] is SVM, the learning algorithm used in [35] is ANN. When using SVM the maximum accuracy the authors got is 99.7 and it is achieved when the Lucas-Kanade method is used. The maximum accuracy they acquired when ANN is used is 82.0% and it is again achieved when Lucas-Kanade

method is used. Despite the fact that these two works include the motion information in the feature extraction step, they were very selective when choosing data (with no background movement) making their model less performant on real environment. In addition, they only train and test their system with only one person as a dance performer resulting in using a dataset that did not represent Ethiopian traditional dance which mostly performed in groups.

3.3.2 Video Classification using Deep Learning

Ji *et al.* [38] develop 3D CNN model for action recognition. Their model extracts feature from both the spatial and temporal dimensions by performing 3D convolutions, thereby capturing the motion information encoded in multiple adjacent frames. They develop a 3D CNN architecture that generates multiple channels of information form adjacent video frames and perform convolution and subsampling separately in each channel. The final feature representation is obtained by combining information from all channels. To boost the performance, the authors proposed regularizing the 3D CNN model by augmenting the model with auxiliary outputs computed as high-level motion features. In addition, combining the outputs with high-level features was proposed. Their method could be applicable to scenes such as noise, background interference and occlusion. The authors evaluated the model on the TREC Video Retrieval Evaluation (TRECVID 2008) data, which consist of surveillance video data recorded at London Gatwick Airport and KTH dataset and they get an accuracy of 0.8853 AUC when the false positive rate is 1%. Their experiment results show its effectiveness in real environment but the performance of their Training a model that is designed using 3D CNN from scratch requires large amount of video data, memory and processing power which are impossible to have at the time this research was conducted. Therefore, designing an architecture that can process both visual and sequential information in more efficient manner is needed.

Simonyan *et al.* [39] aim at extending deep ConvNets to action recognition in video data. Their architecture was based on two separate recognition streams (spatial and temporal) that are implemented as ConvNets and are then combined by late fusion. The spatial stream performs action recognition from still video frames, whereas the temporal stream is trained to recognize action from motion in the form of dense optical flow. They stated that decoupling the spatial and temporal networks allowed them to use transfer learning. The dataset used by the authors were UCF-101 and HMDB-51. They get an accuracy of 88.0%

on UCF-101 dataset when the fusion is done by SVM and 59.4% on HMDB-51 dataset. Even though this research focuses mainly on action recognition, we cannot use it as it is to classify Ethiopian traditional music due to the fact that the authors did not including audio data. We have to find a way to include audio features to the video features to maximize the accuracy of our system.

Li *et al.* [40] tried to recognize a 3D human action using and end-to-end deep CNN. They proposed a representation of the dataset using skeleton sequence rather than RGB or depth. A pre-trained VGG-19 model was used by the authors to perform the learning. They first performed data pre-processing to normalize and filer the skeleton joints. Secondly, they transformed a skeleton sequence with an arbitrary number of frames into RGB images. Finally, they fed the generated images in to VGG-19 [61] model, which is a pretrained model with ImageNet [62] to perform fine-tuning and action recognition. Finally, they performed end-to-end fine tuning and perform the learning process. For the hyper parameter setting, they selected 40 for the size of mini-batches for the SGD. In addition, initial learning rate of 0.001, which was decreased by multiplying it by 0.1 at every $10{,}000^{th}$ iteration was set. Moreover, 50 epochs were chosen by the authors to stop the fine-tuning process. They tested their proposed method on NTU RGB-D dataset, which is at that time the largest skeleton based human action recognition dataset and achieved 82.1% for cross-view and 75.2% for cross subject when evaluating. They also experimented using AlexNet [63] as a pre-trained model but got lower accuracy (72.96 % and 75.87%). We do not have skeleton based dataset that shows Ethiopian traditional dance dataset. In addition, performing fine tuning requires access to GPU and needs large amount of training time.

Ijjina *et al.* [41] investigated the problem of human action recognition by proposing 2D CNN for human action recognition using action bank features. Their main motivation was the use of frequency spectrogram of audio data for speech recognition using CNN. As it is stated in their paper, human action recognition in vides is a challenging task due to the existence of time dimension, i.e. the length of a video capturing the same action may be different. Consequently, they proposed an approach that uses action bank features. Action bank features of a video capture the similarity information of the video against the videos in an action bank. Thus, videos containing similar actions may contain similar patterns in action-bank features. The action bank feature is given as an input to the pattern recognition module for action recognition which is implemented using CNN. The classifier utilizes

this similarity information to assign an action label to the input video. The use of action bank feature to represent a video results in fixed size representation of videos irrespective of their length. Therefore, a deep neural network architecture can be trained to classify a video in a single pass without using temporal window on video frames. The authors consider a CNN classifier to recognize and classify human action from local pattern in the action bank features due to the success of deep CNN architectures for recognizing local visual patterns. UCF50 dataset is used by the authors and the CNN is trained using batch propagation algorithm in batch-mode. The performance of the CNN classifier during training was evaluated after ever 5 epochs on the test dataset. By considering 5 epochs as an iteration, the authors train the CNN 200 times (1000 epochs) maintaining the CNN classifier with minimum error at the end of iteration. They evaluated the proposed solution using two methods. The first one is 5-fold cross validation and the second one is splitting the entire data into 3 groups to perform 3-fold cross-validation. They achieved an accuracy of 94.02% using 5-fold and 93.76% using the 3-fold. The same as the above research [39], this work did not include audio feature and cannot be used directly as it is to classify Ethiopian traditional music using both audio and visual feature.

3.4 Summary

The research by Tzanetakis *el al*. [24] and Ren *et al*. [25] was purely using the traditional or shallow machine learning approach to do the audio music classification and the use of traditional machile learning for music genre classification have already been outshined by the use of deep learning. We can see the improvement of the result uisng deep leaning approach in many researches [55, 60, 56]. The accuracy has improved tremondously to around 97% [55] using the same dataset used by [24] that used the tradition machine learning approach. Therefore, we can infrer from the results of the previous researches, the use of deep learning approach for music genre classification is by far the best approach given we have the required amount of data and processing power.

Researches in [33, 34, 37, 35, 36] all use the traditional machine learning approach to do dance classification and none of them consider the audio of the music when doing the classification. From the above researches, only [37, 35, 36] focuses on Ethiopian traditional music in which all of them did an investigation on video clips with a single dancer and the research that is presented here is not limited to only one dancers and will investigate dances which are performed by groups or couples equally.

Researches in [38, 39, 40, 41] are all action or activity recognition and all use deep learning approach to solve the problem.

But the genre of Ethiopian music is a bit different from the genres that have been investigaed by the previous researches. The music genre in Ethiopian traditional music is more or less related to the ethenicity or culture of the people. As a result, there are some similarity such as use of the same kind of musical instruments. Moreover, the intermingleance of culture between two or more ethinicity makes it harder to put clear boundary in the music or dance movement, we can see this effect on the music of the Wollo people. In addition, most of the songs in the northern and cetral part of the contry use the diatonic scale [18] leading to a similar sound.

Even though a number of researches have been conducted on classification of music using audio processing, there has not been any research that considers information extracted from both audio and video for the purpose of Ethiopian music classification. This research, therefore, aims to fill this gap by investigating the use of deep learning in Ethiopian traditional music classification.

The summary of related works is shown in Table 1 and Table 2.

Table 1 Summary of Related Work on Music Genre Classification

Author	Feature	Algorithm	Accuracy	Dataset	App	Year
Tzanetakis *et al.* [24]	STFT, MFCC	SPR	61%	GTZAN	ML	2002
Ren *et al.* [25]	Sequential temporal pattern	Nearest neighbor	74.19%	Magnatunes	ML	2010
Wu *et al.* [55]		IndRNN/ LSTM	96/97%	GTZAN	DL	2018
Panwar *et al.* [60]	Mel-spectrogram	CRNN	0.893	MagnaTagATune	DL	2017
Song *et al.* [56]	transfer learning	RNN with GRU	95.8%	Magnatagtune/ GTZAN	DL	2017

Table 2 Summary of Related Work on Action/Dance Classification

Author	Feature	Algorithm	Accuracy	Dataset	App	Year
Kapsouras *et al.* [34]	K-means, codewords, ISA, STIPS	SVM	89.47%- ISA, 87.08%- STIPS	Personal	ML	2013
Kumar *et al.* [33]	Wavelet reconstructed local binary patterns	Multiclass Multilabel Adaboost (MCMLA)	0.66 – 0.99	Personal	ML	2017
Fitsum Tamene [37]	Log Gabor filter	SVM and Naïve Bayes	76% and 64.5%	Personal	ML	2013
Andargie Mekonen *et al.* [21]	optical estimation (Horn-Schunck and Lucas-Kanade)	ANN	82.0%	Personal	ML	2016
Andargie Mekonen *et al.* [22]	optical estimation (Horn-Schunck and Lucas-Kanade)	SVM	99.7%	Personal	ML	2016
Ji [38] *et al*		3D CNN		TRECVID 2008/ KTH	DL	2013
Simonyan *et al* [39]		ConvNets + optical flow	88.0%/59.4%	UCF-101 and HMDB-51	DL	2014
Li *et al.* [40]		CNN + Skeleton Sequence + transfer learning	82.1%	NTU RGB-D	DL	2017
Ijjina *et al.* [41]		2D CNN	94.02%	UCF50	DL	2014

Chapter 4: Design of Ethiopian Traditional Music Classifier

4.1 Introduction

In order to solve the problem that we have stated in Section 1.3, we have designed a deep neural network architecture that takes both audio and visual information of a music as an input and outputs the class of traditional music. The architecture is composed of 4 components each performing different processes. The first one is the audio feature extracting component which does the audio feature extraction process. The second component, which is the video feature extracting component is used for video feature extraction. Thirdly we have the audio-visual feature merging component that merges the audio and video features and a finally there is a prediction component that outputs the class of the music.

4.2 Architecture of Ethiopian Traditional Music Classifier

As discussed above the architecture of the Ethiopian Traditional Music Classifier (ETMC) is composed of 4 components and the 4 components are shown on Figure 17. This architecture is designed to classify Ethiopian traditional music using both audio and visual features that performs well in a situation where the available data is small and processing power of the machine where the model is trained on is very low. The problem of small dataset is tackled by using transfer learning. In addition, to avoid the situation where we have a very low processing power and memory, each step in the whole process were partitioned to be independent components. By doing that, when training the model is needed by changing its configuration, each and every step will not be repeated. The only process that needs to be trained is the last prediction (classifying) process.

The audio features from the video will be extracted beforehand and saved on a permanent storage to be concatenated to the visual features latter on. The extraction of audio features happens only once and is not repeated every time the model is trained. The same is true for the visual features. The visual features are extracted using a pre-trained network called VGG-16 prior to the model training and saved on a storage. Then these visual features are given to another module so that sequence or movement information can be extracted. The visual features along with the sequential information will be concatenated to the audio

features by the merging component and given to the predicting component that decides the final class of the music video.

Therefore, by storing the audio and visual features on a permanent storage, running the whole network from end to end is not needed when training the model by changing some configuration. We will discuss each component in detail in the next section.

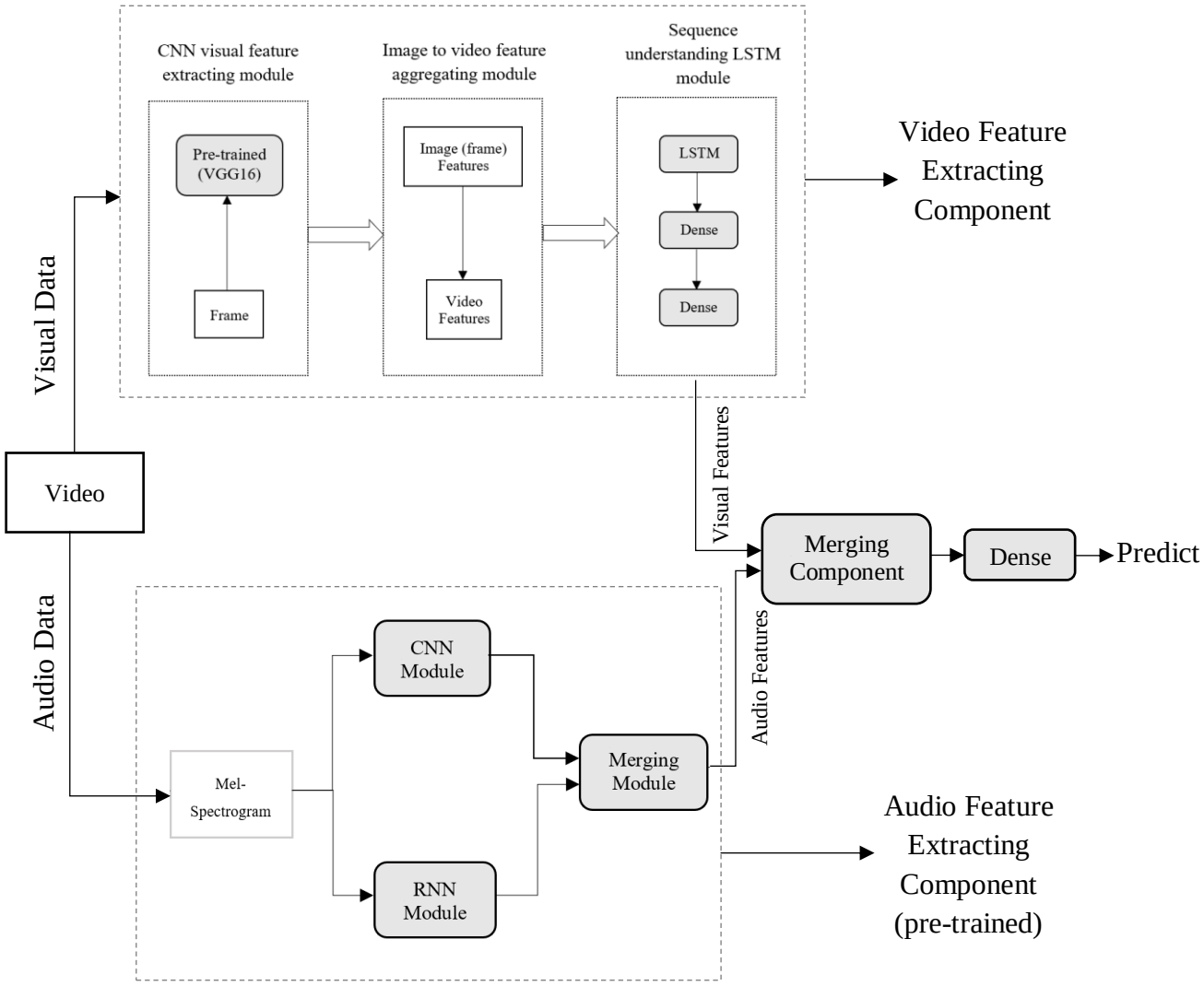


Figure 17 Architecture of Ethiopian Traditional Music Classifier

There are few points worth mentioning about the proposed architecture that makes it unique from others. The first one is that the architecture is made up of from a custom pre-trained audio feature extracting component. This pre-trained feature extracting component will solve our main problem, overfitting, that is caused by not having enough video data (examples) to train the model. At the time this research was conducted there was no a publicly available pre-trained network that was trained on a similar data such as any kind of sound or music. Consequently, the option we had was to train a network with as much as possible Ethiopian traditional music audio data so that we can use it to extract features from the Ethiopian traditional music video.

The second thing that makes this architecture different from the other Ethiopian traditional music classifiers, other than the use of deep learning architecture is that the addition of the audio features to the video features. Moreover, this architecture is able to extract visual features that are related to shape, color (costume) and movement of dancers. The previously conducted researches extracted only one of the visual features such as shape [37] and movement [35, 36].

Lastly, this architecture is composed of independent processes that can run independently without the interference of the rest of processes. Rather than conducting training of the model from end to end, we decided to build an architecture by partitioning it into components each doing a particular task so that we can save time (processing power) and memory. As a result, we were able to train the model in a computer with lower processing power, memory and in a short period of time.

4.3 Audio Feature Extraction

Audio feature extraction is the process in which we extract audio features from the video. This process happens in the audio feature extracting component. The audio feature extracting component is made up of a custom pre-trained network that was trained on a larger number of Ethiopian traditional audio data. This process involves training the network on a large number of audio data and used that network to extract audio features from the video. The first step in the audio feature extraction is extracting mel-spectrogram features from the audio of the video. The mel-spectrogram features will pass through a parallel convolution and recurrent layer. The final output from the two layers will then be merged and taken as the audio feature of the audio. The audio features will then be

consumed by the audio-video feature merging process that merges the visual features with the audio features. The audio feature extraction process is shown in Figure 18.

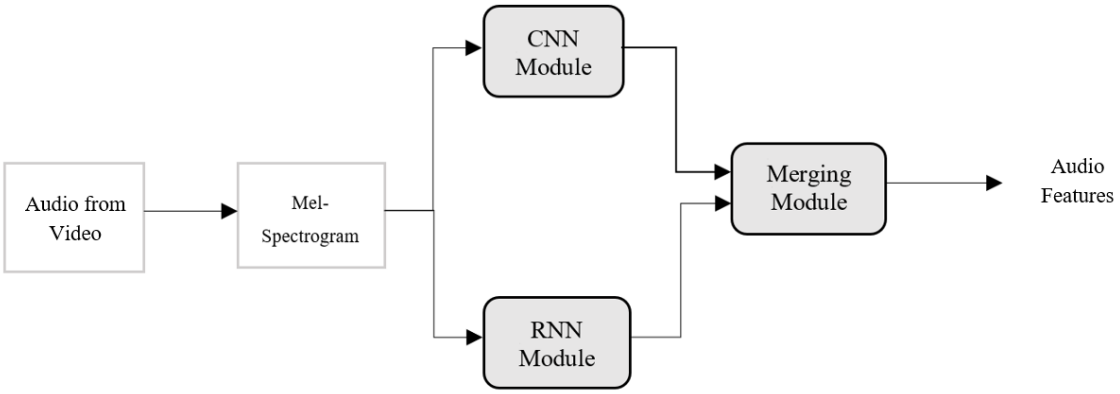


Figure 18 Audio Feature Extracting Component

4.3.1 Mel-Spectrogram

Raw audio data cannot be processed by deep learning models. Therefore, we need to change its representation to make it processable by the model. There are many audio features, such as, Tempogram, Chroma, Mel-Spectrogram, MFCCs, Root Mean Squared (RMS), Zero crossing rate and so on. The one we choose for our model is Mel-Spectrogram. Mel-Spectrogram is when the regular spectrogram (visual representation of spectrum of frequencies over time) of an audio squashed using mel scale to convert the audio frequencies into something a human is more able to understand. A regular spectrogram is a squared magnitude of the short time Fourier transform (STFT) of the audio signal. Mel-spectrogram is chosen for this model because of the performance it showed in different NLP and other audio classification problems [60]. In addition, the mel-scale was proven to be similar to the human auditory system [64].

4.3.2 Architecture of the Audio Feature Extracting Component

The audio feature extracting component of the model consists of 2D-CNN and RNN. As we discussed above, Mel-Spectrogram can be considered as visual representation of an audio music. Therefore, 2D CNN can be used to extract the visual features. Moreover, to process and extract sequential information in the audio RNN (GRU) is used. Rather than training the CNN and RNN separately, a better way is to jointly learn a more accurate model of the data by using a model that can see all available input modalities simultaneously.

Due to the small amount of data that will be used for training of the pre-trained audio feature extracting network, overfitting is expected to be the number one challenge. Therefore, we propose the use of dropout layer at both the convolution and recurrent layer.

4.3.3 Development of Audio Feature Extracting Component

The audio feature extracting component is developed according to the high-level architecture shown in Figure 18. It is implemented using the Keras Functional API. The input to this component, as discussed above, is the mel-spectrogram features extracted from the audio. The mel-spectrogram feature has a dimension of 646 by 128 and expanded to have a dimension of 646 by 128 by 1 i.e. we have changed the dimension from 2D to 3D. The input is given to both the convolution block and recurrent block of the model at the same time so that both blocks can see the feature at the same time.

The convolution block consists of 5 2D-CNN layer each followed by a 2D max polling layer as shown in Figure 19. All the 2D-CNN use relu as their activation function. The output of the last max-pooling layer will then be fed to a Flatten layer so that it is dimension is changed from 3D tensor to a 1D tensor. This 1D tensor will then be used as an input to a Dropout layer (0.5) so that overfitting will be prevented from happening.

The recurrent block consists of a bi-directional GRU layer with a dropout and recurrent drop out value of 0.2. Before the input (mel-spectrogram feature) is fed to this GRU layer, it will first pass through a 2D max-polling and lambda layer. Then the output of the lambda layer is given to the bi-directional GRU layer.

The output from both the convolution and recurrent layer is given to a concatenating layer and the output of this concatenating layer will finally be fed to another concatenating layer so that audio features would be merged to the output of the video feature extracting component. By doing this we were able to extract 384 features from the audio data.

4.4 Video Feature Extraction

Video feature extraction is the process by which we extract visual features from the video. This process happens in the video feature extracting component. This component is composed of a serious of convolution layers followed by LSTM layer. The video feature extraction process begins by converting images found in each video to their respective RGB values. Then the RGB values are given to a CNN so that visual features are extracted. Following this layer is an LSTM layer that is used to extract the sequential information.

Since we are working with a series of images (frames) we need the LSTM layer to understand the relationship between the previous image and the next image. By doing this, motion or movement of the dancers could be understood by the model.

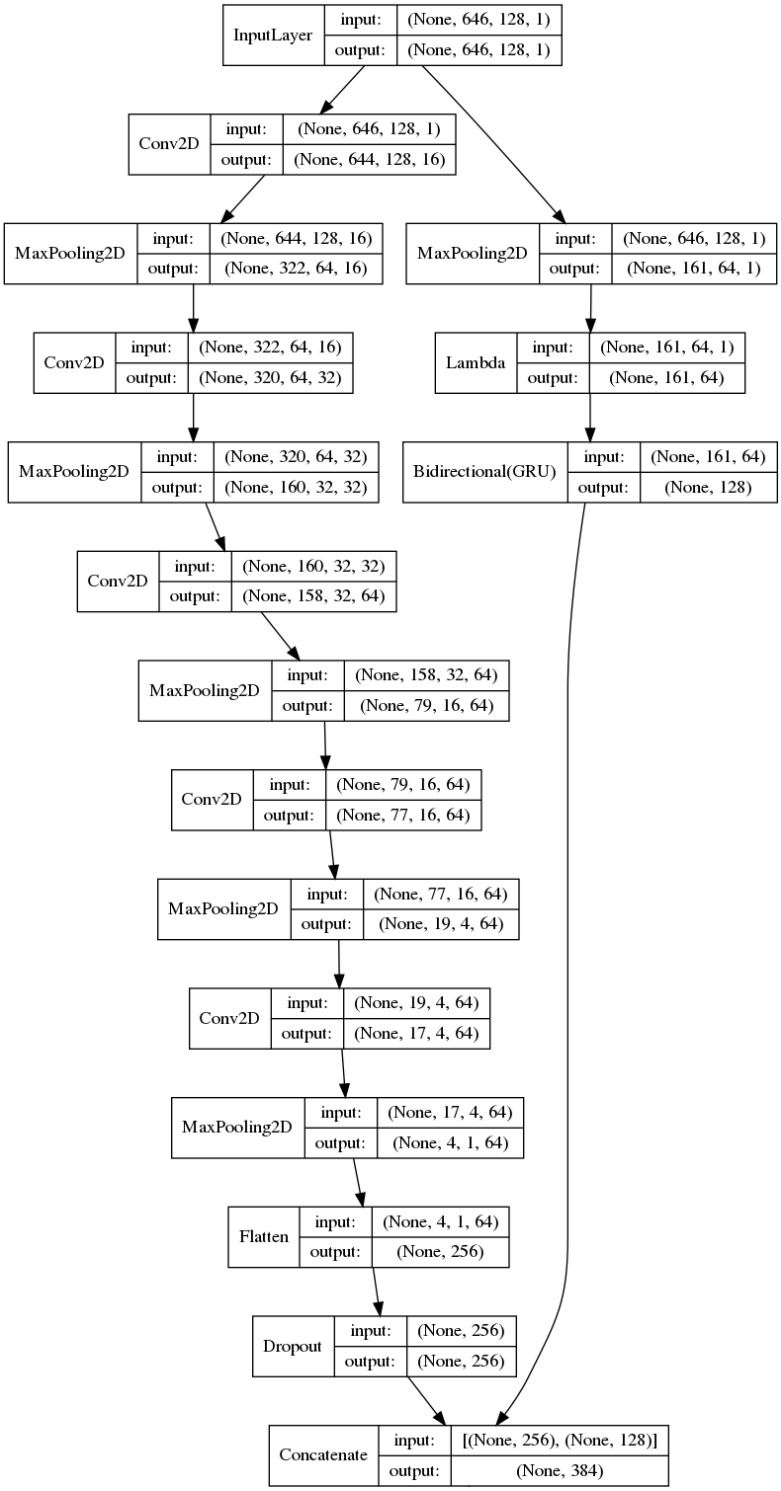


Figure 19 Detail View of Audio Feature Extracting Component

In the video feature extraction process we use transfer learning by using a pre-trained network as a way to extract features from frames found in the video. By cutting off the last(classifier) layer of the pre-trained network we can extract high level features from our video data. A pre-trained network is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task. The model that we are going to use to extract features is VGG16. Frames from each video will pass through this pre-trained model. Then features from each frame will be aggregated to form the whole video data.

4.4.1 Architecture of the Video Feature Extracting Component

As we mentioned above the architecture we propose to classify the music videos consists of a CNN layer followed by RNN. The CNN layer is used to extract visual features and the RNN layer will help us extract the sequential features in the video i.e. the dance movement.

Due to the small amount of data we have to train the network, overfitting might be our number one challenge. In order to fight this overfitting, instead of training the network from scratch, we propose using a pre-trained network to extract features from the frames first then fed the extracted features to the RNN made of bi directional LSTM network. The RNN network will be followed by hidden fully connected Dense layer. Other than the use of transfer learning, we propose the use of dropout and L2 regularization to fight overfitting.

This proposed video feature extracting component has 3 modules each having distinct functionality. The first one is the CNN visual feature extracting module which main functionality is to extract visual features from each frame found in the video. The second module will aggregate the image feature to video feature and the last module which is called Sequence understanding LSTM module extracts movement features from the aggregated video features. The reason we divided the whole process into 3 independent processes is for efficiency and performance.

The first step is to extract visual features and store it in a memory to be used by the letter process. Next, image features will be read from memory and collected together to build the feature of the entire video. After that, the video features will be stored in a disk for a letter use by the final process. Then, the visual features of the video that reside on the disk will be loaded and sequential information will be extracted using LSTM. Following that

the extracted features will be fed to dense layers. Finally, the output from the dense layer will be given to the merging module so that it will be merged to the output of the audio feature extracting component. By partitioning the video feature extracting component in to three modules, we get the advantage of not running everything from scratch every time we need to configure or train the final model. High level architecture of video extracting module can be seen on Figure 20.

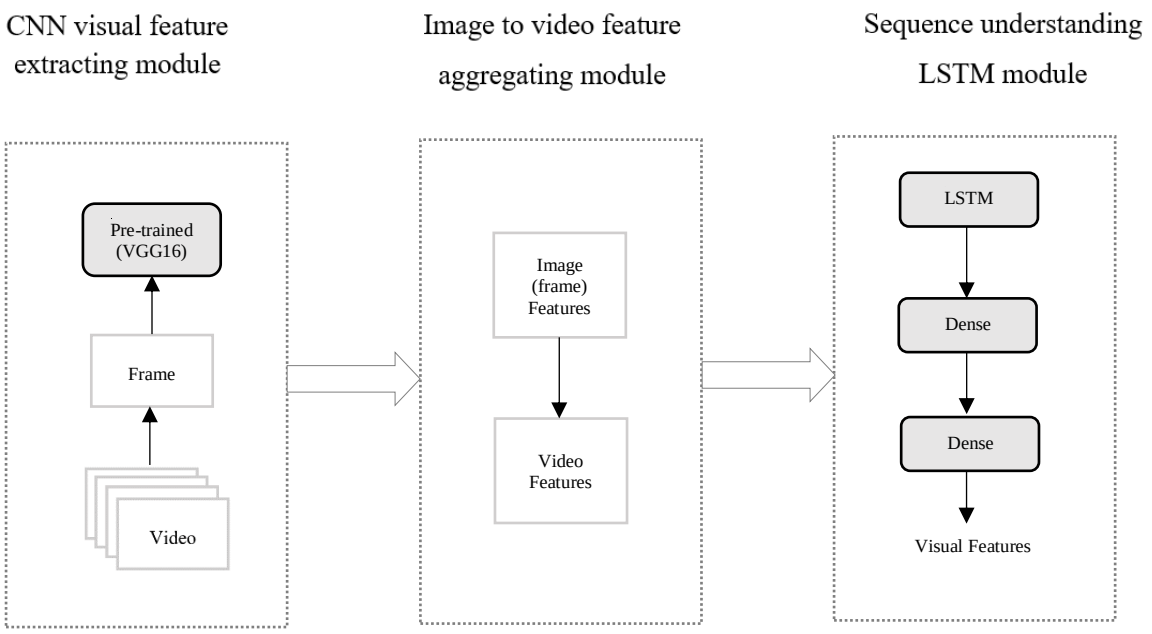


Figure 20 The Video Feature Extracting Component

4.4.2 Development of Video Feature Extracting Component

The component is implemented according to the high-level architecture shown in Figure 20. An input to the CNN feature extracting module (VGG16) is 100 by 100 pixel RGB image and the extracted feature output from this module is 3 by 3 by 512 3D tensor. VGG16 is a deep learning architecture trained on ImageNet dataset. It is composed of 2-dimensional convolution layer followed by 2D max-pooling layer. The detail of its architecture can be seen in Figure 22. This module will extract all the visual information found in each image and give it to the next layer to be aggregated to form the feature of the video. The 3 by 3 by 512 tensor features extracted from the first module is given to the image to video aggregating module. What this module does is take 45 consecutive 3x3x512 features and aggregate them as 1 video feature. We choose 45 because the video

is made up of 45 frames. Then it reshapes the feature and outputs a feature of size 45 by 4608 to be given to the next module.

The last module is made up of 1 Bi-directional LSTM layer with recurrent drop out of 0.2, another dropout layer of 0.5 in value and 2 fully connected dense layer with relu activation and an L2 regularizer of 0.001. From this last layer, we were able to extract 100 visual and sequential features from the video that can be concatenated with the features extracted from the audio feature extracting component. The detail architecture of LSTM module of ETM-VC can be found in Figure 21.

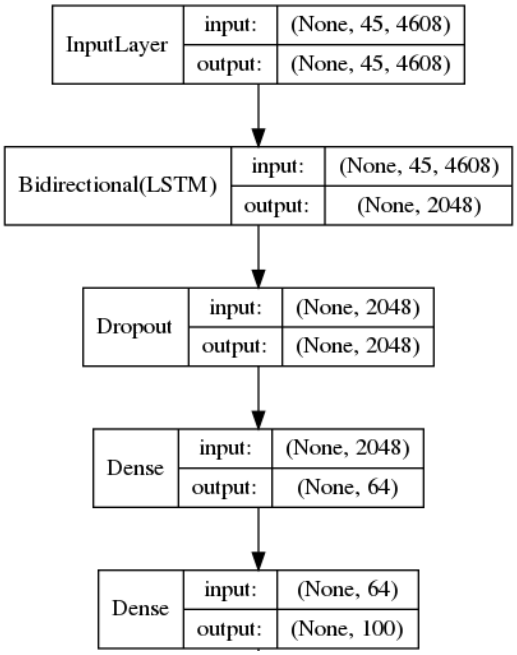


Figure 21 Sequence Understanding LSTM module of Video Feature Extracting Component

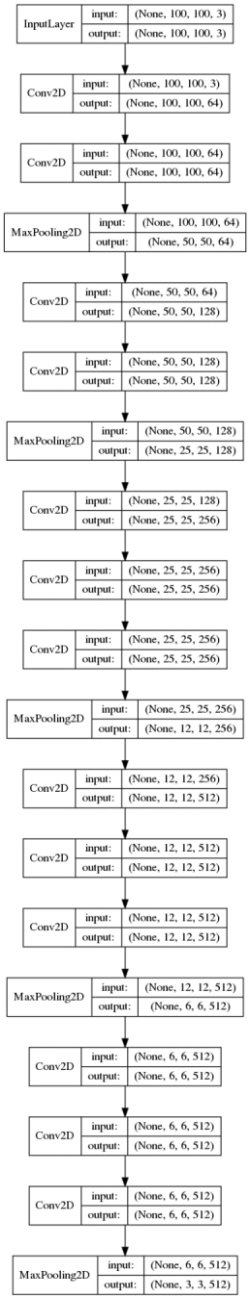


Figure 22 Architecture of VGG-16

4.5 Audio-Visual Feature Merging

Audio-visual feature merging is the concatenation of the audio features and visual features of the video which is performed by the audio-visual feature merging component. This component will take the features extracted from both the audio feature extracting component and video feature extracting component and merge them together. It will take 384 audio features and 100 visual features as an input and concatenate them to form 484 audio-visual features of the video as shown in Figure 23. These 484 audio-visual features will then be given to the predicting component.

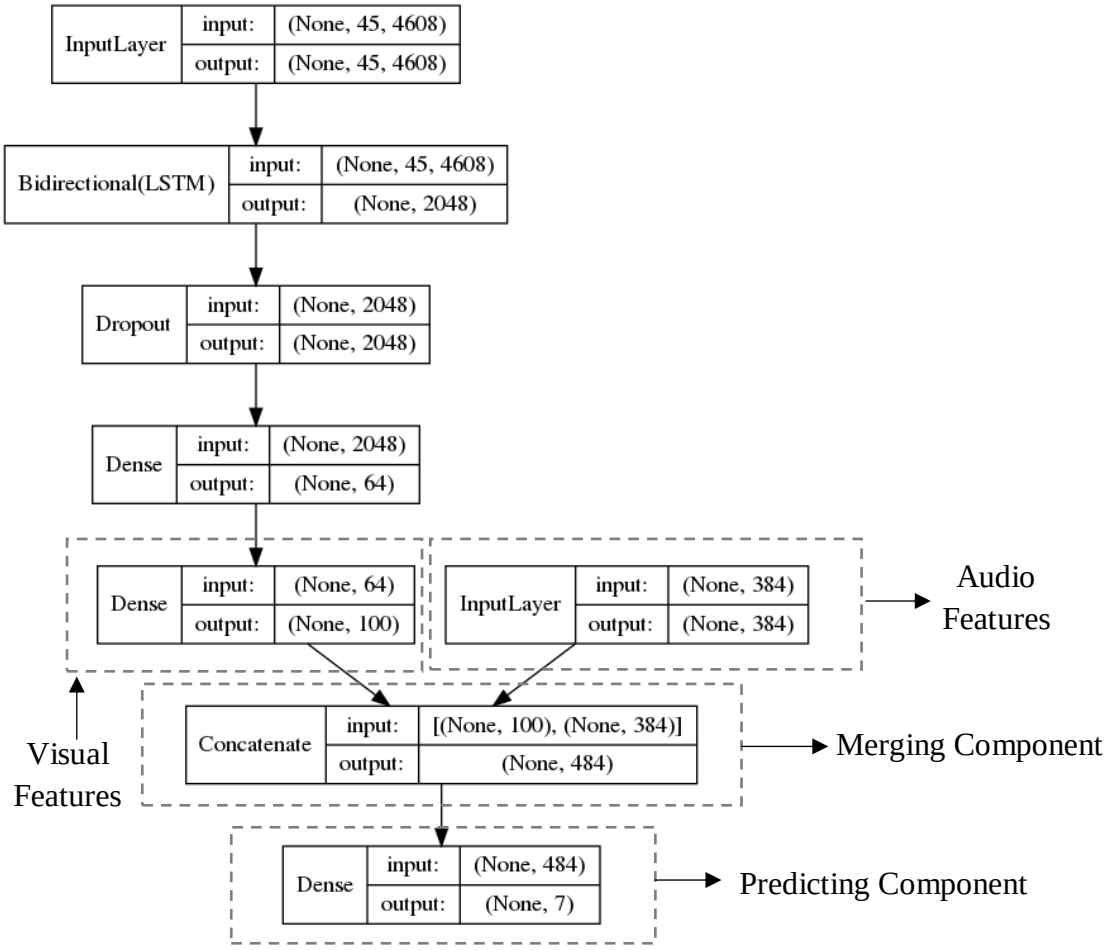


Figure 23 Merging and Predicting Component

4.6 Prediction

The last step in the whole process is prediction. Prediction is guessing the class of music it belongs to by using the features extracted from both the audio and video. Prediction is performed by the last component which is prediction module. The concatenated features

from the previous process, which are 484 in number are fed to this module which is made up of dense layer. The output of the dense layer will be one of the seven selected Ethiopian traditional music (Agew, Awi, Gondar, Harrer Oromo, Temben, Tigray, Wolayta). This module after predicting the class will adjust and propagate the error to the previous layers of the video extracting module so that each layer would change its configuration.

The entire model has 46,286,567 trainable parameters and it was compiles using RMSProp as an optimizer with a learning rate of 0.00001. The loss function that was used is categorical cross entropy and it was trained for 37 epochs using 16 as a batch size.

Chapter 5: Experimentation

This chapter discusses about data collection, tools used when developing the prototype and performance of the system. First, we will discuss about collection, preparation and preprocessing of the data used as an input to the proposed system, then the next Section will be about the tools used when developing the prototype. The final Section is about the performance of proposed system.

5.1 Data Collection

This section is about collection, preparation and preprocessing of dataset. We prepare our own dataset because we were not able to find the data we want for this research from publicly available sources or from previous researches. To the best of our knowledge, there was no previous research on the classification of Ethiopian traditional music using audio feature. Therefore, we had no other option but to prepare our own database that represents some of Ethiopian traditional songs to build the pre-trained audio classifier system.

Even though there were previous researches on the classification of Ethiopian traditional music using features from the video, the data that was collected by the researches were not adequate enough for the research that we are doing now. All of the previous researches focus on dance movements that is performed only by a single person and did not focus on dance movements that involves many people. Therefore, using their dataset would not help us to achieve the goal of the research.

The music data for this research was collected mostly from YouTube and personal music collections. We have collected a total of 354 music video that represents 10 Ethiopian traditional music. The 10 categories are, Agew, Awi, Gojjam, Gondar, Gurage, Harrer Oromo (Harrerge), Shoa Oromo, Temben, Tigray and Wolayta. The detail of the collected data is shown in Table 3. The number of categories (classes) used for the pre-trained audio feature extracting component is 10 while we reduced it to 7 for the final system that aggregates both audio and visual features due to of limitation of time and processing power of machine used for the research. Sample sequential frames and mel-spectrogram of audio used in this research can be seen in Appendix A and Appendix B.

After collection of the data, we had to manually classify the music to the proper class it belongs. This manual classification was performed and verified by collecting information from friends and families from different areas of the country.

Table 3 Number of Dataset Used in the Research

Class	Total Music Data	Audio (mp3)	Video Clip (mp4)	No. of frames
Agew	21	214	75	3375
Awi	36	340	64	2880
Gojjam	39	470	-	-
Gondar	42	506	67	3051
Gurage	60	611	-	-
Harrer Oromo	35	378	80	3600
Shoa Oromo	33	330	-	-
Temben	27	295	76	3420
Tigray	61	756	71	3195
Wolayta	25	245	70	3160
Total	354	4154	503	22,635

From Table 3 we can see that the number of audio and video used for the research is slightly higher than the total number of music video collected. This is because we trained the audio feature extracting component by using 30 seconds of audio data and we had to partition 1 song into many 30 seconds of data. The same is true for the video data, we did not use the whole video, instead we extracted 3 seconds of video clip from each video that shows the Ethiopian tradition dance.

5.1.1 Audio Data Preparation

All of the collected data were music videos in mp4 format. Therefore, we had to convert the data to an audio format (mp3) to use it for the pre-trained audio feature extracting module. In addition, we had to do removal of unwanted scenes to the music video because most music videos have production or record company advertisement at the beginning and sometimes making of the video at the end. Moreover, we had to split each audio data into 30 seconds of sound for the next step in the process. After doing this, the total number of audio data for train, validation and testing become 4154. The details can be seen on Table 3 and from the figure in Figure 24, we can see that the class distribution is skewed with more of the data falling in few classes.

After that we had to split the data in to train, validation and test. We used 2/3 of the data for train and 1/3 for test (hold out method). From the training data we had, 1/3 of the data was used for validation. The reason we did that was to know what kind of configuration of layers and hyper parameters works best for the problem at hand by seeing its performance on the validation data.

Then we extracted mel-spectrogram features using Librosa [65] library. An mp3 formatted audio data of 30 seconds that is quantized at 44KHz (samples per second) is fed for preprocessing step to be used as an input for feature extraction. The frame size (short audio clip) that was used for FFT was 2048 therefore making it STFT. A hop length (number of samples between frames) of 1024 was chosen and128 mel-spectral features were extracted. After that the data was shuffled and stored as a NumPy array on a disk for a latter use by the system. We did that for performance reason because we do not want to calculate the feature every time we train the neural network.

5.1.2 Video Data Preparation

Since the main input to the system is a video data, we had to prepare a video data that represents the traditional dance movements. We first selected video scenes of 3 seconds from the music clip collected. By doing this, the total number of data (video) became 503, representing 7 traditional dances of Ethiopia (more detail can be found from Table 3 and Figure 25). The 3 second video data is made up of 15 frames per second making the total frames in one video to be 45. So, we used a total of 22,635 images for train, test and validation data. The size of every video was reduced to 100-by-100 pixel for efficient processing of the data.

All the inputs and targets in neural network must be a tensor of floating point. Therefore, each frame in the video was converted to an RGB representation of the data. In order to be processed by the network efficiently or to make learning easier for the network, the RGB representation of each pixel must be normalized i.e. be in the range 0 – 1. Thus, we divided the RGB value by 255.

As for the audio of the video, we extracted 3 seconds audio from the 3 seconds video and turned it to 30 second audio by making use of repetition. We did this because we believe giving more sequential information to the system will give us better result than padding the array with empty value to make it 30 seconds long. This 30 second raw audio data will pass through all the steps we discussed in section 5.1.1 so that it can be fed to the model.

Of all the data we collected 70% of the data was used for training, 15% of the data for validation and the rest 15% for testing purpose.

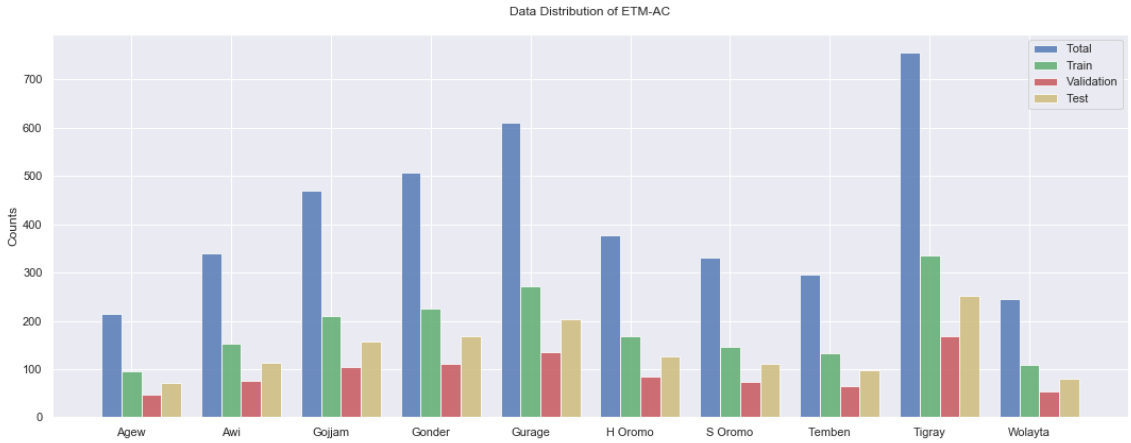


Figure 24 Graph Showing Number *of Audio Data Used for the Pre-trained Audio Feature Extracting Component*

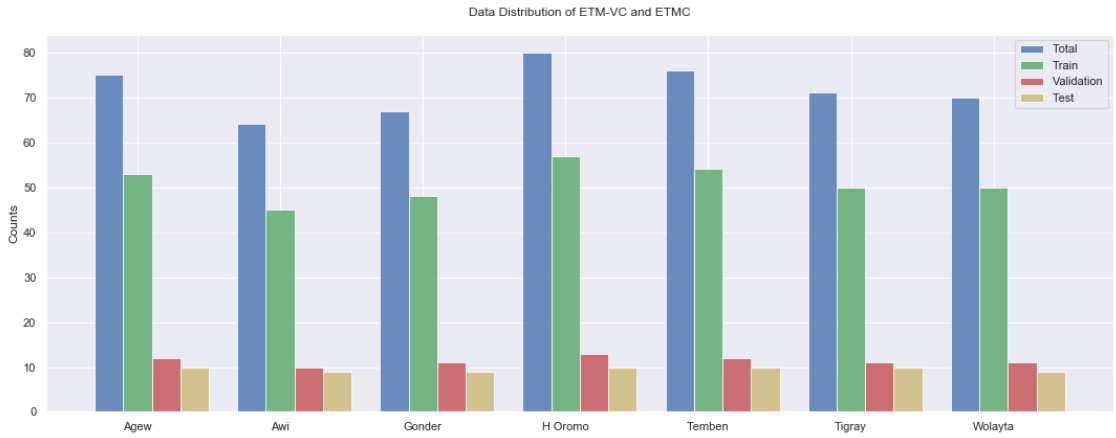


Figure 25 Graph Showing Number of Video Data Used for Video Data Only Classifier and for the Proposed System (ETMC)

5.2 Tools Used

In order to develop the system, the following tools (programming language, libraries and framework) were used:

a) Python 3 [66]: The programming language that is used to implement the models is python. We decided to use python because of the richness of libraries in data manipulation and frameworks in deep learning and data processing area.

b) Keras 2.3.1 [42]: It is a deep learning framework or a library providing high-level building blocks for developing deep learning models.

c) Scikit-learn 0.22 [67]: It is a machine learning library with various features and tools.

d) Jupyter Notebook [68]: Jupyter notebooks are great way to run deep-learning experiments. It allows you to break up a long experiment into smaller pieces that can be executed independently which makes the development interactive. All the experiments in this research were run in Jupyter.

e) NumPy 1.18.0 [69] – It is a multi-dimensional array (tensor) manipulation library. When doing deep learning every data must be represented in a tensor of different size and for storing and manipulating the arrays NumPy was used.

f) Librosa 0.7.1 [65]: It is a library for music and audio analysis and it provides the building blocks necessary to create music information retrieval systems. we use this library to extract features from the audio.

g) PyDub [70] – It is a library to manipulate audio data with a simple high-level interface.

h) OpenCV [71] – It is a library to solve computer vision problems. We use the library to read video data from disk and dismantle it to the image pieces that constitutes the video.

i) Moviepy 1.0.1 [72] – It is a video processing and editing library. We used it to extract the audio from the video.

j) Matplotlib 3.1.2 [73] – It is a python 2D plotting library.

k) Seaborn 0.9.0 [74] – It is a data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphs.

l) Graphviz [75] – It is a graph visualization software.

m) Pydot [76] – It is an interface to Graphviz and we use it to draw the graphical representation of the neural network architectures.

In addition to the above tools, we used two different Computers to train the models. The audio data processing component was trained on a computer with 4 GB RAM and core i3 processor that runs windows operating system whereas the video data processing component along with the final merging and predicting component was trained on a

computer with 16GB RAM and core i7 processor running Ubuntu operating system with 500GB SSD. For all the experiments GPU was not used.

Sample source code and model training process can be seen in Appendix C.

5.3 System Evaluation

This section is about the performance of the proposed system. There are different evaluation matrices and here we have provided some to show how the proposed system perform with previously unseen data (test data). We used the hold-out method in which we hold some data for a latter evaluation purpose. The evaluation matrix that we present here are accuracy, precision, recall, F1 score, confusion matrix, P/R AUC, micro average and weighted average.

We wanted to see if the audio only features or video only features can classify Ethiopian traditional music by itself and to what extent. Therefore, in addition to the proposed system, we build an audio only classifier and video data only classifier system. As discussed above, we also use the audio only feature classifier as a pre-trained network to extract the audio feature in the ETMC.

5.3.1 Evaluation of Audio Data Only Classifier

The audio data only classifier was trained for 60 epochs using 16 as a batch size. Moreover, it was compiled using RMSProp as an optimizer with learning rate 0.0005 and loss function categorical cross entropy. In addition, its architecture is the same as the audio feature extracting module of the ETMC.

After the training process has finished, the final saved best model was taken and used to predict the label (class) of the previously unseen test data. The accuracy that was achieved was 85% where some class performed over 90% and some less than 60%. The class that performs well on the classification was Awi (F1 score of 100%) followed by Tigray and Gondar (F1 score of 97%) and the one with least performance was Agew (F1 score of 62%). As we can see from Figure 24, the train data is not distributed evenly making the total number of the song of Tigray the highest and Agew the least therefore we can conclude that the class imbalance may have affected the performance of the system. Detail of classification report of the audio data only classifier is shown on Table 4.

Table 4 Classification Report of Audio Data Only Classifier

	Precision	Recall	F1-Score	Support
Agew	0.54	0.71	0.62	70
Awi	1.00	1.00	1.00	112
Gojjam	0.76	0.72	0.74	156
Gondar	0.96	0.99	0.97	168
Gurage	0.89	0.91	0.90	203
Harrer Oromo	0.81	0.70	0.75	125
Shoa Oromo	0.66	0.73	0.70	109
Temben	0.80	0.68	0.73	97
Tigray	0.95	1.00	0.97	251
Wolayta	0.92	0.74	0.82	81
Accuracy			0.85	1372
Macro Avg	0.82	0.82	0.82	1372
Weighted Avg	0.85	0.85	0.85	1372

We can also visually assess the performance of the system using a confusion matrix that is shown in Figure 26. Since the test data we had for some classes was more than the other (skewed data), the color code (on the diagonal line) would not give us much information other than the total number of correct data classified under it.

From the confusion matrix we can see that most of the cells are filled up with zero values meaning that the falsely labeled songs are very low. Another interesting thing we can see in the confusion matrix is that most of the falsely labeled Gojjam songs are actually Agew songs and most of the falsely labeled Agew songs are actually Gojjam. In addition, we get an interesting result that shows the confusion label of the model between the two Oromo songs (Harrer Oromo and Shoa Oromo) and Gojjam. Some of the falsely labeled Oromo songs actually belongs to Gojjam and some of the wrongly labeled Gojjam songs belongs to one of the two Oromo Songs.

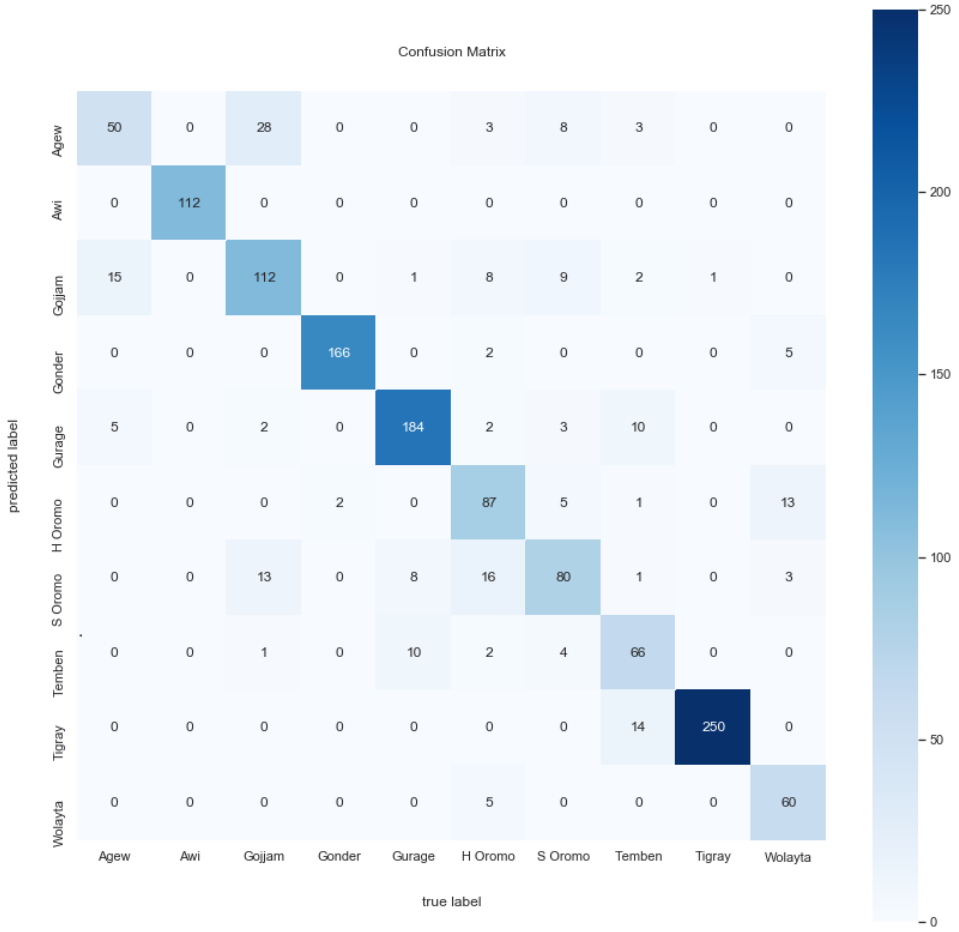


Figure 26 Confusion Matrix of Audio Data Only Classifier

We have also presented another view of the performance of the system using P/R curve (Precision Recall curve) to show how one class performed as compared to the other classes. The P/R curve of the classifier along with AUC P/R is shown in Figure 27.

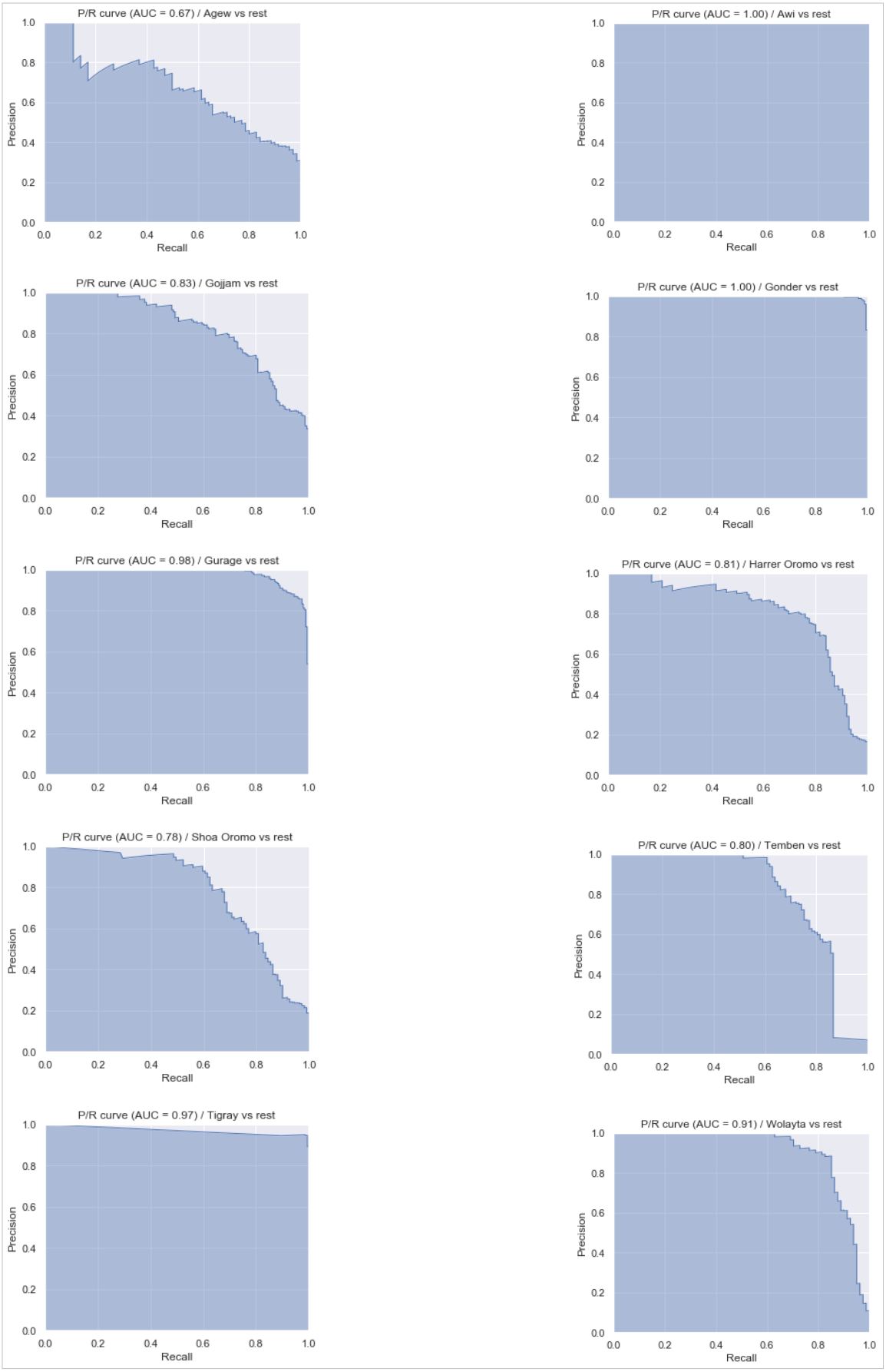


Figure 27 P/R Curve for Audio Data Only Classifier

5.3.2 Evaluation of Video Data Only Classifier and Ethiopian Traditional Music Classifier

The video data only classifier was trained for 42 epochs using 16 as a batch size. In addition, it was compiled using RMSProp as an optimizer with learning rate 0.00001 and loss function categorical cross entropy. Moreover, its architecture is the same as the video feature extracting module of the ETMC.

After the training process is finished for the video data only classifier, we use the best model to evaluate the system on the test data. The accuracy that was achieved was 78% which was lower than the audio data only classifier. But comparing audio data only with video data only classifier will not give us the correct information since we train audio data only classifier on a vast amount of data as compared to video data only classifier. Comparing the video data only classifier with ETMC would make more sense since both of were trained on the same dataset and ETMC is an addition to video data only classifier by incorporating the audio feature of the video. Therefore, it is better if we see their performance side-by-side.

In the video data only classifier, few of the classes score more than 90% (only Agew and Awi) whereas the rest score in between 67% and 78 %. Since the train data is almost evenly distributed for this model, class imbalance will not be a factor determining the performance of this model.

ETMC was trained for 37 epochs and scored an accuracy of 85% which is a lot better than video data only classifier but for some categories the F1 score measure have decreased (Agew and Temben) in ETMC or remain the same (Harrer Oromo). Details of the classification report can be seen on Table 5.

Table 5 Classification Report of Video Data Only Classifier (Left) and ETMC (Right)

	Precision	Recall	F1-Score	Support
Agew	0.83	1.00	0.91	10
Awi	1.00	0.89	0.94	9
Gondar	0.70	0.78	0.74	9
Harrer Oromo	0.69	0.90	0.78	10
Temben	0.86	0.60	0.71	10
Tigray	0.75	0.60	0.67	10
Wolayta	0.67	0.69	0.67	9
Accuracy			0.78	67
Macro Avg	0.79	0.78	0.77	67
Weighted Avg	0.79	0.78	0.77	67

	Precision	Recall	F1-Score	Support
Agew	0.82	0.90	0.86	10
Awi	1.00	1.00	1.00	9
Gondar	1.00	0.89	0.94	9
Harer Oromo	0.69	0.90	0.78	10
Temben	0.75	0.60	0.67	10
Tigray	0.82	0.90	0.86	10
Wolayta	1.00	0.78	0.88	9
Accuracy			0.85	67
Macro Avg	0.87	0.85	0.85	67
Weighted Avg	0.86	0.85	0.85	67

The confusion matrix shown on Figure 28 shows the visual information of the performance of both systems and the highest improvement is seen in the song of Tigray people.

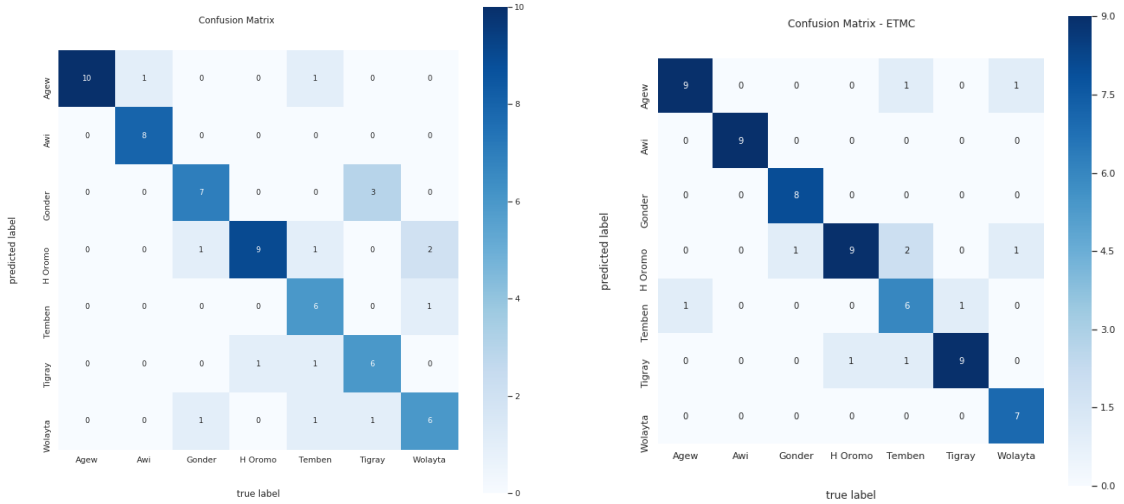


Figure 28 Confusion Matrix of Video Data Only Classifier (Left) and ETMC (Right)

We have also presented the P/R curve and AUC for the P/R curve to show how one class performes as compated to others for both the Video Data Only Classifer Figure 29 and ETMC Figure 30.

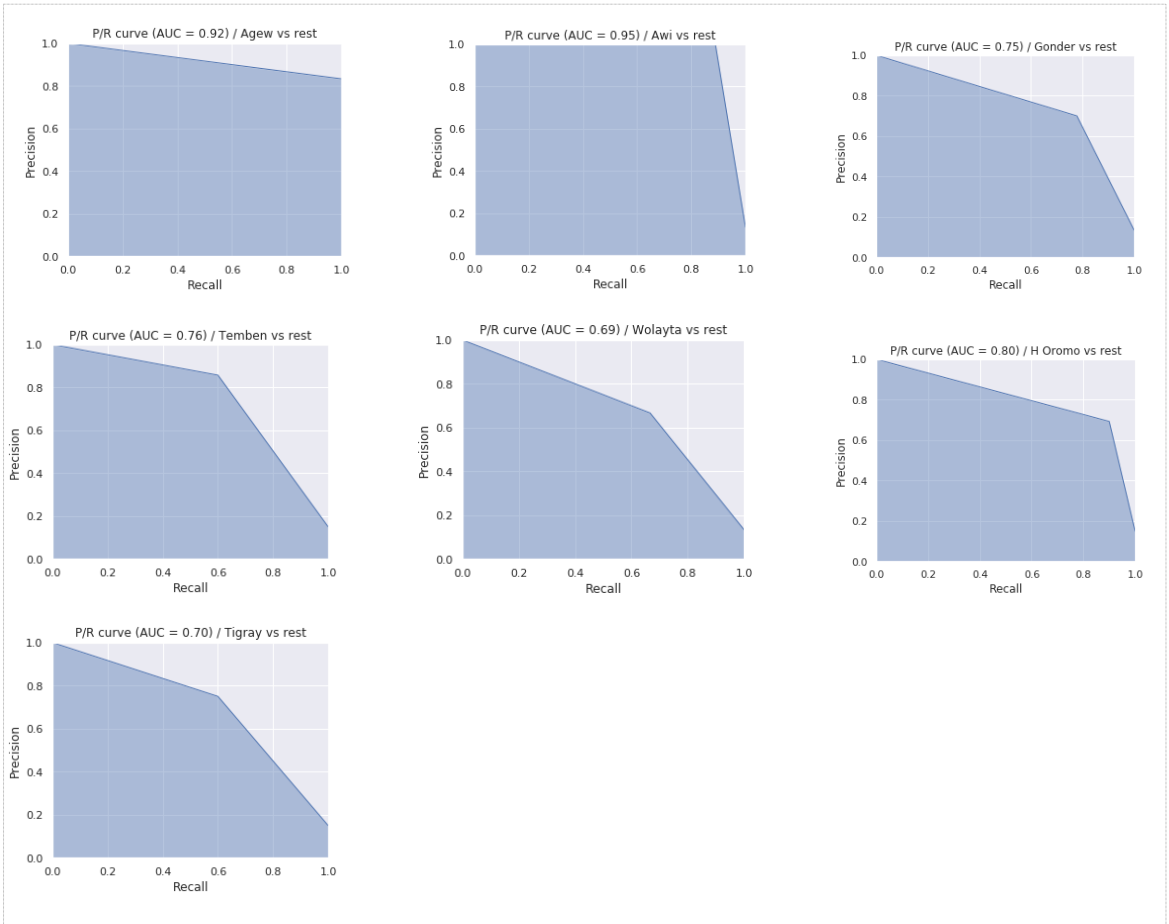


Figure 29 P/R Curve for Video Data Only Classifier

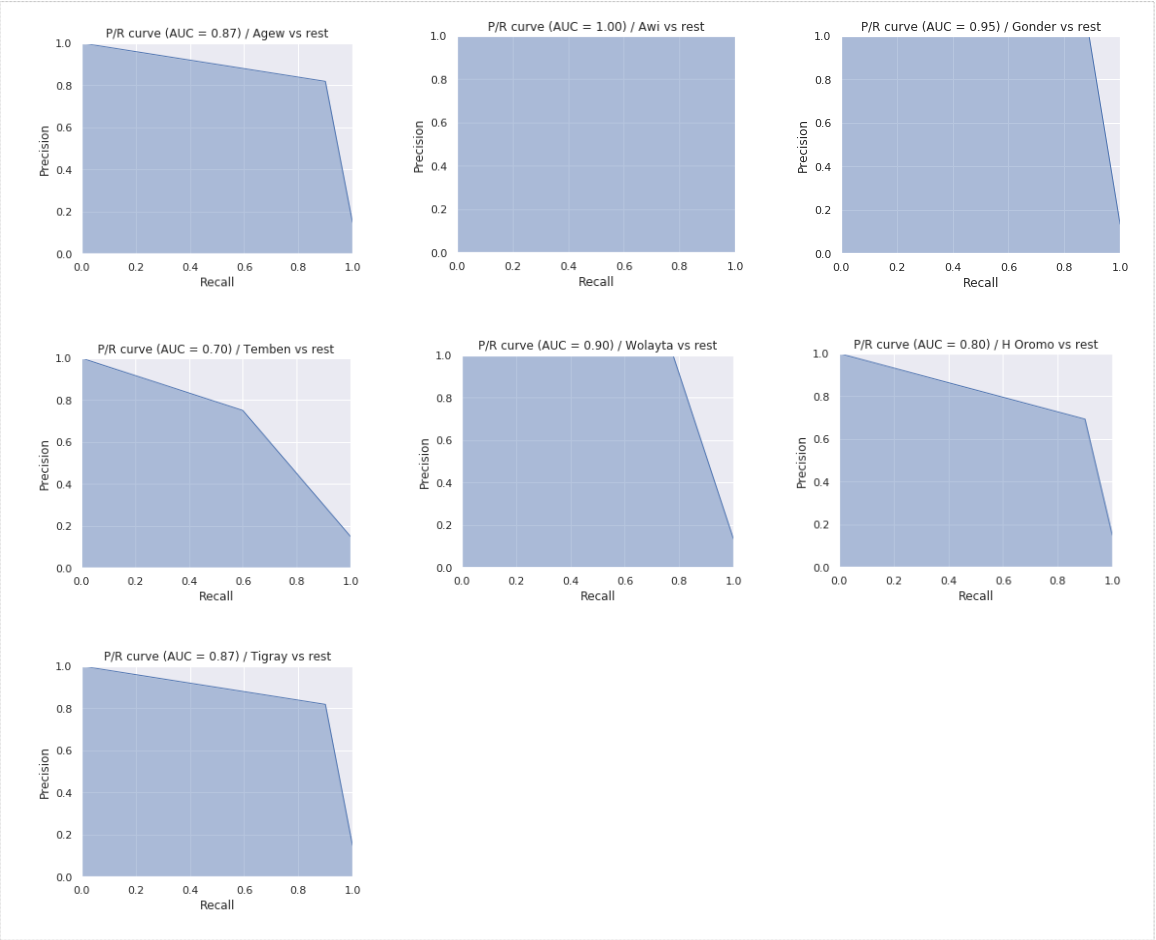


Figure 30 P/R Curve for ETMC

5.4 Discussion

From the experiment we did, we have achieved good results and it is worth mentioning some points. Mel-spectrogram of an audio as a feature can give good result when classifying Ethiopian traditional music. We also believe by adding more training data we can even increase the classification accuracy of the system to a larger number. Moreover, the use of neural network architecture that combines CNN and RNN for Ethiopian music retrieval problems can give good results. The other thing we have noticed from this experiment is that by adding the audio feature to the visual data only classifier the accuracy was improved. Not only that but we have shown even with small number of training data, we can get the same accuracy to the model that was trained with much larger data (audio data only classifier) by combining the audio-visual feature. Therefore, it would not be wrong if we say the fusion of this audio-visual information will lead to a greater accuracy.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

Music is a very ancient and highly valued feature of all known living cultures that shows many aspects of daily life. Storage and format of music passes through many eras, from polyvinyl chloride to cassette then from CD to digital MP3 and more recently streaming. This revolution in the music distribution and storage brought by digital technology has fueled tremendous interest in and attention to the way information is retrieved to this kind of content. It resulted in the rapid growth of digitally available music data that requires novel technologies to allow users to browse personal collections or discover new music on the internet.

Ethiopia is an East African country with a population estimated around 109 million (in 2018). It is the land of many nation, nationalities and people each having their own distinct music and dance style. It is believed that more than 80 ethnic group exists in the country making it very hard to distinguish one type of music from another.

Every data, be it image, video or audio is represented using a digital signal in a computer. Signal is a representation of how one parameter is related to another parameter. For example, in the case of an audio, a signal represents how the sound pressure varies with time. Any signal in digital format needs to be processed or manipulated before fed to any kind of learning algorithm by a digital signal processing.

Machine learning or deep learning came to learn the data-processing rules automatically rather than manually crafted by a programmer, i.e. they learn from the data itself. Deep learning is a subfield of machine learning in which not only it learns the data-processing rules automatically but also the representation of the data. Therefore, deep learning shortens the tedious task in machine learning which is feature engineering by automating that process. One of the building blocks of deep learning is a layer and we have many deep learning layers and convolution layer and recurrent layer being among them. Convolution layers use convolution in place of general matrix when doing multiplication but not great when processing sequential information. On the other side recurrent layers process sequential data.

Classification is a supervised way of learning where objects are assigned certain categories based on the features they have and the main aim of this research is to classify Ethiopian

traditional music to their locality using audio-visual features and deep learning. To achieve that we proposed a deep learning neural network architecture that is composed of CNN and RNN. This architecture is composed of audio feature extracting module, video feature extracting module, merging module and predicting module.

The input to the audio feature extracting module is an audio feature called mel-spectrogram. Mel-spectrogram can be considered as visual representation of an audio signal. These features will pass through a custom made pre-trained neural network that consists of parallel CNN and RNN so that both visual and sequential information from the mel-spectrogram of the audio data can be extracted. This pre-trained network was designed for this research as a way to facilitate the extraction of audio data features from the video. The extracted feature from both CNN and RNN will be concatenated using a merging layer. These concatenated features will then again be merged with the visual features by a latter component.

The input to the video feature extracting component is an RGB image or frames of a video. One video consists of 45 frames and each image will be considered as independent data by the first module and then latter on aggregated to form the feature of the video. The images will pass through convolutional visual feature extracting module. This module is made up of a pre-trained network on ImageNet called VGG-16 by using transfer learning. The extracted visual features from the image will be aggregated by the next module to form the features of 1 whole video. Features of the video will then be fed to the sequence understanding LSTM module. This module will extract movement (dance) features from the given feature and give it to the next merging component that merges visual features with audio features. After the features are merged they are given to the predicting module so that the class of the video is known.

The classifier designed using the proposed architecture has 46,286,567 trainable parameters and it was compiles using RMSProp as an optimizer with a learning rate of 0.00001. The loss function that was used was categorical cross entropy and it was trained for 37 epochs using 16 as a batch size.

Data for this research was collected from YouTube and personal collection. The collected data passes through pre-processing steps and given to the neural network architecture so that the model could be trained. A total of 345 music video of 10 classes (Agew, Awi, Gojjam, Gonder, Gurage, Harrer Oromo, Shoa Oromo, Temben, Tigray, Wolayta) were

collected. To develop the prototype, we have used python programming language and keras deep learning framework with TensorFlow as a back-end. In addition, we have used Jupyter Notebook to run all the experiments.

We did 3 experiments to measure the performance of the proposed system. One using only audio features, one using only visual features and lastly using both audio and video features. Using only audio features the performance of the classifier was 85% whereas using only visual features we were able to get an accuracy of 78%. But by adding audio feature to the video only classifier, we were able to increase the accuracy by 7 units making the final performance of the proposed system 85%.

6.2 Contribution of the Research

This research has the following contributions:

- The architecture is designed to work well where we do not have very large data to train the model. It consists of a pre-trained network called VGG-16 to extract visual information and custom pre-trained network to extract audio features from the video. The custom pre-trained network was trained on a very large Ethiopian traditional songs and this pre-trained network was used as an audio feature extracting module in the proposed architecture. We decided to use a custom made pre-trained network for audio feature extraction because it was not possible to get large amount of video data so we need a way to use transfer learning and since we could not find an already trained pre-trained network for audio data we designed our own pre-trained network that was trained on a relatively large amount of audio data.
- In addition, the architecture was designed to work well when processing power and available memory of the machine to train the model is very low. We partitioned the end to end process into independent components to minimize the processing power and memory used by the whole component. We extracted audio features and part of the visual features prior to the model training. By doing that we decreased the time it might have taken to train the model because the features have already been extracted beforehand. If we have not done that it might have taken months to train the network using the processing power of the computer we have (core i7 with no GPU support). In addition, we might have encounter the problem of not having

enough memory because the memory that was available by the machine was only 16GB.

- Furthermore, we showed that the fusion of audio and visual features will lead to an increase in accuracy of the video data only classifier. We extracted all features related to appearance of the dancers such as shape, color, outfit and movement by using both CNN and RNN. This will make our research the first one that uses appearance, movement and audio feature form video to classify Ethiopian traditional music. Moreover, we showed that it is possible to use dataset similar to real time environment (data) such as a video that does not have plane background and has more than 1 dance performer which was not possible in the previous Ethiopian traditional dance classifier researches [35, 36, 37].
- Lastly, this work showed that it is possible to classify Ethiopian traditional music using the newly emerging state of the art deep learning algorithm which reduces the time it would have taken extracting features manually.

6.3 Future Work

We have achieved good results in this research but that does not mean it could not be improved. To increase the accuracy or performance of the system we recommend trying different approaches such as:

- Applying data augmentation techniques on the audio signal such as addition of a noise, using different loudness range, time stretching and pitch shifting [77].
- Adding textual information (features) other than audio and video such as meta data found in the music itself such as artist name could give us additional information.
- Using a pre-trained network to do fine-tuning of hyperparameters besides feature extraction.
- Applying data augmentation technique to the frames of the video.
- Using more representational data and complex network structure such as 3D CNN that learns the visual and temporal feature from the vided the same time.
- Adding optical-flow features to the network might give us better performance.

References

[1] A. Killin, "The origins of music: Evidence, theory, and prospects," *Society for Education, Music and Psychology Research: Music & Science*, vol. I, pp. 1-23, 2018.

[2] R. J. Burgess, The History of Music Production, New York: Oxford University Press, 2014.

[3] D. a. B. C. L. a. A. I. Cohen, Music: Its Language, History, and Culture, CUNY Academic Works, 2015.

[4] R. Casati and J. Dokic, "Sounds," 10 April 2020. [Online]. Available: http://plato.stanford.edu/entries/sounds. [Accessed 31 May 2020].

[5] C. Taintor, "Chronology: Technology and the music industry," 2004. [Online]. Available: http://www.pbs.org/wgbh/pages/frontline/shows/music/inside/cron.html. [Accessed 31 May 2020].

[6] B. Fly, "How Does Music Consumption Impact the Music Industry and Benefit Artists?," Accounting Undergraduate Honors Theses, 2016.

[7] "iTunes," [Online]. Available: https://www.apple.com/itunes. [Accessed 31 May 2020].

[8] "Spotify," [Online]. Available: https://www.spotify.com. [Accessed 31 May 2020].

[9] "Tidal," [Online]. Available: https://tidal.coom. [Accessed 31 May 2020].

[10] "Apple Music," [Online]. Available: https://www.apple.com/apple-music. [Accessed 31 May 2020].

[11] "YouTube Music," [Online]. Available: http://www.music.youtube.com. [Accessed 31 May 2020].

[12] "Pandora," [Online]. Available: http://www.pandora.com. [Accessed 31 May 2020].

[13] "Amazon Music," [Online]. Available: https://music.amazon.com. [Accessed 31 May 2020].

[14] C. Willman, "RIAA Study Shows 2019 Music Revenue Increased 13%, Subscription Incode Up 25%," 26 02 2020. [Online]. Available: https://money.yahoo.com/riaa-study-shows-2019-music-004624993.html. [Accessed 31 May 2020].

[15] "How cities benefit from helping the music industries grow," 09 2015. [Online]. Available: https://www.wipo.int/wipo_magazine/en/2015/05/article_0009.html. [Accessed 31 May 2020].

[16] G. Dougherty, Pattern Recognition and Classification, New York: Springer Science+Business Media, 2013.

[17] G. Martin, "Dance Types in Ethiopia," *Journal of the International Folk Music Council*, vol. 19, pp. 23-27, 1967.

[18] E. Abate, "Ethiopian Kiñit (scales) Analysis of the formation and structure of the Ethiopian scale system," in *Proceedings of the 16th International Conference of Ethiopian Studies*, 2009.

[19] "Introducing Ethiopian folk dance," [Online]. Available: http://saba.air-nifty.com/mocha_ethiopia_dance_e/introducing-ethiopian-fol.html. [Accessed 31 May 2020].

[20] Y. Abebe, "A Study of Ethiopian Traditional Music," Unpublished, 2007.

[21] S. Forsyth, "Ethiopian FOCUS Music," [Online]. Available: https://docplayer.net/59644487-Ethiopian-focus-music.html. [Accessed 31 May 2020].

[22] "Ethiopian Dance and Music," [Online]. Available: https://rainbowftf.ngo/the-ethiopians/ethiopian-dance/. [Accessed 31 May 2020].

[23] C. C. Aggarwal, "Audio Visual Fusion," in *Data Classification Algorithms and Applications*, CRC Press Taylor & Francis Group, 2015, p. 345.

[24] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," *IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING*, vol. 10, pp. 293-302, July 2002.

[25] J.-M. Ren, Z.-S. Chen and J.-S. R. Jang, "On the Use of Sequential Patterns Mining as Temporal Features for Music Genre Classification," *IEEE,* pp. 2294-2297, 2010.

[26] M. M. Misron, N. Rosli, N. A. Manaf and H. A. Halim, "Music Emotion Classification (MEC): Exploiting Vocal and Instrumental Sound Features," *Recent Advances on Soft Computing and Data Mining. Advances in Intelligent Systems and Computing*, vol. 287, pp. 539-549, 2014.

[27] R. Sarkar, S. Dutta, A. Roy and S. K. Saha, "Emotion Based Categorization of Music Using Low Level Features and Agglomerative Clustering," *Computer Vision, Pattern Recognition, Image Processing, and Graphics. NCVPRIPG 2017. Communications in Computer and Information Science,* vol. 841, pp. 506-516, 26 April 2018.

[28] F. Zhang, H. Meng and M. Li, "Emotion extraction and recognition from music," in *12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, Changsha, China, 2016.

[29] T. Starner and A. Pentland, "Real-time American Sign Language recognition from video using hidden Markov models," in *Proceedings of International Symposium on Computer Vision - ISCV*, Coral Gables, FL, USA, USA, 1995.

[30] J. F. Lichtenauer, E. A. Hendriks and M. J. Reinders, "Sign Language Recognition by Combining Statistical DTW and Independent Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, no. 11, pp. 2040 - 2046, 2008.

[31] M. Panwar and P. S. Mehra, "Hand gesture recognition for human computer interaction," in *2011 International Conference on Image Information Processing*, Shimla, India, 2011.

[32] R. K. Tripathi, A. S. Jalal and S. C. Agrawal, "Suspicious human activity recognition: a review," *Artificial Intelligence Review,* vol. 50, no. 2, p. 283–339, 2018.

[33] K. V. V. Kumar, P. V. V. Kishore and D. A. Kumar, "Indian Classical Dance Classification with Adaboost Multiclass," *Hindawi Mathematical Problems in Engineering*, vol. 2017, 2017.

[34] I. Kapsouras, S. Karanikolos, N. Nikolaidis and A. Tefas, "Folk Dance Recognition using a Bag of Words Approach and ISA/STIP Features," in *Proceedings of the 6th Balkan Conference in Informatics*, Thessaloniki, 2013.

[35] A. Mekonnen, S. K. Selvi, V. S. Kumar and B. Tesfaye, "Optical Flow Based Ethiopian Traditional Dance Video," *GESJ: Computer Science and Telecommunications,* pp. 69-79, 2016.

[36] S. K. Selvi and A. Mekonnen, "Content Based Classification of Ethiopian Traditional Dance Videos Using Optical Flow and Histogram of Oriented Gradient," *International Journal of Innovations in Engineering and Technology (IJIET),* vol. 6, no. 3, pp. 371-380, February 2016.

[37] F. T. Tesisa, *Ethiopian Traditional Dance Classification Using Visual Features*, Msc. Thesis, University of Gondar, Unpublished, 2013.

[38] S. Ji, W. Xu, M. Yang and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, no. 1, pp. 221-231, 2013.

[39] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," *Advances in Neural Information Processing Systems,* p. 568–576, 2014.

[40] C. Li, S. Sun, X. Min, W. Lin, B. Nie and X. Zhang, "End-to-End Learning of Deep Convolutional Neural Network for 3D Human Action Recognition," in *IEEE International Conference on Multimedia and Expo Workshops (ICMEW*, China, Hong Kong, 2017.

[41] E. P. Ijjina and C. K. Mohan, "Human action recognition based on recognition of linear patterns in action bank features using convolutional neural networks," in *International Conference on Machine Learning and Applications*, 2014.

[42] "Keras," [Online]. Available: https://keras.io/. [Accessed 31 May 2020].

[43] "Ethiopia Overview," World Bank, 26 September 2019. [Online]. Available: https://www.worldbank.org/en/country/ethiopia/overview. [Accessed 31 May 2020].

[44] "Traditional Ethiopian Music and Ethiopian Culture," 2010 7 10. [Online]. Available: https://tigray.net/2010/10/07/traditional-ethiopian-music-and-ethiopian-culture/. [Accessed 31 May 2020].

[45] "Oromia Dance of Ethiopia," 03 August 2015. [Online]. Available: https://indianvagabond.com/2015/08.03/oromian-dance-of-ethiopia/. [Accessed 31 May 2020].

[46] S. W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, San Diego: California Technical Publishing, 1999.

[47] M. Müller, Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications, Switzerland: Springer International Publishing, 2015.

[48] A. Bovic, Handbook of Image and Video Processing, Elsevier Academic Press, 2005.

[49] S. Gong, C. Liu, Y. Ji, B. Zhong, Y. Li and H. Dong, Advanced Image and Video Processing Using Matlab, Springer International Publishing AG, 2019.

[50] A. M. Tekalp, Digital Video Processing, Prentice Hall PTR, 1995.

[51] F. Chollet, Deep Learning with Python, New York: Manning Publications Co, 2018.

[52] A. W. Trask, Grokking Deep Learning, New York: Manning Publications, 2019.

[53] T. Mitchell, *Machine Learning*, New York: McGraw-Hill, 1997.

[54] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.

[55] W. Wu, G. Song, F. Han and Z. Wang, "Music Genre Classification Using Independent Recurrent Neural Network," *IEEE*, pp. 192-195, 2018.

[56] G. Song, Z. Wang, F. Han and S. Ding, "Transfer Learning for Music Genre Classification," *Intelligence Science I ICIS*, vol. 510, 2017.

[57] J. Han, M. Kamber and J. Pei, "Metrics for Evaluating Classifier Performance," in *Data Mining Concepts and Techniques*, Elsevier Inc, 2012, pp. 364 - 370.

[58] L. P. Coelho and W. Richert, Building Machine Learning System with Python, Birmingham - Mumbai: Packet Publishing, 2015.

[59] Z. Qiu, J. Sun, M. Guo, M. Wang and D. Zhang, "Survey on Deep Learning for Human Action Recognition," in *International Conference of Pioneer Computer Scientists, Engineers and Educators ICPCSEE*, Guilin, 2019.

[60] S. Panwar, A. Das, M. Roopaei and P. Rad, "A Deep Learning Approach for Mapping Music Genres," in *System of Systems Engineering Conference*, Waikoloa, HI, USA, 2017.

[61] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *ICLR*, 2015.

[62] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla and M. Bernstein, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, 2015.

[63] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Neural Information Processing Systems*, 2012.

[64] B. . C. Moore, An introduction to the psychology of hearing, Brill, 2012.

[65] "Librosa," [Online]. Available: https://librosa.github.io/. [Accessed 31 May 2020].

[66] "Python3," [Online]. Available: https://www.python.org/. [Accessed 31 May 2020].

[67] "scikit-learn," [Online]. Available: https://scikit-learn.org/. [Accessed 31 May 2020].

[68] "Jupyter Notebook," [Online]. Available: https://jupyter.org/. [Accessed 31 May 2020].

[69] "numpy," [Online]. Available: https://numpy.org/. [Accessed 31 May 2020].

[70] "PyDub," [Online]. Available: https://pypi.org/project/pydub/. [Accessed 31 May 2020].

[71] "OpenCV," [Online]. Available: https://opencv.org/. [Accessed 31 May 2020].

[72] "MoviePy," [Online]. Available: https://pypi.org/project/moviepy/. [Accessed 31 May 2020].

[73] "matplotlib," [Online]. Available: https://matplotlib.org/. [Accessed 31 May 2020].

[74] "seaborn," [Online]. Available: https://seaborn.pydata.org/. [Accessed 31 May 2020].

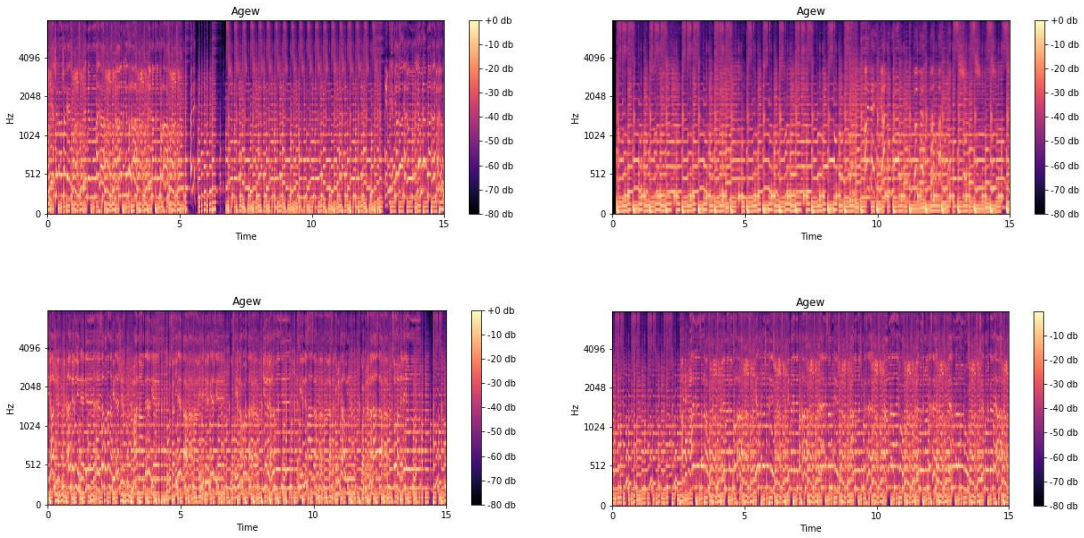
[75] "graphviz," [Online]. Available: https://www.graphviz.org/. [Accessed 31 May 2020].

[76] "PyDot," [Online]. Available: https://pypi.org/project/pydot/. [Accessed 31 May 2020].

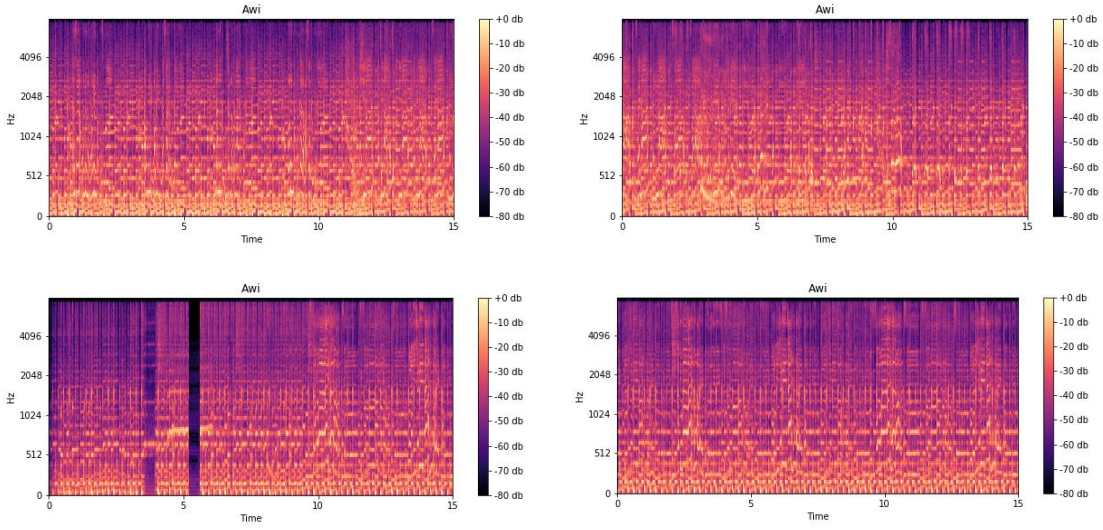
[77] R. L. Aguiar, Y. M. G. Costa and C. N. Silla, "Exploring Data Augmentation to Improve Music Genre Classification with ConvNets," in *International Joint Conference on Neural Networks (IJCNN)*, 2018.

Appendix A: Sample Mel-Spectrogram of the Audio Data

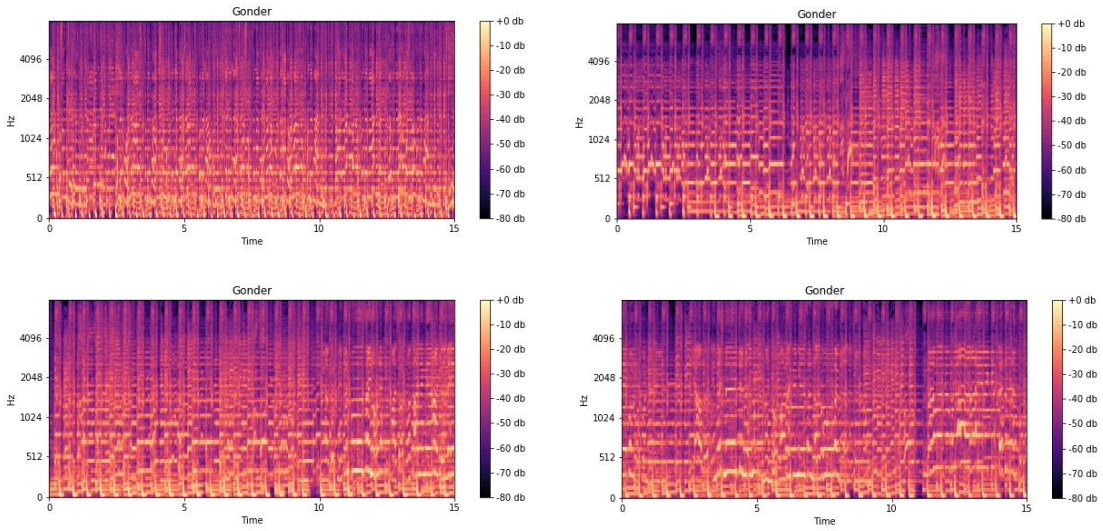
A.1 Agew Wag Himra



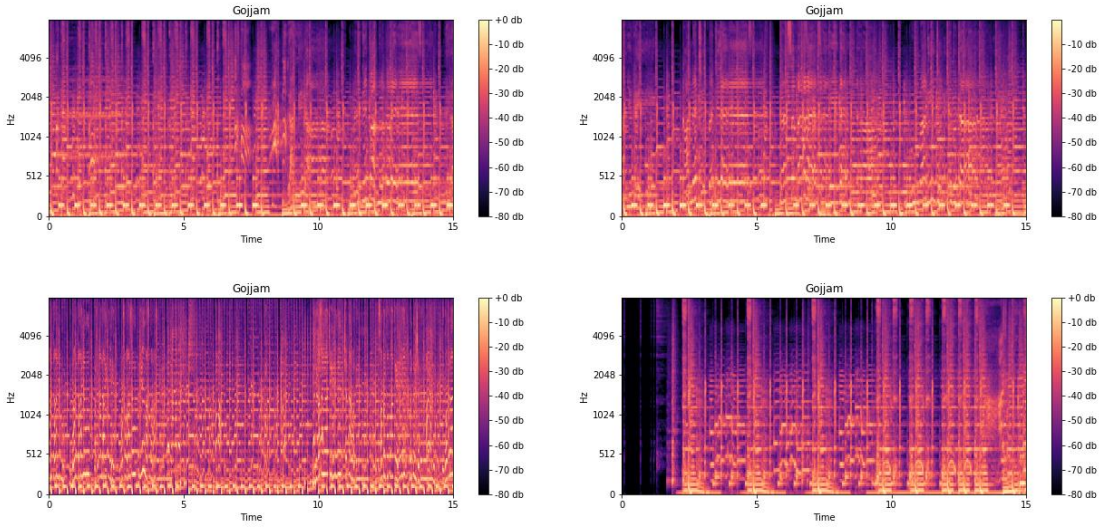
A.2 Awi Agew



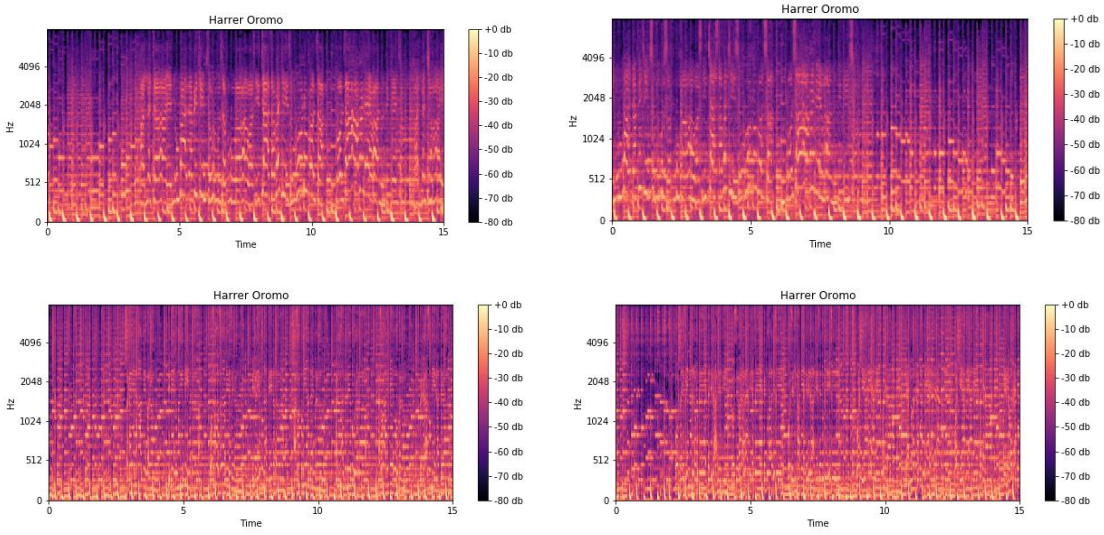
A.3 Gondar



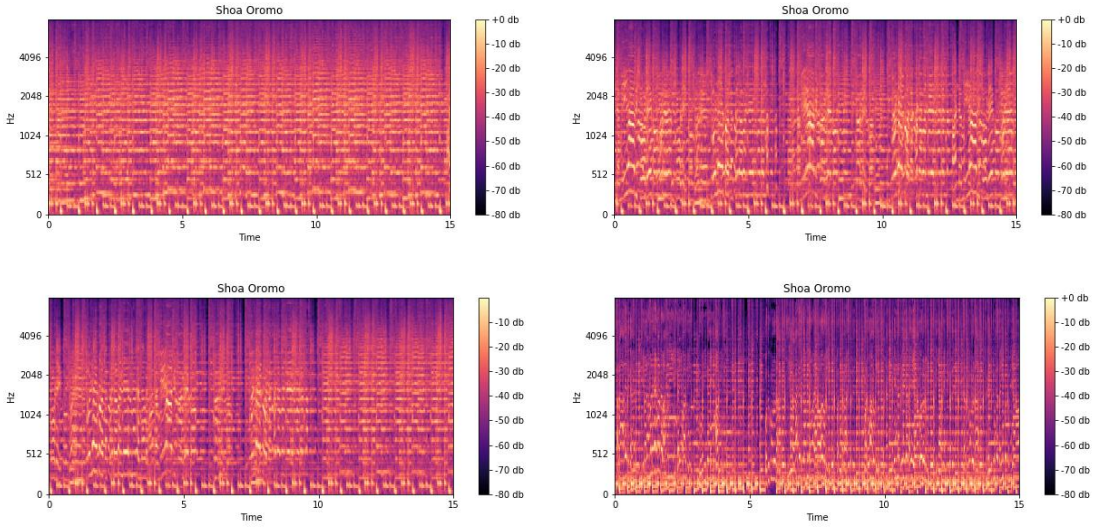
A.4 Gojjam



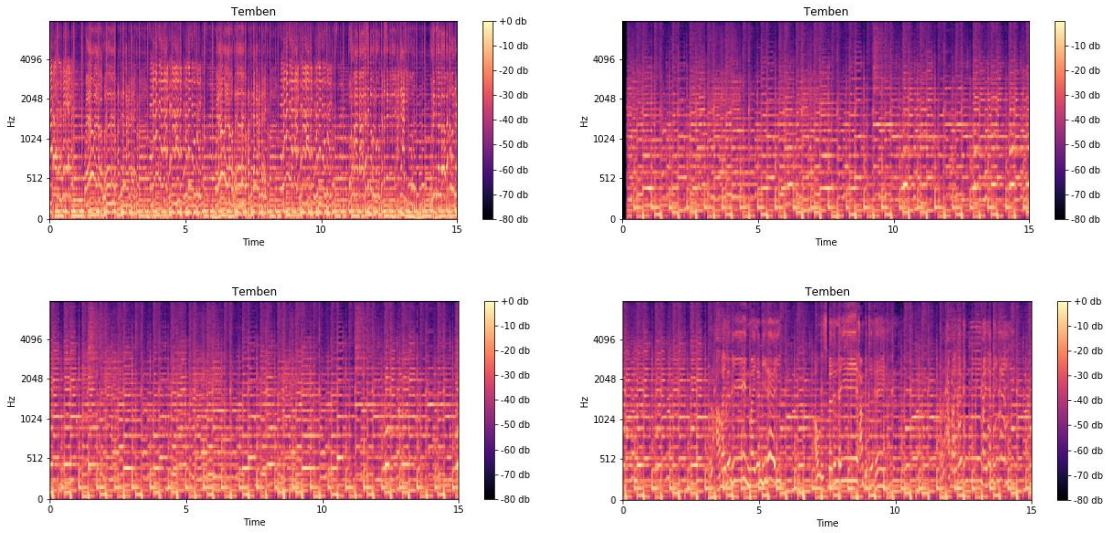
A.5 Harrer Oromo



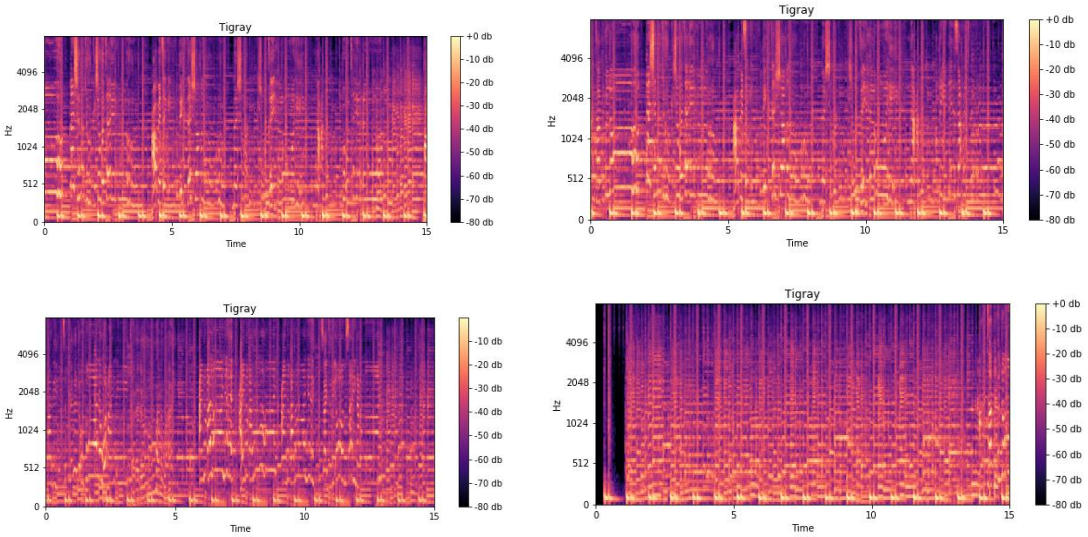
A.6 Shoa Oromo



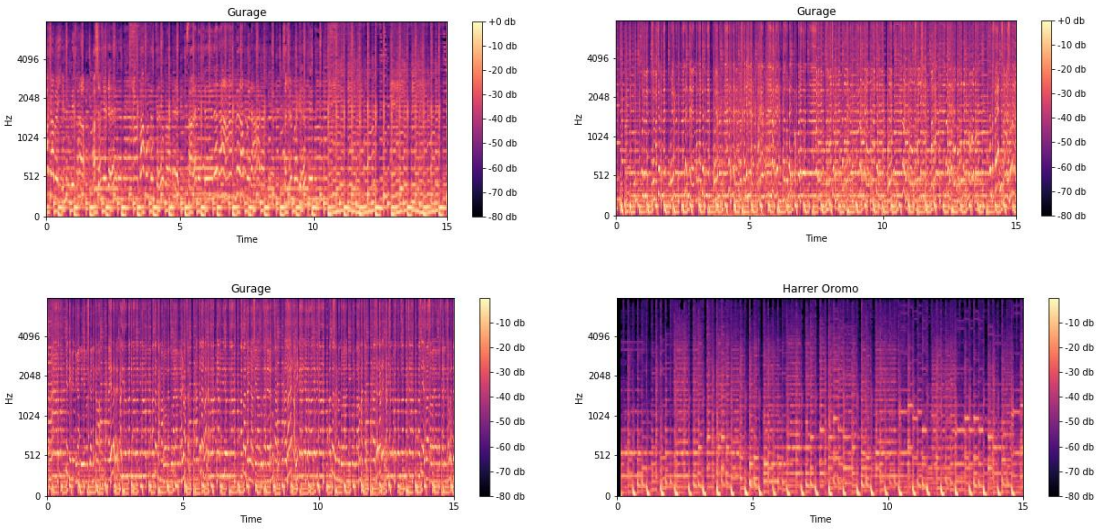
A.7 Temben



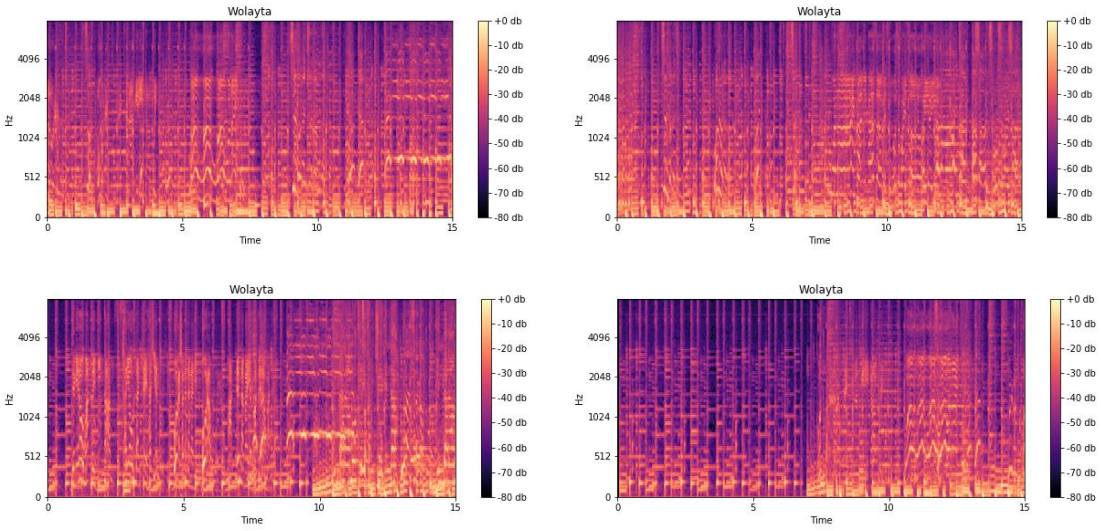
A.8 Tigray



A.9 Gurage



A.10 Wolayta



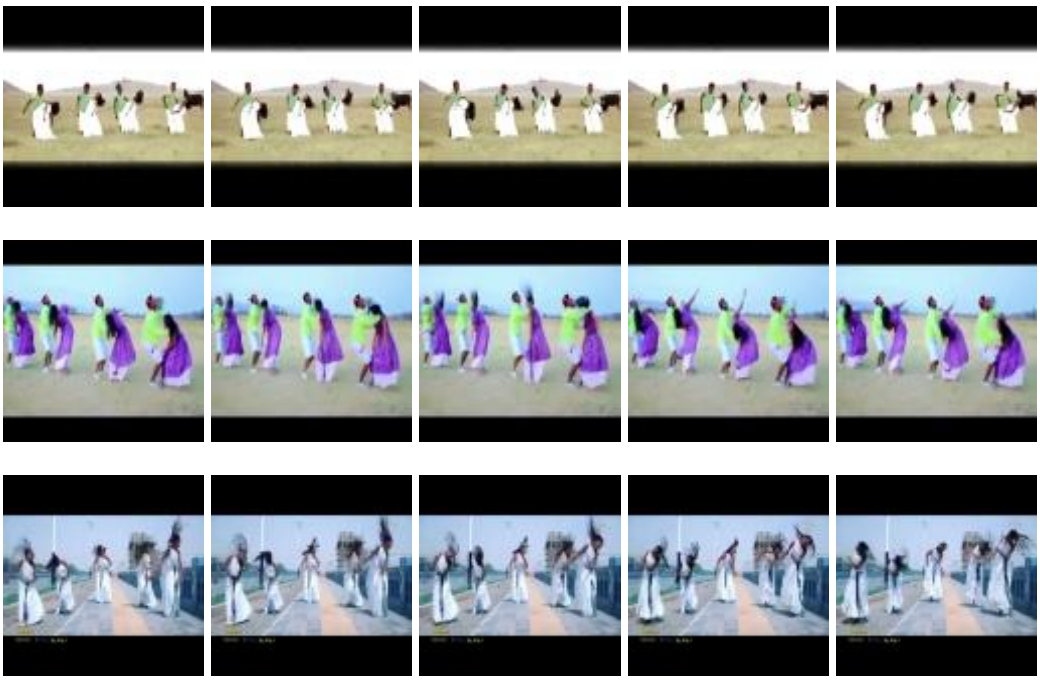
Appendix B: Sample Sequential Frames

B.1 Agew Wag Himra

B.2 Awi Agew

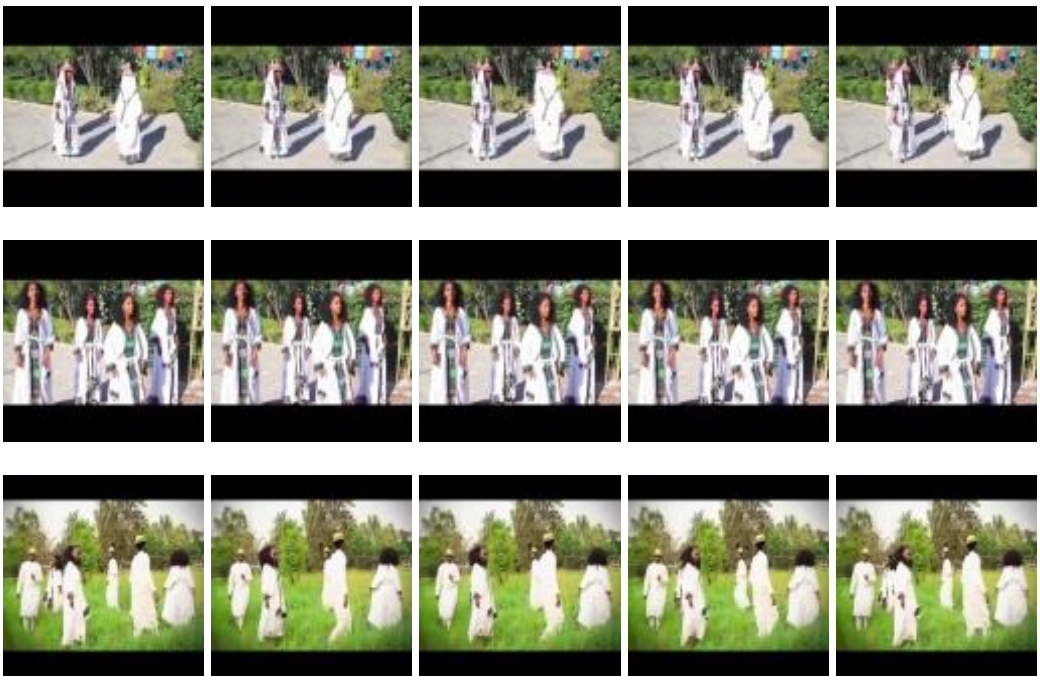
B.3 Gondar

B.4 Harrer Oromo



B.5 Temben

B.6 Tigray



B.7 Wolayta

Appendix C: Sample Source Code and Model Training

C.1 Extract mel-spectrogram of an audio data

```
# fucntion to create spectogram
def create_spectogram_for_video(video_id, genre):
    filename = get_audio_path(genre, video_id)
    y, sr = librosa.load(filename)
    spect = librosa.feature.melspectrogram(y=y, sr=sr, n_fft=2048, hop_length=1024, n_mels=128)
    spect = librosa.power_to_db(spect, ref=np.max)
    return spect.T
```

C.2 Pre-trained audio feature extracting component

```
### Input
input_shape = (128, 646, 1)
model_input = Input(input_shape, name='input')

### Convolutional blocks
conv_1 = Conv2D(filters = 16, kernel_size = (3,1), activation='relu', name='conv_1')(model_input)
pool_1 = MaxPooling2D((2,2))(conv_1)

conv_2 = Conv2D(filters = 32, kernel_size = (3,1), activation='relu', name='conv_2')(pool_1)
pool_2 = MaxPooling2D((2,2))(conv_2)

conv_3 = Conv2D(filters = 64, kernel_size = (3,1), activation='relu', name='conv_3')(pool_2)
pool_3 = MaxPooling2D((2,2))(conv_3)

conv_4 = Conv2D(filters = 64, kernel_size = (3,1), activation='relu', name='conv_4')(pool_3)
pool_4 = MaxPooling2D((4,4))(conv_4)

conv_5 = Conv2D(filters = 64, kernel_size = (3,1), activation='relu', name='conv_5')(pool_4)
pool_5 = MaxPooling2D((4,4))(conv_5)

flatten1 = Flatten()(pool_5)

dropout = Dropout(0.5)(flatten1)

### Recurrent Block
# Pooling Layer
pool_lstm1 = MaxPooling2D(pool_size_3, name = 'pool_lstm')(layer)

squeezed = Lambda(lambda x: K.squeeze(x, axis= -1))(pool_lstm1)

# Bidirectional GRU
lstm = Bidirectional(GRU(64, dropout=0.2, recurrent_dropout=0.2))(squeezed)

# Concat Output
concat = concatenate([dropout, lstm], axis=-1, name ='concat')

# Softmax Output
output = Dense(num_classes, activation = 'softmax', name='preds')(concat)

model_output = output
model = Model(model_input, model_output)

opt = RMSprop(lr=0.0005)  # Optimizer
model.compile(
        loss='categorical_crossentropy',
        optimizer=opt,
        metrics=['accuracy']
    )

model.fit(x_train, y_train, batch_size=16, epochs=60,  callbacks=callbacks_list)
```

C.3 Sample training of audio feature extracting module

```
Epoch 00009: acc improved from 0.63036 to 0.68590, saving model to ./models/etmac-final.best.h5
Epoch 10/60
2773/2773 [==============================] - 201s 73ms/step - loss: 0.8453 - acc: 0.7075

Epoch 00010: acc improved from 0.68590 to 0.70754, saving model to ./models/etmac-final.best.h5
Epoch 11/60
2773/2773 [==============================] - 201s 72ms/step - loss: 0.7738 - acc: 0.7353

Epoch 00011: acc improved from 0.70754 to 0.73530, saving model to ./models/etmac-final.best.h5
Epoch 12/60
2773/2773 [==============================] - 203s 73ms/step - loss: 0.7114 - acc: 0.7537

Epoch 00012: acc improved from 0.73530 to 0.75370, saving model to ./models/etmac-final.best.h5
Epoch 13/60
2773/2773 [==============================] - 203s 73ms/step - loss: 0.6454 - acc: 0.7797

Epoch 00013: acc improved from 0.75370 to 0.77966, saving model to ./models/etmac-final.best.h5
Epoch 14/60
2773/2773 [==============================] - 201s 72ms/step - loss: 0.6102 - acc: 0.7952

Epoch 00014: acc improved from 0.77966 to 0.79517, saving model to ./models/etmac-final.best.h5
Epoch 15/60
2773/2773 [==============================] - 202s 73ms/step - loss: 0.5651 - acc: 0.8042

Epoch 00015: acc improved from 0.79517 to 0.80418, saving model to ./models/etmac-final.best.h5
Epoch 16/60
2773/2773 [==============================] - 201s 73ms/step - loss: 0.5283 - acc: 0.8287

Epoch 00016: acc improved from 0.80418 to 0.82871, saving model to ./models/etmac-final.best.h5
Epoch 17/60
2773/2773 [==============================] - 201s 72ms/step - loss: 0.5222 - acc: 0.8258

Epoch 00017: acc did not improve from 0.82871
Epoch 18/60
2773/2773 [==============================] - 201s 73ms/step - loss: 0.4792 - acc: 0.8345

Epoch 00018: acc improved from 0.82871 to 0.83448, saving model to ./models/etmac-final.best.h5
Epoch 19/60
2773/2773 [==============================] - 201s 72ms/step - loss: 0.4451 - acc: 0.8449

Epoch 00019: acc improved from 0.83448 to 0.84493, saving model to ./models/etmac-final.best.h5
Epoch 20/60
2773/2773 [==============================] - 200s 72ms/step - loss: 0.4196 - acc: 0.8565
```

C.4 Extract audio feature using weights from the pre-trained network

```
model.load_weights('../models/etmac-final-weight.best.h5', by_name=True)

train_audio_x_x = np.expand_dims(train_audio_x, axis = -1)
valid_audio_x_x = np.expand_dims(valid_audio_x, axis = -1)
test_audio_x_x =  np.expand_dims(test_audio_x, axis = -1)

pred = model.predict(train_audio_x_x)
pred_val = model.predict(valid_audio_x_x)
pred_test = model.predict(test_audio_x_x)

np.savez('audio_feature_train', pred, train_audio_y)
np.savez('audio_feature_valid', pred_val, valid_audio_y)
np.savez('audio_feature_test', pred_test, test_audio_y)

model.summary()
```

C.5 CNN visual feature extracting module and image to video feature aggregating module

```
# Extract Visual Features Using VGG-16

conv_base = VGG16(weights='imagenet',
    include_top=False,
    input_shape=(WIDTH, HEIGHT, 3))

def extract_features(directory, sample_count):
    features = np.zeros(shape=(sample_count, 3, 3, 512)) #last output of conv base
    features_after_global = np.zeros(shape=(sample_count, 512)) #last output of conv base
    labels = np.zeros(shape=(sample_count,CLASS_NO))

    generator = datagen.flow_from_directory(
        directory,
        target_size=(WIDTH, HEIGHT),
        batch_size=batch_size,
        shuffle=False,
        class_mode='categorical')

    i = 0
    for inputs_batch, labels_batch in generator:
        features_batch = conv_base.predict(inputs_batch)
        features[i * batch_size : (i + 1) * batch_size] = features_batch
        labels[i * batch_size : (i + 1) * batch_size] = labels_batch

        i += 1

        if i * batch_size >= sample_count:
            break

    return features, labels

train_features, train_labels = extract_features(train_dir, 16020) #total number of train i - 16020
validation_features, validation_labels = extract_features(validation_dir, 3600) # 3600
test_features, test_labels = extract_features(test_dir, 3015) # 3015

# Image to Video Feature Aggregation
train_features = np.reshape(train_features, (264, 45,  3* 3* 512)) # total no of image / 45
validation_features = np.reshape(validation_features, (89, 45,  3* 3* 512))
test_features = np.reshape(test_features, (99, 45,  3* 3* 512))

train_labels = np.reshape(train_labels, (264, 45, CLASS_NO))
validation_labels = np.reshape(validation_labels, (89, 45,  CLASS_NO))
test_labels = np.reshape(test_labels, (99, 45, 7))
```

C.6 Sequence understanding LSTM module

```
model = models.Sequential()

model.add(layers.Bidirectional(layers.LSTM(1024,
                                           recurrent_dropout=0.2,
                                           return_sequences=False),
                               input_shape=(45, 4608) ))


model.add(Dropout(0.5))

model.add(Dense(64, activation='relu',  kernel_regularizer=regularizers.l2(0.001)))


model.add(Dense(100, activation='relu', kernel_regularizer=regularizers.l2(0.001)))

model.add(Dense(CLASS_NO, activation='softmax'))

model.summary()
```

C.7 Concatenation and prediction component

```
#video_module
num_classes = 7

input_video = Input((45, 4608), name="input_video")

lstm = Bidirectional(layers.LSTM(1024,recurrent_dropout =0.2, return_sequences=False), name='bidirectional_lstm')(in
dropout = Dropout(0.5, name="dropout")(lstm)
dense1 = Dense(64, activation = 'relu', name='dense_video_1', kernel_regularizer=regularizers.l2(0.001))(dropout)
dense2 = Dense(100, activation = 'relu', name='dense_vidoe_2', kernel_regularizer=regularizers.l2(0.001))(dense1)

#audio_module
input_audio = Input((384,), name="input_audio")

#concatenate
concat = concatenate([dense2, input_audio], axis=-1, name ='concat')

output = Dense(num_classes, activation = 'softmax', name='prediction_layer')(concat)

model_output = output
model = Model([input_video,input_audio], model_output)

model.summary()
```

```
Model: "model_3"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_video (InputLayer)        (None, 45, 4608)     0
__________________________________________________________________________________________________
bidirectional_lstm (Bidirection (None, 2048)         46145536    input_video[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 2048)         0           bidirectional_lstm[0][0]
__________________________________________________________________________________________________
dense_video_1 (Dense)           (None, 64)           131136      dropout[0][0]
__________________________________________________________________________________________________
dense_vidoe_2 (Dense)           (None, 100)          6500        dense_video_1[0][0]
__________________________________________________________________________________________________
input_audio (InputLayer)        (None, 384)          0
__________________________________________________________________________________________________
concat (Concatenate)            (None, 484)          0           dense_vidoe_2[0][0]
                                                                 input_audio[0][0]
__________________________________________________________________________________________________
prediction_layer (Dense)        (None, 7)            3395        concat[0][0]
==================================================================================================
Total params: 46,286,567
Trainable params: 46,286,567
Non-trainable params: 0
```

C.8 Model training for ETMC

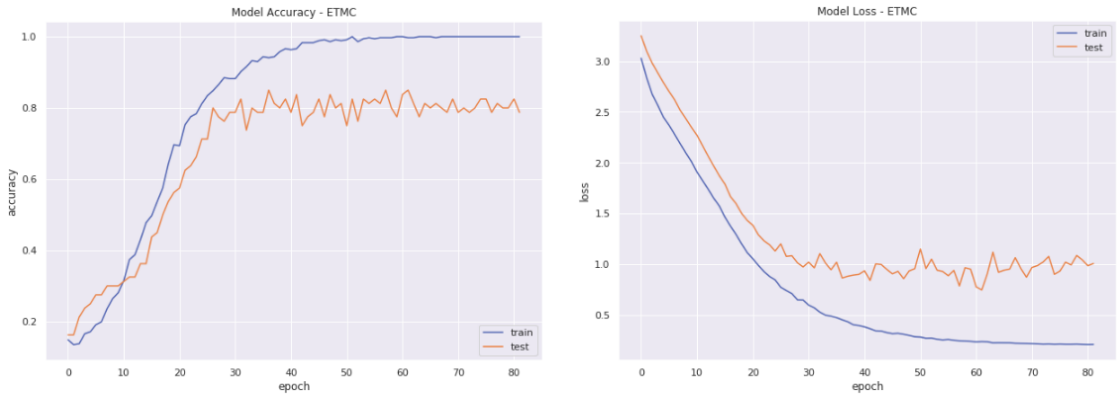
```
model.compile(optimizer=optimizers.RMSprop(lr=0.00001), loss='categorical_crossentropy', metrics=['accuracy'])

history = model.fit({'input_audio': x_train_audio, 'input_video': x_train_video}, y_train_video,
        epochs=200,
        batch_size=16,
        callbacks=callbacks_list,
        validation_data=({'input_audio': x_valid_audio, 'input_video': x_valid_video}, y_valid_video))
```

```
Epoch 00077: val_accuracy did not improve from 0.85000
Epoch 78/200
356/356 [==============================] - 77s 217ms/step - loss: 0.2084 - accuracy: 1.0000 - val_loss: 0.9910 -
val_accuracy: 0.8125

Epoch 00078: val_accuracy did not improve from 0.85000
Epoch 79/200
356/356 [==============================] - 77s 217ms/step - loss: 0.2095 - accuracy: 1.0000 - val_loss: 1.0847 -
val_accuracy: 0.8000

Epoch 00079: val_accuracy did not improve from 0.85000
Epoch 80/200
356/356 [==============================] - 77s 216ms/step - loss: 0.2072 - accuracy: 1.0000 - val_loss: 1.0414 -
val_accuracy: 0.8000

Epoch 00080: val_accuracy did not improve from 0.85000
Epoch 81/200
356/356 [==============================] - 77s 217ms/step - loss: 0.2054 - accuracy: 1.0000 - val_loss: 0.9836 -
val_accuracy: 0.8250
```

C.9 Sample training process

```
Epoch 28/200
356/356 [==============================] - 90s 253ms/step - loss: 0.7083 - accuracy: 0.8652 - val_loss: 1.0820 - va
l_accuracy: 0.7750

Epoch 00028: val_accuracy did not improve from 0.80000
Epoch 29/200
356/356 [==============================] - 88s 248ms/step - loss: 0.6463 - accuracy: 0.8848 - val_loss: 1.0119 - va
l_accuracy: 0.7625

Epoch 00029: val_accuracy did not improve from 0.80000
Epoch 30/200
356/356 [==============================] - 91s 257ms/step - loss: 0.6447 - accuracy: 0.8820 - val_loss: 0.9718 - va
l_accuracy: 0.7875

Epoch 00030: val_accuracy did not improve from 0.80000
Epoch 31/200
356/356 [==============================] - 94s 265ms/step - loss: 0.5953 - accuracy: 0.8820 - val_loss: 1.0191 - va
l_accuracy: 0.7875

Epoch 00031: val_accuracy did not improve from 0.80000
Epoch 32/200
356/356 [==============================] - 77s 218ms/step - loss: 0.5677 - accuracy: 0.9017 - val_loss: 0.9623 - va
l_accuracy: 0.8250

Epoch 00032: val_accuracy improved from 0.80000 to 0.82500, saving model to ./models/ETMC-FINAL.h5
Epoch 33/200
356/356 [==============================] - 76s 212ms/step - loss: 0.5257 - accuracy: 0.9157 - val_loss: 1.1029 - va
l_accuracy: 0.7375


Epoch 00033: val_accuracy did not improve from 0.82500
Epoch 34/200
356/356 [==============================] - 76s 212ms/step - loss: 0.4956 - accuracy: 0.9326 - val_loss: 1.0107 - va
l_accuracy: 0.8000

Epoch 00034: val_accuracy did not improve from 0.82500
Epoch 35/200
356/356 [==============================] - 76s 212ms/step - loss: 0.4859 - accuracy: 0.9298 - val_loss: 0.9435 - va
l_accuracy: 0.7875

Epoch 00035: val_accuracy did not improve from 0.82500
Epoch 36/200
356/356 [==============================] - 76s 212ms/step - loss: 0.4709 - accuracy: 0.9438 - val_loss: 1.0185 - va
l_accuracy: 0.7875

Epoch 00036: val_accuracy did not improve from 0.82500
Epoch 37/200
356/356 [==============================] - 76s 213ms/step - loss: 0.4500 - accuracy: 0.9410 - val_loss: 0.8622 - va
l_accuracy: 0.8500

Epoch 00037: val_accuracy improved from 0.82500 to 0.85000, saving model to ./models/ETMC-FINAL.h5
Epoch 38/200
356/356 [==============================] - 91s 256ms/step - loss: 0.4308 - accuracy: 0.9438 - val_loss: 0.8798 - va
l_accuracy: 0.8125

Epoch 00038: val_accuracy did not improve from 0.85000
Epoch 39/200
356/356 [==============================] - 91s 254ms/step - loss: 0.4009 - accuracy: 0.9579 - val_loss: 0.8910 - va
l_accuracy: 0.8000
```

C.10 Graph showing training process of ETMC



I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all sources of materials used for the thesis have been duly acknowledged.

Declared by:

Name: _____

Signature: _____

Date: _____

Confirmed by advisor:

Name: _____

Signature: _____

Date: _____