ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

3D FACE ANIMATION GENERATION FROM AUDIO USING CONVOLUTIONAL NEURAL NETWORKS

M.Sc. THESIS

Türker ÜNLÜ

Department of Computer Engineering

Computer Engineering Programme

JUNE 2022

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

3D FACE ANIMATION GENERATION FROM AUDIO USING CONVOLUTIONAL NEURAL NETWORKS

M.Sc. THESIS

Türker ÜNLÜ
(504171557)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Sanem Sarıel

JUNE 2022

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

EVRİŞİMSEL AĞLAR İLE SESTEN 3B YÜZ ANİMASYONU ÜRETİLMESİ

YÜKSEK LİSANS TEZİ

Türker ÜNLÜ
(504171557)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Assoc. Prof. Dr. Sanem Sarıel

HAZİRAN 2022

Türker ÜNLÜ, a M.Sc. student of ITU Graduate School student ID 504171557 successfully defended the thesis entitled "3D FACE ANIMATION GENERATION FROM AUDIO USING CONVOLUTIONAL NEURAL NETWORKS", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Sanem Sarıel** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Hazım Kemal Ekenel** Istanbul Technical University	.....
	Prof. Dr. Engin Erzin Koç University	.....
		.....

Date of Submission : **3 June 2022**
Date of Defense : **22 June 2022**

To my loved ones,

FOREWORD

I would like to thank my advisor Assoc. Prof. Dr. Sanem Sarıel for taking me as her student in her laboratory and for guidance about my thesis topic.

I would like to thank Arda İnceoğlu and Ahmet Levent Taşel for their contributions to this study. Ahmet helped during the beginning of this study. And although this is my thesis, without the works of Arda, this study would hardly be finished. I also thank Doğay Kamar and Sadık Uğursoy for their help.

We acknowledge the support of NVIDIA Corp. with the donation of the Quadro P6000 GPU used for this research.

This thesis is part of a project funded by a grant from the Scientific and Technological Research Council of Turkey (TUBITAK), Project Number: 3180690.

June 2022

Türker ÜNLÜ
(Computer Engineer)

TABLE OF CONTENTS

ABBREVIATIONS

ANN	**:** Artificial Neural Network
AU	**:** Action Unit
CNN	**:** Convolutional Neural Network(s)
FACS	**:** Facial Action Coding System
FFT	**:** Fast Fourier Transform
FPS	**:** Frame per second
GAN	**:** Generative Adversarial Network
GRU	**:** Gated Recurrent Unit
Hz	**:** Hertz
LSTM	**:** Long Short Term Memory
MFCC	**:** Mel Frequency Cepstral Coefficents
MHSA	**:** Multi Head Self Attention
ReLU	**:** Rectified Linear Unit
RNN	**:** Recurrent Neural Network(s)

SYMBOLS

b	**:** Neural network bias weights
K	**:** 2D convolution kernel size
L_s	**:** Shape loss term
L_m	**:** Motion loss term
L	**:** Total loss term
N	**:** 2D convolution number of filters
p	**:** Dropout probability
S	**:** 2D convolution stride
w	**:** Neural network multiplicative weights
x	**:** Input data sample
y_t	**:** Ground truth face parameter values
y_t'	**:** Face parameter values predicted by neural network

LIST OF TABLES

Page

LIST OF FIGURES

Page

3D FACE ANIMATION GENERATION FROM AUDIO USING CONVOLUTIONAL NEURAL NETWORKS

SUMMARY

Problem of generating facial animations is an important phase of creating an artificial character in video games, animated movies, or virtual reality applications. This is mostly done manually by 3D artists, matching face model movements for each speech of the character. Recent advancements in deep learning methods have made automated facial animation possible, and this research field has gained some attention.

There are two main variants of the automated facial animation problem: generating animation in 2D or in 3D space. The systems that work on the former problem work on images, either generating them from scratch or modifying the existing image to make it compatible with the given audio input. The second type of systems works on 3D face models. These 3D models can be directly represented by a set of points or parameterized versions of these points in the 3D space. In this study, 3D facial animation is targeted. One of the main goals of this study is to develop a method that can generate 3D facial animation from speech only, without requiring manual intervention from a 3D artist.

In the developed method, a 3D face model is represented by Facial Action Coding System (FACS) parameters, called action units. Action units are movements of one or more muscles on the face. By using a single action unit or a combination of different action units, most of the facial expressions can be presented.

For this study, a dataset of 37 minutes of recording is created. This dataset consists of speech recordings, and corresponding FACS parameters for each timestep. An artificial neural network (ANN) architecture is used to predict FACS parameters from the input speech signal. This architecture includes convolutional layers and transformer layers.

The outputs of the proposed solution are evaluated on a user study by showing the results of different recordings. It has been seen that the system is able to generate animations that can be used in video games and virtual reality applications even for novel speakers it is not trained for. Furthermore, it is very easy to generate facial animations after the system is trained. But an important drawback of the system is that the generated facial animations may lack accuracy in the mouth/lip movement that is required for the input speech.

EVRİŞİMSEL AĞLAR İLE SESTEN 3B YÜZ ANİMASYONU ÜRETİLMESİ

ÖZET

Yüz animasyonu üretme problemi, sanal ortamlardaki yapay karakterlerin konuşmaları için 3 boyutlu (3B) yüz modellerinin hareket ettirilmesi işidir. Bu problem, oyunlarda, animasyon filmlerde ve sanal gerçeklik uygulamalarında karşılaşılan karakterlerin konuşturulması, bu karakterlerin konuşmaları sırasında, yapımcıları tarafından aktarılmak istenen duyguların gerçek bir insana ait duygularmış hissi verebilmesi için çözülmesi gereken bir problemdir.

Bu iş çoğunlukla alanında bilgisi olan 3B çizimciler tarafından yapılmaktadır. Animasyonun üretimi için, bu çizimcilerin her konuşma ve her 3B yüz modeli için oynatılması gereken animasyonu önceden hazırlaması gerekmektedir.

Makine öğrenmesi ve özellikle bunun bir dalı olan derin öğrenmeye dayalı yöntemlerin son yıllarda gelişip yaygınlaşması ile, konuşmalar için yüz animasyonu üretilme işinin otomatik olarak yapılması üzerine çalışmalar yapılmıştır. Bu çalışmalar ile animasyonun 3B çizimciler tarafından elle üretilmesi için gereken maliyet ve emeğin düşürülmesi hedeflenmedir. Böylece bu yöntemler sadece büyük stüdyolar dışındaki yerlerde de kullanılabilir hale gelebilecektir.

Literatürdeki çalışmalar incelendiğinde, otomatik yüz animasyonu üretilmesi probleminin iki ana başlık altında incelendiği görülebilir. Bunlardan ilki, animasyonun sadece 2B görüntüsünü elde etmeye yönelik çalışmaları, ikincisi ise tamamen 3B bir yüz modeli elde etmeye yönelik çalışmaları içermektedir. Birinci ana grupta geliştirilen yöntemler doğrudan resimler üzerinde çalışırlar ve sıfırdan bir resim üretme üzerine, veya elde olan temel bir resim üzerinde gerekli değişiklikleri yaparak resmi, sisteme girdi olarak verilen konuşma bilgisine uygun bir hale gelecek şekilde değiştirirme üzerinedir. İkinci ana gruptaki çalışmalar önceden hazırlanmış 3B yüz modelleri üzerinde çalışırlar. Bu yöntemlerin çıktıları, baz alınan 3B yüz modelinin o anki verilen konuşma sinyaline göre durumudur. Elde edilen çıktı ortamdan bağımsız bir yüz modelidir. Bu çalışmalarda baz alınan yüz modeli dijital ortamda farklı şekillerde temsil edilebilmektedir. Temsil şekli, doğrudan yüz modeli üzerindeki noktaların 3B uzaydaki koordinatları olabileceği gibi, yüzdeki değişmesi olası bölgelerin parametrik hale getirilmiş bir temsil de olabilir. Bu tez çalışmasında, konuşma animasyonunun 3B bir yüz modeli baz alınarak üretildiği bir yöntem açıklanmıştır. Çalışmanın temel hedeflerinden biri, herhangi bir 3B çizimcinin elle müdahalesine gerek kalmadan, tamamen otomatize bir şekilde girdi olarak verilen konuşma sesinden 3B bir yüz animasyonu üretebilecek bir yöntem tasarlamaktır.

Geliştirilen yöntem için, yüz modeli temsili olarak yüzdeki hareket eden kasları temel alarak oluşturulmuş Yüz Hareketleri Kodlama Sistemi (Facial Action Coding

System (FACS)) kullanılmaktadır. Bu sistemde yüz, aksiyon birimleri denen ve her biri yüzdeki bir veya bir grup kasın hareketi sonucu yüzdeki noktaların değişimini tanımlayan parametreler ile temsil edilmektedir. Bu aksiyon birimleri temel alınarak, herhangi bir aksiyon biriminin veya bir grup farklı aksiyon biriminin hangi durumlar sonucu değiştiği belirlenebilmekte, böylece neredeyse istenen bütün olası yüz ifadeleri daha az parametre kullanılarak ifade edilebilmektedir. Çalışmada önerilen sistem ile yüz animasyonu elde edebilmek için, hareketli bölgeleri FACS ile uygun olacak şekilde önceden ayarlanmış bir 3B yüz modeli gerekmektedir. Fakat eğitim için kullanılan veri kümesindeki model ile daha sonra animasyon elde edebilmek için kullanılan yüz modeli aynı olmak zorunda değildir.

Çalışmanın gerçeklenebilmesi için, toplam 37 dakika sürelik bir veri kümesi oluşturulmuştur. Veri kümesi, farklı hikaye ve film kesitlerinin seslendirilmesiyle oluşan 11 farklı kayıttan oluşmaktadır. Veri seti içerisinde konuşma kayıtları, konuşmacının görsel kaydı ve videodaki her kare için konuşmacının yüz modelinin FACS temsili için gereken parametrelerin değerleri bulunmaktadır.

Sisteme girdi olarak gelen konuşma sesi sinyalinden, 3B yüz modelinin animasyonu için gereken FACS parametre değerlerini tahmin etmek için evrişimsel sinir ağı katmanı ve dönüştürücü katmanı içeren bir Yapay Sinir Ağı (Artificial Neural Network) mimarisi tasarlanmıştır. Girdi olarak kullanılacak olan ses sinyali, yapay sinir ağı mimarisine aktarılmadan önce bazı ses işlemlerinden geçmektedir. Öncelikle ses, 16 milisaniye uzunluğundaki pencerelere bölünür. Her penceredeki ses sinyali normalize edilip Hızlı Fourier Dönüşümü (Fast Fourier Transform) hesabı yapılır ve son olarak MFCC öznitelikleri elde edilir. Yapay sinir ağının girdi formatı bu MFCC öznitelik vektörleridir. Üretilecek her bir animasyon karesi için, o anın öncesinden ve sonrasından toplam 520 milisaniye olacak şekilde bir pencere kümesi alınır ve yapay sinir ağına girdi olarak verilir.

Tasarlanan yapay sinir ağı mimarisi içerisinde evrişimsel (convolutional) katmanlar ve dönüştürücü (transformer) katmanları bulunmaktadır. Eğitim performansını artırmak için çeşitli normalizasyon yöntemleri de uygulanmıştır.

Geliştirilen sistemin sonuçlarını değerlendirmek için, sistemin bazı sesler için ürettiği video çıktıları bir grup katılımcıya gösterilip görüşleri alınmıştır. Rastgele seslerden elde edilen videolara ek olarak, başka bir takım 3B yüz animasyonu üreten çalışmalarda kullanılan ve çıktı videoları mevcut bulunan belirli ses kayıtları için de video çıktıları üretilmiş ve katılımcılardan bunları karşılaştırmaları istenmiştir. Elde edilen sonuçlara göre, sistemin ürettiği animasyonun bilgisayar oyunları ve sanal gerçeklik uygulamaları için yeterli kalitede olduğu gözlenmiştir.

Sistemin avantajlarından biri, eğitim veri kümesinde olmayan kullanıcıların sesi için de konuşma animasyonları üretebilmesidir. Başka bir avantajı da, bir seferlik eğitim süreci tamamlandıktan sonra herhangi bir ses için konuşma animasyonu üretmenin kolaylığıdır. Fakat mevcut sistemin önemli bir dezavantajı, elde edilen animasyonlardaki ağız ve dudak hareketlerinin detayları açısından her zaman doğru sonuçlar verememesidir, bazı durumlarda girdi olarak veren ses için gereken ağız/dudak senkronizasyonunda sapmalar görülmektedir.

Çalışmaya daha sonrasında yapılabilecek geliştirmeler arasında, daha sağlıklı bir karşılaştırma yapabilmek için diğer 3B yüz animasyonu üreten çalışmalarda kullanılan yüz modelleri ve veri kümelerinin formatının bu çalışmaya uyarlanması, eğitim için kullanılan veri kümesinin farklı konuşmacıları içerecek şekilde oluşturulmasıdır.

1. INTRODUCTION

1.1 Purpose of Thesis

The main goal of this study is to propose a system that is able to generate facial animation for an artificial character from the audio speech signal. For this purpose, a deep neural network architecture is proposed and implemented. Also, a 3D face model representation that is easy to interpret is used for the output of the system, which allows easy post-processing by 3D artists if desired.

A dataset is created which includes the audio recordings, the camera recordings of a voice actor during these audio recordings, and the face parameter values representing the facial expression in each frame.

1.2 Motivation

Creating 3D facial animation of a talking person in a digital environment is a mandatory work in applications such as animated movies, video games, and some virtual reality applications. There are different methods of obtaining the facial animation of a speech. The existing methods differ in terms of the desired level of animation detail, the required budget, and the available amount of time. The most commonly used method is to record a voice actor during dubbing of the given speech with cameras and then transfer these recordings to a digital environment using computer vision techniques. Although the resulting animation quality of this method has been improved over recent years, due to the cost of obtaining the required software/hardware for processing and post-processing of the first obtained result makes this method expensive. Therefore, only a small amount of studios can afford to use such methods.

Another problem with creating the required facial animations for speeches manually is that sometimes a face model needs to be animated for a newly encountered

speech, therefore, it is not possible to create animations beforehand. This problem usually occurs in virtual reality applications where people are represented by artificial characters in a digital environment, and their characters need to be animated as they speak. It can also be a problem in video games if the game contains dynamically generated text/speech but such usage is rare, if ever.

Upon viewing methods in the literature of procedurally generating facial animation, it has been observed that the level of animation detail obtained using these methods is still not as high as manual animating methods. Furthermore, some methods require more hand-tuning than just receiving audio input. The purpose of this study is to propose a method that can procedurally generate facial animation from speech without the usage of any specialized recording devices or studios, just by using audio input and obtaining the facial animation output for the given speech. For this purpose, we propose a neural network architecture that is composed of convolutional layers and self-attention layers. With this system, it is possible to generate facial animations from speech.

The dataset required for the training of the proposed neural network architecture consists of speech recordings of an actor and the FACS [7] parameter values that animate the target 3D face model. Once the training process is complete, the system can be used to generate facial animations for speeches of different people other than the people in the training dataset. The quality of these generated animations depends on the generalization of audio features during the training process.

Using FACS [7] representation, different facial expressions can be obtained by linear combinations of different parameters. Also, FACS parameters being easier to interpret by people allows easier modifications on the output if desired. Figure 1.1 shows an example of changing the value of a single FACS parameter thus making the animation model open its mouth.

The most important contributions of this study are designing a convolutional (based on Karras et al [2]) and transformed based neural network architecture that can be used to generate facial animations from speech and the creation of a dataset that consists of speech voice recordings and corresponding FACS parameter values for each frame of the speech.

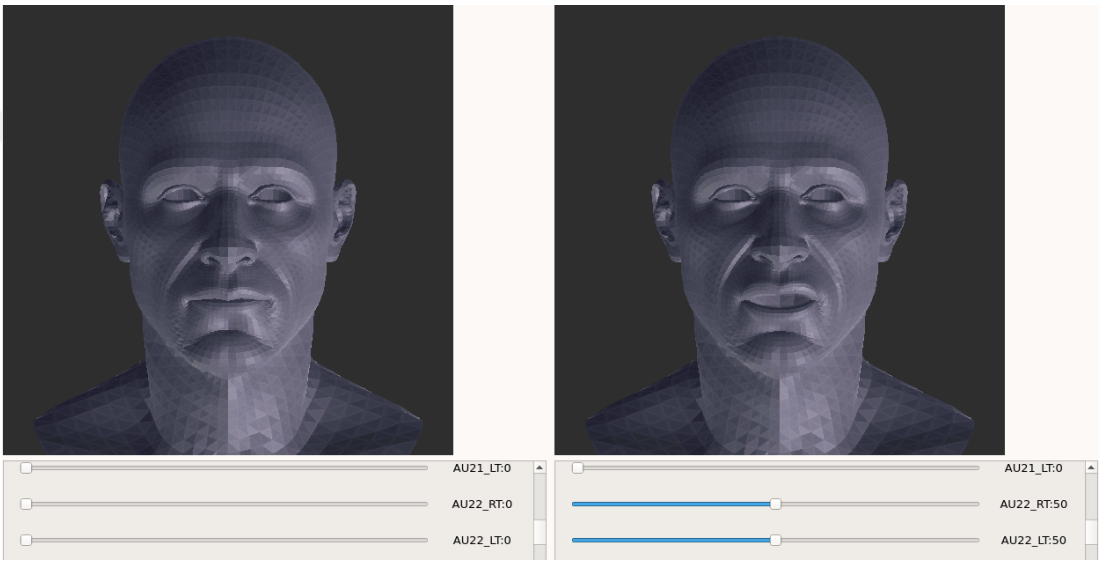

Figure 1.1 : An example 3D face model used in the study in which moving parts of the face are encoded as FACS parameter values. The image on the left is the base model and the image on the right is the resulting model after the value of parameter 22 is changed.

2. BACKGROUND

In this chapter, the methods used in this study are described in order to understand the presented work. First, neural networks are explained briefly as they are the core of the used method. Then Convolutional Neural Networks (CNN) and Transformer architecture are explained as they are the main feature extractor components in our neural network architecture. Lastly, a system to encode facial expressions is presented which is used for representing facial expression parameters in the dataset created for the study and in the 3D face model used for rendering the obtained results.

2.1 Artificial Neural Networks

Neural networks are systems composed of processing units called neurons. The terms "neural network" and "neuron" are given because the system is claimed to be inspired by the neural system in the body. Neurons take the numerical value(s) as their input and output a new numerical value by performing some operations. Each input of a neuron x_i is multiplied by a weight value w_i. The results of these multiplications are summed up, and a bias value is added to this sum, then an activation function is applied to the final sum:

$$f(\sum_i w_i x_i + b) = 1 \tag{2.1}$$

The number obtained from the function becomes the output of the neuron. Figure 2.1 shows the inputs and outputs of a single neuron in a neural network.

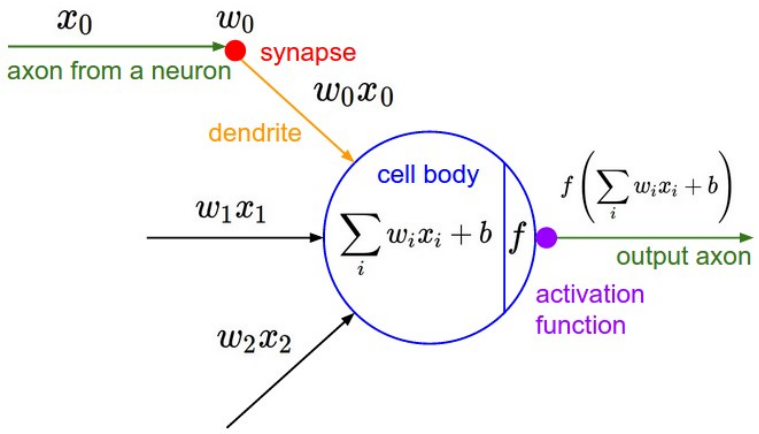

Figure 2.1 : Demonstration of inputs and outputs of a single neuron [3].

The activation function at the end of the neuron is the main reason why neural networks can approximate any function. This function should be a non-linear function, otherwise, the neural network can only learn linear functions. Some of the commonly used activation functions such as Sigmoid, Tanh, ReLU, Leaky ReLU and their graphs are shown in Figure 2.2.

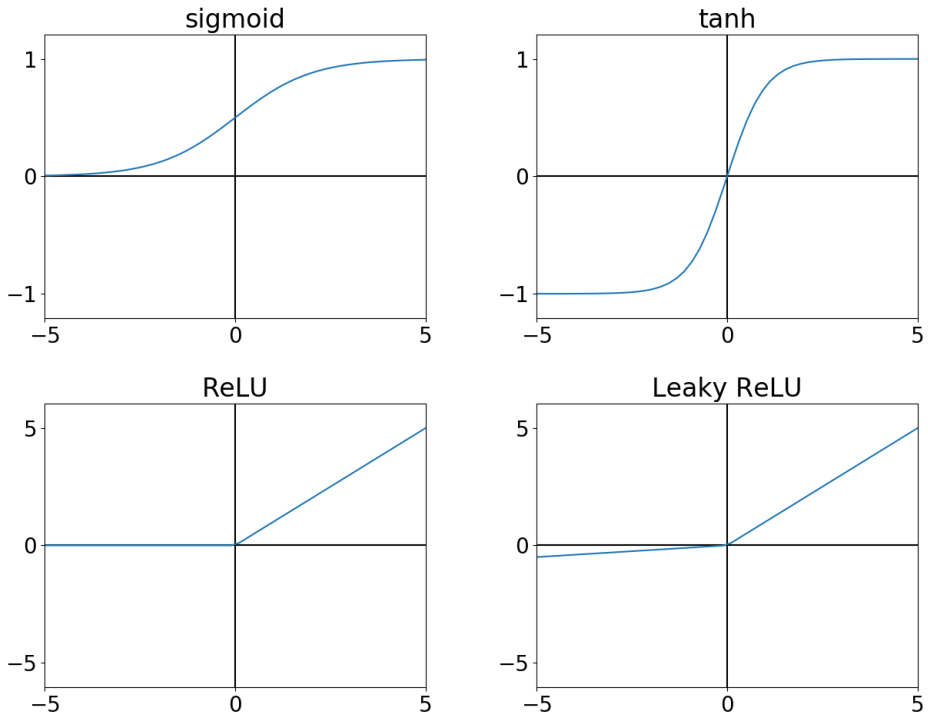

Figure 2.2 : Some of the commonly used non-linear activation functions in artificial neural networks.

Neural networks are formed by groups of neurons. Each group is called a layer. Each neuron is connected to all neurons in the previous layer, and all neurons in the next layer. Outputs of neurons in a layer are the input of the next layer. The first layer is called the input layer, and the last layer is called the output layer.

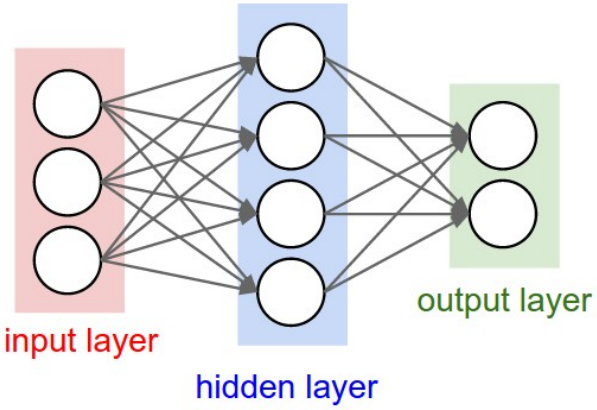

Figure 2.3 : An example neural network architecture with one hidden layer [3]

Training of a neural network is an optimization process. Parameters of the neural network are adjusted so that for an input value, the obtained output value from the neural network is closer to the true output value. Error in the prediction is measured by a loss function. Different loss functions are used depending on the type of the problem being solved. The most common way to optimize neural network parameters is the backpropagation [8] algorithm. Learning the correct values of neural network parameters is done by minimizing the loss function for the given samples.

2.2 Convolutional Neural Networks

Although neural networks are capable of learning any target function, using fully connected neural networks for every problem may not be practical due to the large number of parameters required for some problem domains. One of these domains is computer vision.

Convolution in computer vision is a method that is based on applying a sliding window filter to the image. Figure 2.4 shows an example convolution operation process. This method is used for extracting local features from the image. The technique was already popular in the computer vision field without the usage of neural networks. After the

success of Krizhevsky et al. [9] in ImageNet competition 2012, combining convolution with neural networks by using parametric filters became the mainstream approach to solve many computer vision problems. For each position of the filter, after applying the dot product between image values and filter parameters, the resulting value is passed to a non-linear activation function such as ReLU.

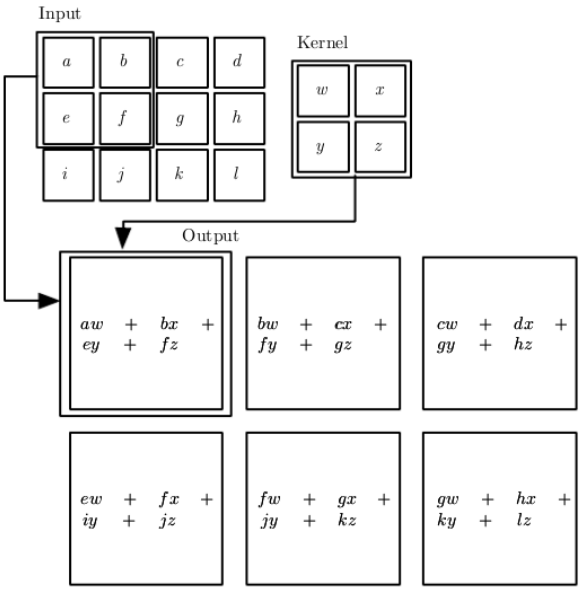

Figure 2.4 : An example convolution operation performed on a 3x4 matrix, with kernel size 2x2 and stride 1x1. [4]

Downsampling methods such as max-pooling are also frequently used in convolutional neural networks, to reduce input size.

2.3 Transformer

In order to capture dependencies in a sequential input in neural network architectures, using Recurrent Neural Networks (RNN) was the standard solution for a long time. Although naive RNNs could only capture short-time dependencies in a sequence, their performance is improved by variants such as LSTM (Long Short Term Memory), GRU (Gated Recurrent Unit), etc.

Vaswani et al. [5] have shown that the self-attention mechanism can be used instead of neural networks to capture dependencies in the sequences. Not only it performs better on similar data, but it is also able to capture dependencies that are farther away.

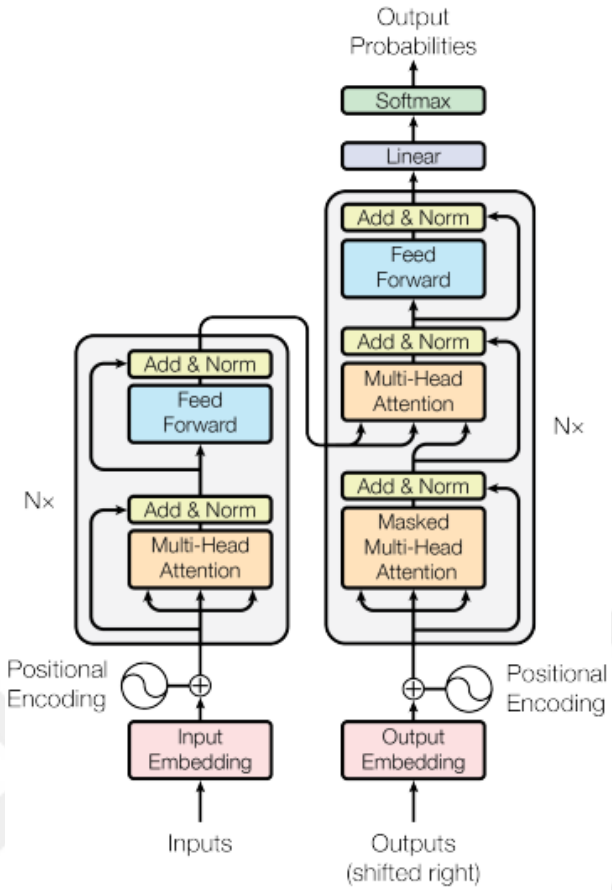

Figure 2.5 : Transformer architecture shown in Vaswani et al. [5].

In this study, we do not perform sequence-to-sequence generation. Instead, the main purpose of using the transformer component is to have an encoder-decoder structure that is able to capture important parts of the given speech signal sequence. The output of the transformer component is used as a feature vector.

2.4 FACS

There have been various studies about whether facial expressions can be used to decide the emotions of a person, regardless of linguistic and cultural differences. One of the main research areas in which this topic is being researched is psychology. Psychologists often try to relate some facial expressions to known emotion types such as anger, happiness, sadness, etc.

Paul Ekman and Wallace Verne Friesen were two of the psychologists who worked on the relationship between facial expressions and emotions. In 1978, they published

Facial Action Coding System (FACS) [7]. In FACS, different moving muscles or groups of muscles on the face are defined as action units. Movements on the face are defined as contracting and relaxing of these muscles. Figure 2.6 shows some example action units and which region/muscle of the face they represent.

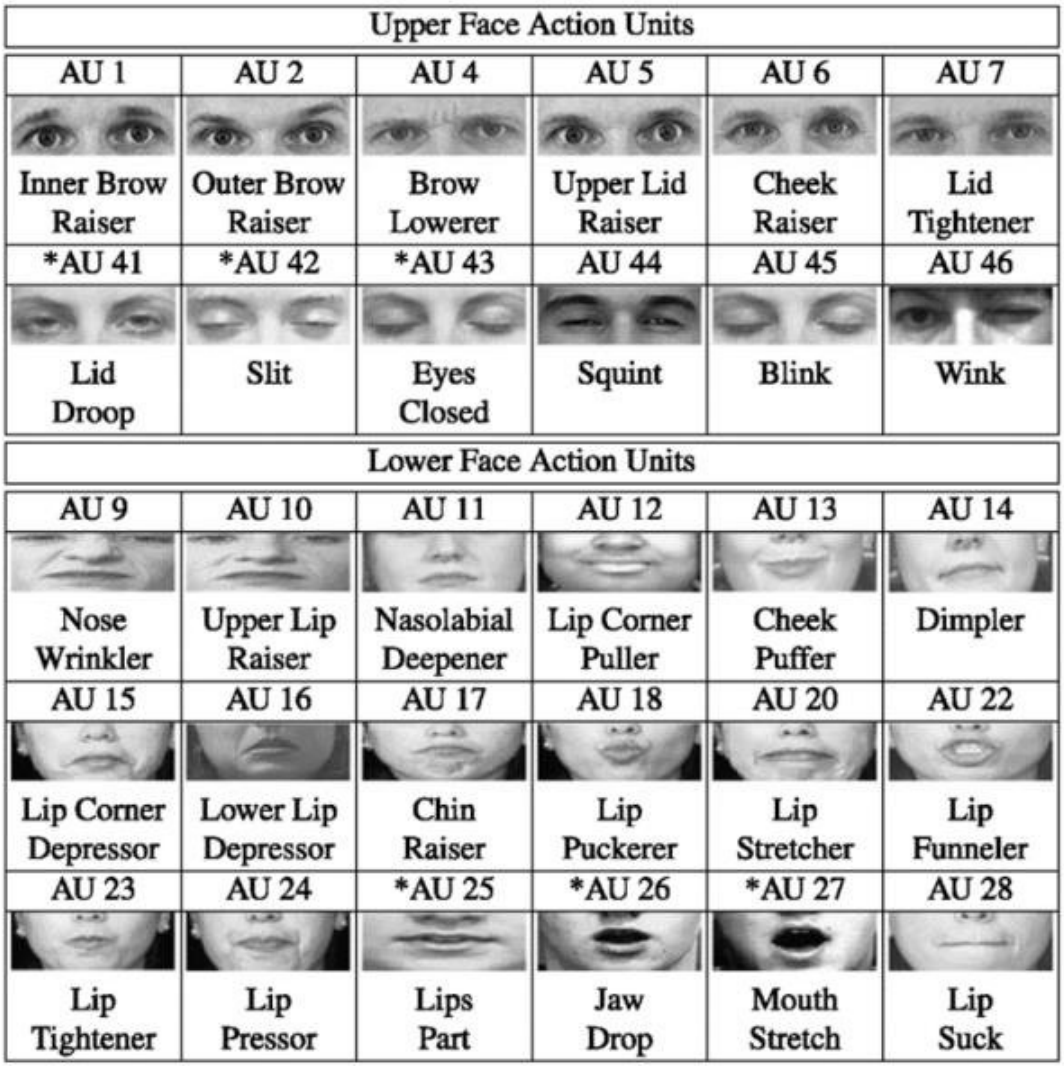

Figure 2.6 : Demonstration of action units in the FACS from [6].

Ekman and Friesen also claimed that using these action units can be used to identify the emotion of people from their facial expressions [1]. Table 2.1 shows an example mapping between emotions and some action units. Although, FACS representation is not the only way used in emote recognition tasks. Other feature representations, such as landmark points, or just using raw face image input are also used in emotion recognition.

Table 2.1 : An example matching from emotions to action units [1].

Emotion	**Action Units**
Happiness	6+12
Sadness	1+4+15
Surprise	1+2+5B+26
Fear	1+2+4+5+7+20+26
Anger	4+5+7+23

3. LITERATURE REVIEW

Procedurally generating facial animations from the speech is a known problem in computer graphics. Before the adaptation of deep learning methods, approaches to solve this problem relied mostly on rule-based solutions such as Cohen et al. [10]. There are also methods such as Taylor et al. [11] that try to group similar-looking sounds, known as visemes, by clustering instead of using a completely rule-based approach. The drawback of these methods is that they are not only trying to detect the correct pose of the lips from audio, but they also need to have a method for transitioning between different poses.

With the recent success of deep learning methods, the problem is also revisited. The facial animation generation from speech problem has different variations. Studies that use machine learning methods for solving the problem can be categorized based on the inputs and outputs used in the proposed systems. Based on the input, the studies can be categorized as follows:

1. Only audio input
2. Audio and text input
3. Audio and single image input
4. Audio and video input
5. Combinations of different modalities

And based on the output, studies can be categorized as follows:

1. 2D face image
2. 3D face model

Most of the studies in the literature about facial animation generation problem aim to generate a 2D image of a talking person. Works in these studies consider the facial animation generation problem as a supervised machine learning problem, and the aim is to find a mathematical relationship between input speech audio signal and the output images. These studies again can be separated into 2 groups whether they use Generative Adversarial Networks (GAN) [12] or not. Studies such as [13], [14], [15], [16] model the problem as a classical machine learning problem and try to learn a function that matches from the input speech signal to output image. The second group consists of studies that use GANs to produce the desired facial animations [17], [18], [19].

Studies on the 3D face models [2], [20], [21] differ from each other in terms of:

1. Input type
2. Used method
3. Output format

In one of the studies, Karras et al. [2] propose a system which only takes speech audio signal as its input, and the output of the system is the raw 3D coordinates that represent a mesh of a person's face. After training is complete, the system also uses an additional embedded vector during inference time which is called an emotion vector in the study. In Cudeiro et al. [20], the input of the proposed system is speech audio signals and a 3D face model defined in FLAME [22] format, and the output is the parameters that represent the changes on a 3D FLAME model. In Zhou et al. [21], the input is speech audio signals and the output is the keypoints around the mouth that represents the position/movement of lips. Also, the system tries to identify phonetic groups in the speech as its auxiliary output.

The problem that is investigated in this study lies in the group of 3D facial animation generation studies. We propose a different 3D face model representation and a different network architecture. In the study, it is observed that using CNN layers and transformer layers together produces better results compared to using only CNN layers. Also, using FACS parameters for face model representation allows greater control over the output

of the system compared to the systems that use 3D raw points directly or only use points around the mouth, thus allowing easier modifications either in the system itself or by a 3D artist for post-processing.

4. FACIAL ANIMATION GENERATION

In this section, the stages of the proposed system are explained. First, how the created 3D face model and its moving parts are represented is presented. Then, collection of a dataset consisting of audio recordings and corresponding face model parameters with 30 frames per second is explained. Then, preprocessing operations performed on the input audio signal before feeding it to the neural network are explained. Finally, the deep neural network architecture used in the study and its loss function is explained.

4.1 Face Model Representation

In this study, facial animation generation is performed on a 3D face model and the moving muscles that are defined on this face model. In order to present facial expressions, moving muscles on the face must be defined with Facial Action Coding System (FACS). For the 3D face model used in this study, 69 different FACS parameters are defined where each of them takes values in range $[0, 1]$. Different facial expressions and emotions can be obtained by using different linear combinations of these FACS parameters. The artificial neural network model that is used in this study predicts the values of these 69 parameters as its output.

Representing the face model with these FACS parameters instead of directly using 3D points in the space makes it easier to use a trained model on a different face it is not trained for. This is because the learned parameters of the model are not specific to the topology of the used 3D face model, but they are rather specific to moving parts of a high-level face definition, such as upper lip. But an important disadvantage of this face representation is that in order to obtain results from the model, a face model whose moving parts are defined as FACS parameters must be defined beforehand.

4.2 Dataset

A 37-minute long dataset is created for the training process that is formed by collecting stories and film scenes voiced by a single actor. Speeches are chosen in such a way that the voice actor needs to speak with different tones of voice. The dataset is composed of speech audio recordings and values of FACS parameters that correspond to the facial expression of the speaker at each frame. The sampling rate of the recorded audio clips is 48 kHz. In order to obtain FACS parameter values required for the dataset, the voice actor is recorded from multiple views during the dubbing, as shown in Figure 4.1. Then these recordings are processed using Dynamixyz [1] recording program. Finally, the animation information is transferred to a 3D computer graphics application Autodesk Maya[2] to a 3D face model.

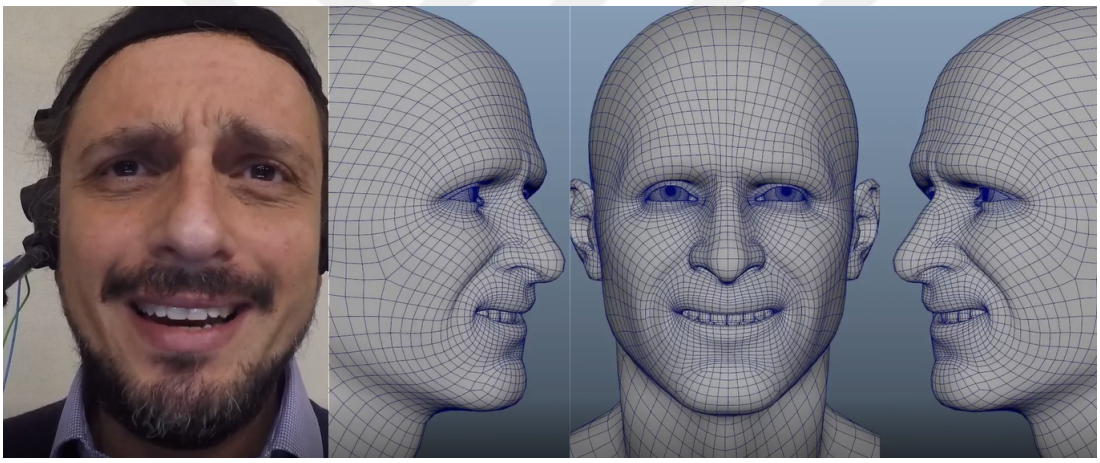

Figure 4.1 : A scene from the dataset recording process. The face of the voice actor is recorded from multiple angles during voice recording.

[1] https://www.dynamiyz.com/
[2] https://www.autodesk.com/

Table 4.1 : Recordings in the created dataset for the study.

Recording	Duration
1	01:06
2	07:31
3	04:43
4	00:52
5	01:30
6	04:13
7	02:26
8	08:34
9	00:48
10	05:21
11	01:28

Each recording includes 30 FPS image and animation data. Image data is not used during the training process directly. Inputs of the system are only speech audio signals and the FACS parameter values. An example speech sequence from the created dataset is shown in Figure 4.2.

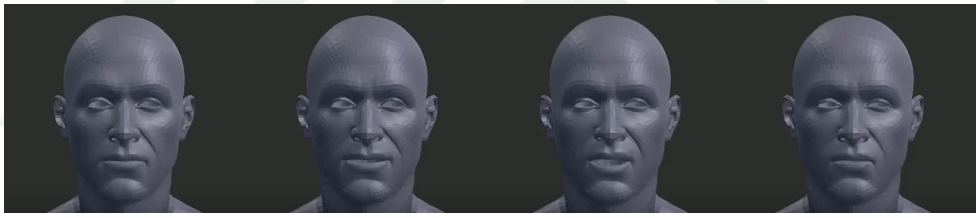

Figure 4.2 : An example short speech from the dataset created for the study. Speaker is saying word "*for*" in English.

4.3 Preprocessing

The input speech signal is converted into Mel Frequency Cepstral Coefficients (MFCC) features and then processed in the neural network. Using MFCC features instead of raw audio signals is a widespread practice in audio processing tasks. During the processing of audio signals, audio is divided into windows with length of 16 milliseconds, then the audio signal is converted from the time domain to the frequency domain by applying Fast Fourier Transform (FFT) to all windows. MFCC feature vectors with length of 32 are calculated for each audio window. The audio signal of each window is normalized by subtracting DC (Direct Current) component during the calculation of MFCC feature

vectors. The whole audio signal is processed by using a sliding window approach. In this operation, the window stride is decided as 8 milliseconds. In order to output a single frame of animation, the neural network model takes a total of 64 windows as its input, which corresponds to 520 milliseconds of an audio signal.

Since input audio signal can be recorded with different recording devices, there might be a difference in the sampling rate of recorded audio clips among training time audio clips and test time audio clips, or even among different training audio clips. Input audio signal is always resampled at 16 kHz frequency to prevent this problem.

During the training process, data augmentation is used for both enlarging the dataset and increasing the generalization performance of the model, a common approach in machine/deep learning projects. This is done during the processing of 16 ms audio windows. While processing these windows using a sliding window approach, starting points of some windows are randomly shifted up to 16 ms backwards or forwards. Since this shift operation results in a time between two discrete animation frames, we need to sample face value parameters for this new in-between frame. This information is generated by linearly interpolating between two neighbour frames.

4.4 Artificial Neural Network Architecture

At the beginning of the study, a convolutional neural network model is developed based on the model proposed by Karras et al. [2]. This model is composed of 3 main blocks. These blocks are: (i) Input layer, (ii) sound analysis layer, and (iii) output layer. The input of the model is decided as MFCC features unlike the proposed model in Karras et al. [2] which uses autocorrelation coefficients. In the first two layers, components of human speech are analyzed, and face structures such as tongue, lip, and chin are analyzed morphologically. A structure with 10 convolutional layers is formed for this purpose. Each convolutional network layer performs a 2D convolution operation on its input. Each layer is defined with 3 parameters: filter size K, stride S and number of filters N. After all these 10 convolutional layers, a feature vector with size 256 is obtained at the end.

After the first results are obtained, some modifications are done to the base neural network architecture. The number of filters in each convolution layer is changed. Batch normalization and dropout layers are added after each convolutional layer. Dropout probability is set to *p=0.1*. This value is determined after experimenting with different p values. Added dropout layers help avoiding the overfitting problem. Non-linear activation function after convolutional layers is changed from ReLU to Leaky ReLU.

After investigating the recent works in the literature, it has been seen that works that use attention methods seem to increase the performance of the artificial neural networks in sequence generating problems. Transformer [5] based methods are one type of these attention methods. In this study, Multi Head Self Attention (MHSA) is used. This transformer module works in parallel with the CNN module similar to [23] and also uses MFCC feature vectors as its input. The used transformer has 4 heads and the output of this transformer layer is a feature vector with size of 64.

The output layer of the model is composed of 2 fully connected layers. The output feature vectors of CNN and Transformer layers that are working parallel in previous layers are concatenated into a single feature vector. This combined feature vector is used as input to the first fully connected layer. This first fully connected layer outputs a feature vector with size of 150, and then these values are passed into Leaky ReLU activation function and passed into the last fully connected layer. The output of this final layer is the 69 FACS parameters of the face model the network is trying to predict. Figure 4.3 shows all components of the used artificial neural network that is used in the study.

4.5 Loss Function and Training Process

The loss function is the combination of two different terms: Shape loss and motion loss. For a single timestep t, shape loss (L_S) is calculated from the mean squared error between the two vectors y_t, the ground truth FACS parameter values, and y'_t, FACS parameter values predicted by the model.

$$L_S(t) = \frac{1}{N}\sum_{i}^{N}(y_t^i - y_t^{'i})^2 \tag{4.1}$$

In this equation, N is the number of face parameters. y_t^i and $y_t^{'i}$ are ith scalars of y_t and $y_t^{'}$, respectively.

For the calculation of motion loss (L_M), the difference between the ground truth FACS parameter values of two consecutive frames is calculated first, and then the difference between the FACS parameter values of two consecutive predicted frames is calculated. Finally, the motion loss is defined as the mean squared error between these two differences:

$$L_M(t) = \frac{1}{N}\sum_{i}^{N}((y_{t+1}^i - y_t^i) - (y_{t+1}^{'i} - y_t^{'i}))^2 \tag{4.2}$$

The total loss term is defined as the sum of the shape loss term and the motion loss term:

$$L = L_S + L_M \tag{4.3}$$

The model is trained by using the Adam [24] optimizer method. Usually, a training process is performed for 500 epochs. Learning rate is determined as 1e-4 by experimenting with different values, and batch size is determined as 256.

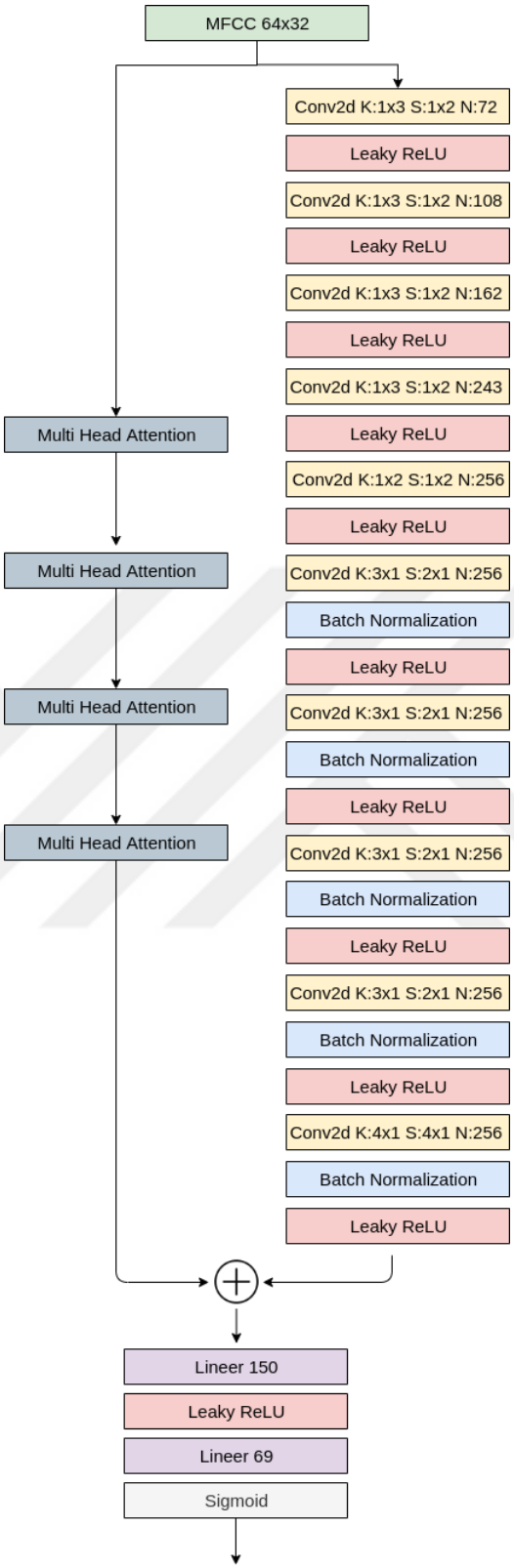

Figure 4.3 : Visualization of the neural network architecture used in the study that uses CNN and transformer components.

5. EXPERIMENTS AND RESULTS

In this section, a brief summary of the tested architectures and their results during the study is presented in the first part. In the second part, the evaluation of our proposed architecture (the architecture with CNN and transformer layers working in parallel) is presented.

5.1 Experimenting with Different Architectures

Different neural network architectures are tested for this study on the generated dataset. During the development of the thesis, a CNN model from Karras et al. [2] is used as the baseline architecture. Other architectures:

1. LSTM Model
2. Wav2Vec2 Model
3. Transformer Model

are created by modifying this baseline CNN architecture appropriately.

5.1.1 Baseline CNN Model

This is the neural network architecture from the work of Karras et al. [2]. The architecture of the model is shown in Table 5.1. It is used as a baseline to generate facial animation videos and compare the results of other models. In the implementation of the framework, there are some minor modifications done to the described model: Its activation functions between layers are changed from ReLU to Leaky ReLU, batch normalization layers are added between convolution layers to normalize input values, and dropout layers are added after convolutional layers to perform regularization of network weights.

Table 5.1 : Neural network architecture of our baseline model from Karras et al. [2].

Baseline CNN Model	
Input Layer	32 MFCC Feature
Formant Analysis Layer	2D Convolution K:1x3, S:1x2, N:144
	2D Convolution K:1x3, S:1x2, N:216
	2D Convolution K:1x3, S:1x2, N:324
	2D Convolution K:1x3, S:1x2, N:243
	2D Convolution K:1x2, S:1x2, N:256
Articulation Layer	2D Convolution K:3x1, S:2x1, N:256
	2D Convolution K:3x1, S:2x1, N:256
	2D Convolution K:3x1, S:2x1, N:256
	2D Convolution K:3x1, S:2x1, N:256
	2D Convolution K:4x1, S:4x1, N:256
Output Layer	Linear 256 Neurons
	Linear 69 Neurons

5.1.2 LSTM Model

In this model, an LSTM layer is used to process sequence data. First neural network layers are changed from CNN layers to LSTM layer as shown in Table 5.2. The input of the model consists of 64 consecutive frames and for each frame, there are 32 MFCC features.

Table 5.2 : The neural network architecture of our LSTM model, feature extraction from the sequence is replaced with LSTM layers instead of CNN layers with appropriate window length.

LSTM Model	
Input Layer	32 MFCC Feature
Formant Analysis Layer	Long Short Term Memory (LSTM) Length of processed sequence: 64 Number of neurons: 256
Articulation Layer	2D Convolution K:3x1, S:2x1, N:256
	2D Convolution K:3x1, S:2x1, N:256
	2D Convolution K:3x1, S:2x1, N:256
	2D Convolution K:3x1, S:2x1, N:256
	2D Convolution K:4x1, S:4x1, N:256
Output Layer	Linear 128 Neurons
	Linear 69 Neurons

5.1.3 Wav2Vec2 Model

Wav2vec2 model by Baevski et al. [25], which is also a transformer based model, is used as the feature extractor. The earlier part of the full architecture is separated and its output is connected directly to the last linear output layers that generate FACS parameters.

At the start of the training process, neural network parameters of the wav2vec2 model are transferred from an instance which is trained on Librispeech [26] dataset consisting of 1000 hours of audio recordings. Since the pretrained model works on the raw audio signal, the raw audio signal is just divided into windows and fed into the neural network without any feature extraction steps when using this architecture.

5.1.4 Summary of All Tested Models

Here, we give a summary of methods/models which have been tested during this study in table 5.3.

Table 5.3 : Comparison of results obtained with different models

Model Number	**Input**	**NN Architecture**	**Description**	**Results**
1	MFCC	CNN	Baseline model	This model can learn a representation from the dataset.
2	MFCC	CNN + LSTM 1	Changed formant analysis layer from Model 1 to LSTM	Capacity of the model seems lower compared to the CNN based model.
3	MFCC	CNN + LSTM 2	Changed output layer from Model 1 to LSTM	Model sometimes could not achieve smooth translations between 520-ms windows.
4	Autocorrelation	CNN	Input of Model 1 changed to autocorrelation features	It is observed that MFCC features produce similar or slightly better results than autocorrelation features.

5	MFCC	CNN, Transformer, Weighted Loss 1	Loss contribution of a few determined action units are increased	It is observed that model can learn selected regions more accurately.
6	MFCC	CNN, Transformer, Weighted Loss 2	Coefficients of shape and motion loss are determined as 1 and 10	Similar to Model 5, this model is heavily affected by changes in voice tone during speech.
7	MFCC	CNN, Transformer	Transformer layer added to Model 1	With added attention-based Transformer layer, the quality of the model is increased compared to that of the baseline.
8	Raw Audio Signal	Wav2Vec2 Base	Model trained on Librispeech dataset used as feature extractor	The model was unable to learn a good representation even from the training dataset. Potential reasons are different audio characteristics between datasets or the lack of different speakers in the study dataset.
9	Raw Audio Signal	Wav2Vec2 Large	The model trained on Librispeech dataset used as feature extractor	Same as Model 8.
10	Mel Spectogram	CNN	Input of Model 1 changed to mel spectogram	Mel spectrogram features have similar representation quality compared to MFCC features, but the resulting animation quality is slightly worse in test audios.
11	Mel Spectogram	CNN, Transformer	Transformer layer is added to Model 10	Same as Model 10.
12	Mel Spectogram	Resnet18	Resnet18 architecture is trained with random weight initialization	The model is unable to generate good-looking animations compared to the baseline model.
13	Mel Spectogram	Resnet18 - pretrained	Resnet18 architecture with weights trained on ImageNet is used	Same as Model 12.

14	Mel Spectogram	Vggish	VGG architecture is trained with random weight initialization	The model is unable to learn a good representation with random initialization.
15	Mel Spectogram	Vggish - pretrained	VGG architecture with weights trained on AudioSet	Although the model is able to learn audio from the train set, videos produced in test audios have errors around different parts of the face model.
16	Mel Spectogram	Vggish - pretrained, Transformer	Transformer layer is added to model 15	Similar to Model 15.

Example results of CNN+Transformer models can be seen at: https://bit.ly/3tmBjRX .

5.2 Evaluation

Since there is no ground-truth information available outside of our training data, we separated last 10 seconds of recordings from the training data and used them as our validation data. There are 11 recordings in the dataset. Since recordings consist of 30 frames per second, validation data has 3300 frames in total. For each frame, we have 69 face parameters.

The rest of this section consists of the evaluation of our transformer-based model.

5.2.1 Perceptual Evaluation

5.2.1.1 Quality of animations for a single speaker

In this user study, the quality of the produced animations for a single speaker is measured. The study is performed by presenting the ground truth animation video and the animation video produced by our network side by side to the participants. In each question, the orders of two videos are shuffled. The participants are asked to choose the video that looks better. There is also a third option "Similar", in case both videos The parts of the face models except the mouth region are cropped from the video so participants can focus only on the mouth movements as seen in Figure 5.1.

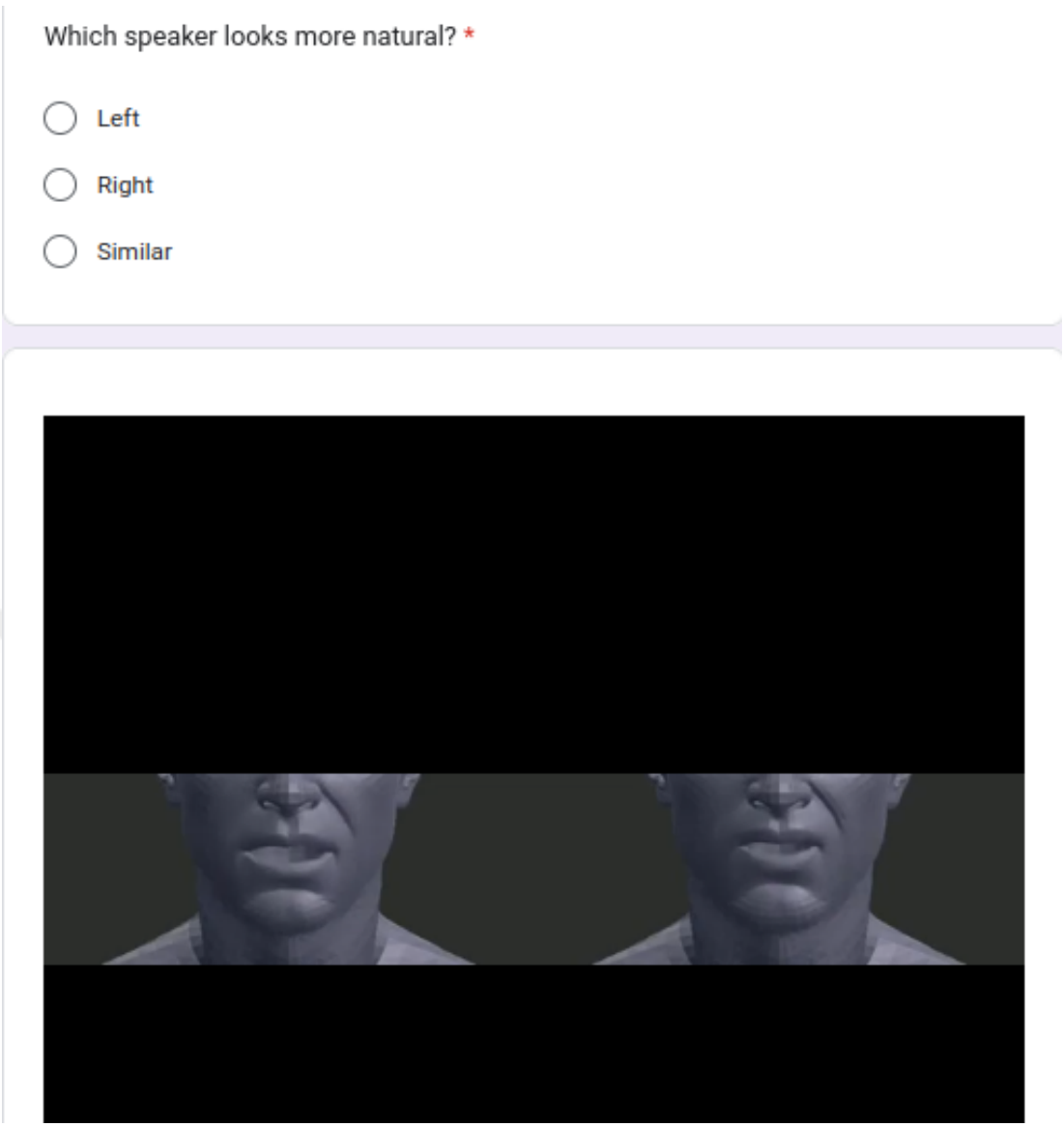

Figure 5.1 : Example question from user study for evaluating the quality of the generated face animation for a single speaker.

Figure 5.2 shows the results of the study. In 22.22% of the cases, the participants preferred the outputs generated by our model, in 45.45% of the cases, the participants preferred the ground truth animation over the neural network output, and in 32.32% of the cases, the results are indifferent to the participants. These results indicate that although it appears that the ground truth has a higher preference rate than the generated facial animation, when we look at the overall results considering the cases where the participants are indifferent between the animations, the artificially generated facial animations with our model are preferable in the majority of the cases.

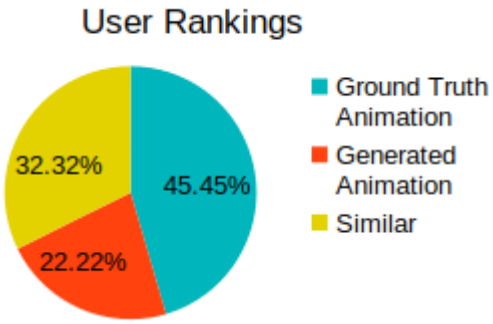

Figure 5.2 : The results of the user study for evaluating the quality of generated face animation for a single speaker. The participants involved in the user study are shown the ground truth animations and the animations produced by our neural network side by side (in random order).

5.2.2 Quantitative Evaluation

Table 5.4 shows the scores obtained for the metrics calculated from the validation data. Error criterion is the L1 distance between the ground truth face parameter values and the face parameter values predicted by the neural network. Accuracy is calculated since we know that predicted values are bounded and always in range of [0, 1].

Table 5.4 : The scores obtained for the metrics calculated from the dataset by comparing ground truth face parameter values and predicted face parameter values.

Recording	Total L1 Error	Accuracy	Duration(s)
All	9459.228	95.48	110
1	811.29	96.08	10
2	877.25	95.76	10
3	628.34	96.96	10
4	1048.99	94.93	10
5	1220.62	94.10	10
6	638.75	96.91	10
7	955.79	95.38	10
8	685.05	96.69	10
9	1136.31	94.51	10
10	702.33	96.60	10
11	754.45	96.35	10

Although numerical values show good results, error in just a few parameters (such as upper lip movement) may cause overall mouth movement to look incorrect, hence numerical evaluation is not completely reliable.

5.2.3 Qualitative Evaluation

5.2.3.1 Adding noise to audio

Similar to Cudeiro et al. [20], audio input used for generating facial animation is combined with a noisy audio input. Used noise is a realistic city noise [27].

Noise audio is added to speech audio with a negative gain of 36dB (low), 24dB (medium), 18dB (slightly high), 12dB (high) and 0dB (hard to hear original speech). Effects of different noise levels were observed as:

- Low level noise does not have an important effect on the result.
- Medium level noise: Sometimes it causes errors in the lip position after words are spoken, as seen in Figure 5.4.
- Slightly high level noise: Results in clearly visible errors in the animation. Transitions from the resting position of the mouth to talking position of the mouth become less accurate.
- High level and full noise: Mouth movement is mostly/completely out of sync.

Unlike Cudeiro et al. [20], damped facial movement start to appear at slightly high level instead of high level. Also they reported that their facial animations still look plausible in slightly high and high levels despite not being accurate. It shows that the proposed system in the study could be improved slightly further if desired.

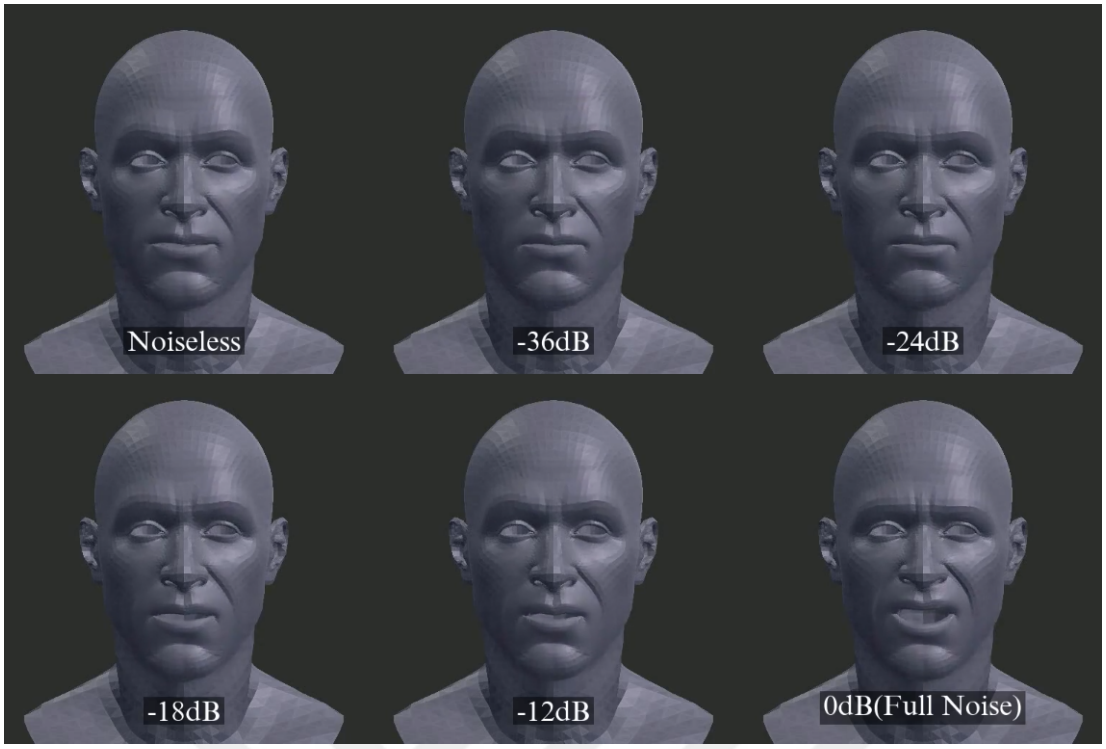

Figure 5.3 : Example animations with different noise levels (noiseless, low level (-36dB), medium level (-24dB), slightly high level (-18dB), high level (-12dB) and full noise from top-left to bottom-right). With low level noise and medium level noise, the mouth correctly returns to the resting position, but with higher level noise movements are incorrect.

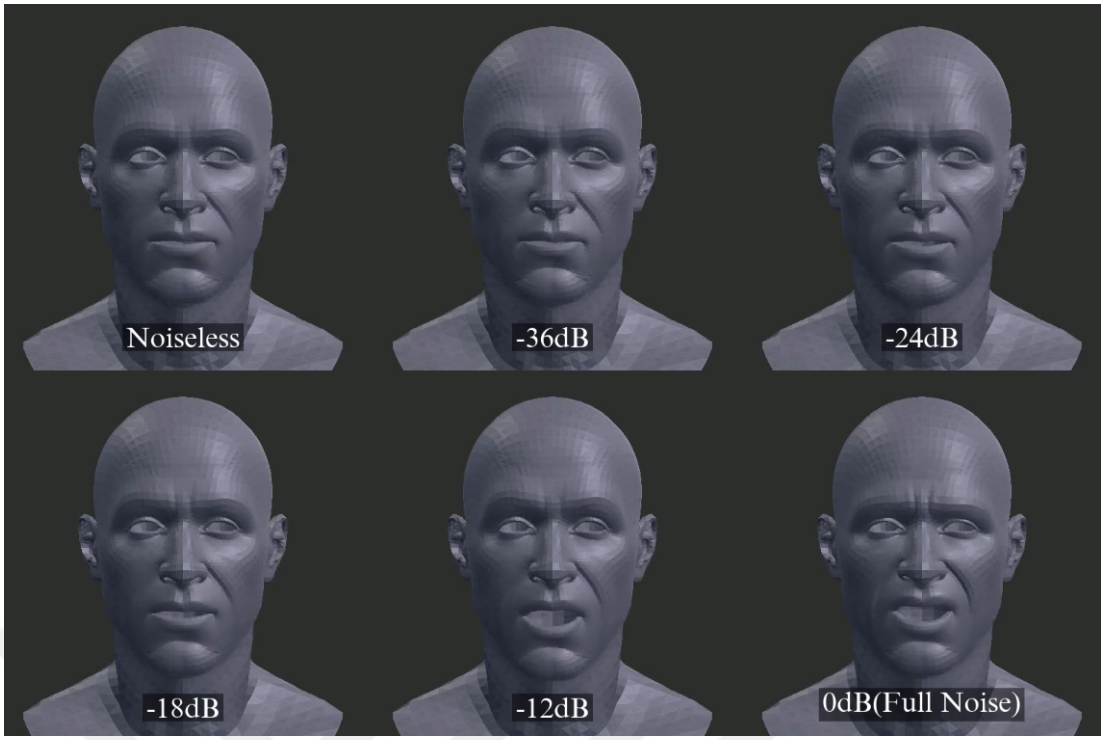

Figure 5.4 : Example animations with different noise levels (noiseless, low level (-36dB), medium level (-24dB), slightly high level (-18dB), high level (-12dB) and full noise from top-left to bottom-right). Animations should appear similar to "noiseless" figure, but errors started to be seen in the medium level noise level.

5.2.3.2 Visualizing network feature layer outputs with different moods

First, we divided validation audio recordings into two classes:

1. Excited/Angry speech recordings

2. Soft/Neutral speech recordings

During the evaluation of these audio recordings, the outputs of feature layers (layers before the last linear layers that predict face parameters) are extracted. For each frame, we obtain a feature vector with length 288. We visualized these feature vectors using t-SNE [28] as seen in figure 5.5:

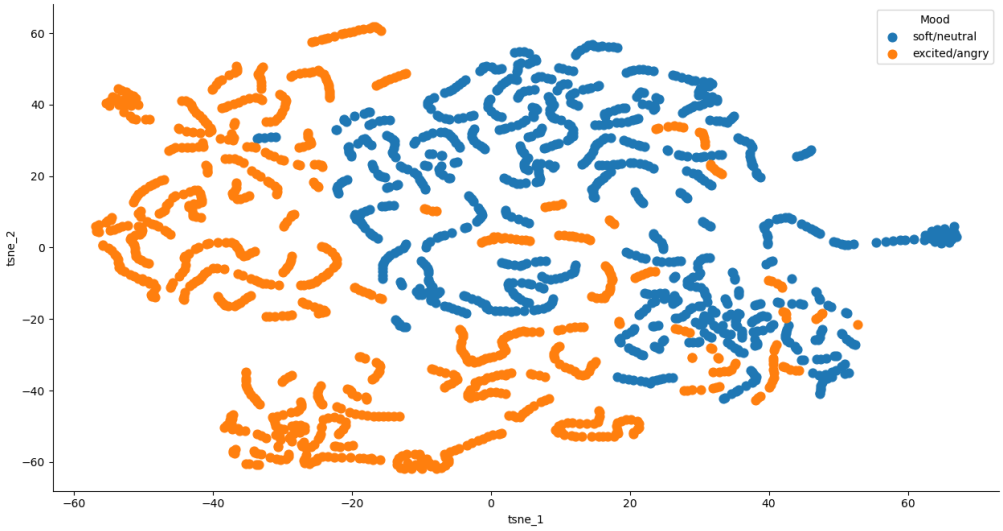

Figure 5.5 : Visualization of feature vectors extracted from audio clips by neural network with transformer component.

It can be seen from these results that the neural network is able to generate feature vectors that allow separation between these two moods. It is possible that high-level features such as the talking speed of the speaker and amplitude of the speaker's voice were the main factor and these information are encoded into these vectors. It can be investigated further if there would be speakers with different voice tones in the dataset.

6. CONCLUSIONS AND FUTURE WORK

In this study, a method is proposed to solve the problem of generating 3D facial animation from audio. Firstly, a face model is defined using FACS parameters. Secondly, a dataset is collected that contains both audio recordings and face parameter values for FACS. In the dataset, there is a total of 37-minute long audio data, and the face parameter values are available at 30 FPS. Finally, a CNN architecture that uses the Transformer component in addition is proposed to generate facial animations from the speech audio signals. As the output face model representation, high-level FACS parameters are used instead of using raw 3D points in the space, thus reducing the size of the underlying machine learning problem.

Before the architecture is finalized with CNN and Transformer, different architectures are tested and their results are presented briefly in the study. The proposed model at the end can produce facial animations with good results for the same speaker it is trained for, including newly encountered speeches of the same actor. Although the results for different voice actors have a lower level of detail, animations look consistent during the speeches.

Future studies about the topic involve:

1. Having speeches from multiple speakers in the dataset allows to measure the generality of the trained model more objectively.

2. Making use of research in the speech recognition literature to improve the generalization of the trained model so it is less affected by speaker differences, and also more robust to noise in the audio.

3. Integrating datasets (with the used face meshes) of similar studies to FACS format to enlarge the training dataset and allow better comparisons of different algorithms on the same dataset.

REFERENCES

[1] **Friesen, W.V.**, **Ekman, P.** *et al.* (1983). EMFACS-7: Emotional facial action coding system, *Unpublished manuscript, University of California at San Francisco*, *2*(36), 1.

[2] **Karras, T.**, **Aila, T.**, **Laine, S.**, **Herva, A. and Lehtinen, J.** (2017). Audio-driven facial animation by joint end-to-end learning of pose and emotion, *ACM Transactions on Graphics (TOG)*, *36*(4), 1–12.

[3] *CS231n Convolutional Neural Networks for Visual Recognition*, `https://cs231n.github.io/neural-networks-1/`, accessed: 2022-06-03.

[4] **Goodfellow, I.**, **Bengio, Y. and Courville, A.** (2016). *Deep learning*, MIT press.

[5] **Vaswani, A.**, **Shazeer, N.**, **Parmar, N.**, **Uszkoreit, J.**, **Jones, L.**, **Gomez, A.N.**, **Kaiser, Ł. and Polosukhin, I.** (2017). Attention is all you need, *Advances in neural information processing systems*, *30*.

[6] **Cohn, J.F.**, **Ambadar, Z. and Ekman, P.** (2007). Observer-based measurement of facial expression with the Facial Action Coding System, *The handbook of emotion elicitation and assessment*, *1*(3), 203–221.

[7] **Ekman, P. and Friesen, W.V.** (1978). Facial action coding system, *Environmental Psychology & Nonverbal Behavior*.

[8] **Rumelhart, D.E.**, **Hinton, G.E. and Williams, R.J.** (1986). Learning representations by back-propagating errors, *nature*, *323*(6088), 533–536.

[9] **Krizhevsky, A.**, **Sutskever, I. and Hinton, G.E.** (2012). Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, *25*.

[10] **Cohen, M.M. and Massaro, D.W.**, (1993). Modeling coarticulation in synthetic visual speech, Models and techniques in computer animation, Springer, pp.139–156.

[11] **Taylor, S.L.**, **Mahler, M.**, **Theobald, B.J. and Matthews, I.** (2012). Dynamic units of visual speech, *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*, pp.275–284.

[12] **Goodfellow, I.**, **Pouget-Abadie, J.**, **Mirza, M.**, **Xu, B.**, **Warde-Farley, D.**, **Ozair, S.**, **Courville, A. and Bengio, Y.** (2014). Generative adversarial nets, *Advances in neural information processing systems*, *27*.

[13] **Wiles, O.**, **Koepke, A. and Zisserman, A.** (2018). X2face: A network for controlling face generation using images, audio, and pose codes, *Proceedings of the European conference on computer vision (ECCV)*, pp.670–686.

[14] **Oh, T.H.**, **Dekel, T.**, **Kim, C.**, **Mosseri, I.**, **Freeman, W.T.**, **Rubinstein, M. and Matusik, W.** (2019). Speech2face: Learning the face behind a voice, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.7539–7548.

[15] **Jamaludin, A.**, **Chung, J.S. and Zisserman, A.** (2019). You said that?: Synthesising talking faces from audio, *International Journal of Computer Vision*, *127*(11), 1767–1779.

[16] **Zhou, Y.**, **Han, X.**, **Shechtman, E.**, **Echevarria, J.**, **Kalogerakis, E. and Li, D.** (2020). Makelttalk: speaker-aware talking-head animation, *ACM Transactions on Graphics (TOG)*, *39*(6), 1–15.

[17] **Vougioukas, K.**, **Petridis, S. and Pantic, M.** (2019). End-to-End Speech-Driven Realistic Facial Animation with Temporal GANs., *CVPR Workshops*, pp.37–40.

[18] **Song, Y.**, **Zhu, J.**, **Li, D.**, **Wang, X. and Qi, H.** (2018). Talking face generation by conditional recurrent adversarial network, *arXiv preprint arXiv:1804.04786*.

[19] **Duarte, A.C.**, **Roldan, F.**, **Tubau, M.**, **Escur, J.**, **Pascual, S.**, **Salvador, A.**, **Mohedano, E.**, **McGuinness, K.**, **Torres, J. and Giro-i Nieto, X.** (2019). WAV2PIX: Speech-conditioned Face Generation using Generative Adversarial Networks., *ICASSP*, pp.8633–8637.

[20] **Cudeiro, D.**, **Bolkart, T.**, **Laidlaw, C.**, **Ranjan, A. and Black, M.J.** (2019). Capture, learning, and synthesis of 3D speaking styles, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.10101–10111.

[21] **Zhou, Y.**, **Xu, Z.**, **Landreth, C.**, **Kalogerakis, E.**, **Maji, S. and Singh, K.** (2018). Visemenet: Audio-driven animator-centric speech animation, *ACM Transactions on Graphics (TOG)*, *37*(4), 1–10.

[22] **Li, T.**, **Bolkart, T.**, **Black, M.J.**, **Li, H. and Romero, J.** (2017). Learning a model of facial shape and expression from 4D scans., *ACM Trans. Graph.*, *36*(6), 194–1.

[23] *Parallel is All You Want: Combining Spatial and Temporal Feature Represention of Speech Emotion by Parallelizing CNNs and Transformer-Encoders*, `https://github.com/IliaZenkov/transformer-cnn-emotion-recognition`, accessed: 2022-06-03.

[24] **Kingma, D.P. and Ba, J.** (2015). Adam: A Method for Stochastic Optimization. ICLR. 2015.

[25] **Baevski, A.**, **Zhou, Y.**, **Mohamed, A. and Auli, M.** (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations, *Advances in Neural Information Processing Systems*, *33*, 12449–12460.

[26] **Panayotov, V.**, **Chen, G.**, **Povey, D. and Khudanpur, S.** (2015). Librispeech: an asr corpus based on public domain audio books, *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, pp.5206–5210.

[27] *City Sounds | Free Sound Effects*, `https://soundbible.com/tags-city.html`, accessed: 2022-07-23.

[28] **Van der Maaten, L. and Hinton, G.** (2008). Visualizing data using t-SNE., *Journal of machine learning research*, *9*(11).

CURRICULUM VITAE

EDUCATION:

- **B.Sc.:** 2017, Istanbul Technical University, Faculty of Computer and Informatics, Department of Computer Engineering
- **M.Sc.:** 2022, Istanbul Technical University, Graduate School, Department of Computer Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2019-2022 Research Assistant at Department of Computer Engineering, Istanbul Technical University

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Ünlü T., İnceoğlu A., Yılmaz E. Ö., Sarıel S.**, Accepted in 2022 30th Signal Processing and Communications Applications Conference (SIU).