Ain Shams University
Faculty of Computer & Information Sciences
Scientific Computing Department

Automated audio to video lip synchronization

By

Ahmed Mohamed Abulfarh
Ahmed Mohamed Wafik
Ahmed Salah Mohamed
Mohamed Ali Salama
Ahmed Mohamed Hassan

Under Supervision of

Dr. Dina Khattab
Scientific Computing Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

TA. Yomna Ahmed
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

TA. Hadeer Hussein
Scientific Computing Department,
Faculty of Computer and Information Sciences
,Ain Shams University.

July 2023

Table of Contents

Acknowledgements

All praise and thanks to ALLAH, who provided the ability to complete this work. We hope to accept this work from us.

We are grateful of *our parents* and *our families* who are always providing help and support throughout the whole years of study. We hope we can give that back to them.

We also offer our sincerest gratitude to our supervisors, *Prof. Dr. Dina Khattab,*
T.A Yomna Ahmed and T.A Hadeer Hussein who have supported us throughout our thesis with their patience, knowledge, and experience.

Finally, We thank our friends and all people who gave us support and encouragement.

Abstract

This work explores the problem of automatically translating a video of a person speaking in one language into another language while maintaining realistic lip synchronization. Existing methods perform well on static images or specific individuals seen during training but struggle with accurately syncing arbitrary identities in dynamic talking face videos. To address this, we used WAV2LIP model that learns from a powerful lip-sync discriminator, resolving key challenges. Extensive quantitative evaluations demonstrate that wav2lip model achieves lip-sync accuracy comparable to real synced videos. Additionally, we fine-tuned the model on two datasets (Lombard grid corpus, grid corpus) one time with same loss function and another time with new loss function called W-GAN and evaluate its performance on a small dataset called (vidtimit), obtaining results close to real videos. To enhance the system's flexibility and applicability in various scenarios, we created face to face translation system and added more functionality to our system that can take different types of inputs such as text/audio with image, or text/audio with video to generate new video synced with the input audio and another pipeline that can take input video in language A to produce another video in language B. This involves leveraging state-of-the-art solutions in speech recognition, speech synthesis, gender detection, and machine language translation.

الملخص:

يستكشف هذا العمل مشكلة الترجمة التلقائية لفيديو لشخص يتحدث بلغة ما إلى لغة أخرى مع الحفاظ على التزامن الواقعي للشفاه. تؤدي الأساليب الحالية أداءً جيدًا على الصور الثابتة أو الأفراد المحددين الذين شوهدوا أثناء التدريب ولكنهم يواجهون صعوبة في مزامنة الهويات التعسفية بدقة في مقاطع فيديو الوجه الناطق الديناميكي. لمعالجة هذا الأمر ، استخدمنا نموذج WAV2LIP الذي يتعلم من أداة تمييز مزامنة الشفاه القوية ، مما يحل التحديات الرئيسية. تُظهر التقييمات الكمية الشاملة أن نموذج wav2lip يحقق دقة مزامنة الشفاه مقارنة بمقاطع الفيديو الحقيقية المتزامنة. بالإضافة إلى ذلك ، قمنا بضبط النموذج على مجموعتي بيانات (Lombard Grid Corpus , Grid Corpus) مرة واحدة مع نفس وظيفة الخسارة ومرة أخرى مع وظيفة خسارة جديدة تسمى W-GAN وتقييم أدائها على مجموعة بيانات صغيرة تسمى (vidtimit) ، والحصول على نتائج قريبة من مقاطع الفيديو الحقيقية. لتعزيز مرونة النظام وقابليته للتطبيق في مختلف السيناريوهات ، أنشأنا نظام ترجمة تعمل على ترجمة الصوت وحركات التعبيرات وأضفنا المزيد من الوظائف إلى نظامنا الذي يمكن أن يأخذ أنواعًا مختلفة من المدخلات مثل النص / الصوت مع الصورة ، أو النص / الصوت مع الفيديو لإنشاء فيديو جديد متزامنة مع صوت الإدخال وخط أنابيب آخر يمكن أن يأخذ إدخال فيديو باللغة أ لإنتاج فيديو آخر بلغة ب. وهذا يتضمن الاستفادة من أحدث الحلول في التعرف على الكلام وتركيب الكلام والكشف عن الجنس وترجمة اللغة الآلية.

List of Figures

List of Tables

List of Abbreviations

ADAM	Adaptive moments estimation, optimizer
ANN	Artificial neural network
ASR	automatic speech recognition
CNN	Convolutional neural network
DCGAN	Deep convolutional generative adversarial network
DGM	Deep generative models
FFT	Fast Fourier transform.
FID	Fréchet inception distance
GAN	Generative adversarial network
GPU	Graphics processing unit
MFC	Mel-frequency cepstrum
MFCC	Mel-frequency cepstral coefficient
MLP	Multi-layer perceptron
S3FD	Single shot scale-invariant face detector
NLP	Natural language processing
PSNR	Peak signal-to-noise ratio
ReLU	Rectified linear unit.
SGD	Stochastic gradient descent
SSH	Secure Shell Protocol
SSIM	Structural similarity index measure
TTS	Text-to-speech
VAE	Variational autoencoders
WGAN	Wasserstein generative adversarial network
WGAN-GP	Wasserstein generative adversarial network - gradient penalty
LSE-loss	Lip sync error distance.
LSE – C	Lip sync error confidence.

Chapter One

Chapter 1: Introduction

With the exponential rise in the consumption of audio-visual content, rapid video content creation has become a quintessential need. At the same time, making these videos accessible in different languages is also a key challenge. For instance, a deep learning lecture series, a famous movie, or a public address to the nation, if translated to desired target languages, can become accessible to millions of new viewers.

When a person tries to listen to important conference for one hour or more the un-synced video with corresponding dubbed audio makes the listener feel bored. A crucial aspect of translating such talking face videos or creating new ones is correcting the lip sync to match the desired target speech. Consequently, lip-syncing talking face videos to match a given input audio stream has received considerable attention in the research community. Initial works using deep learning in this space learned a mapping from speech representations to lip landmarks using several hours of a single speaker. More recent works in this line directly generate images from speech representations and show exceptional generation quality for specific speakers which they have been trained upon.

1.1 Motivation

The motivation to incorporate a visual module into a translation system because most of the information stream today, is increasingly becoming audiovisual. Also, it can be used in many applications such as translating a lecture/TV series, dubbing movies to any language, creating cut scenes for the 3d industry including human characters, translation of important conferences, generate missing video segments and making of visual chatbots. The greatest thing is that with the solution of that problem it can help us to generate lip motion on a single, static image of any identity in any voice which can be used for creating social media content.

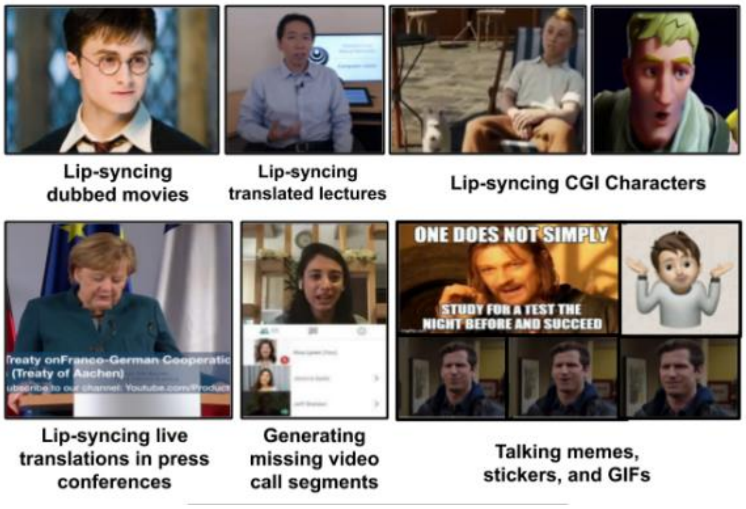

Lip-syncing dubbed movies
Lip-syncing translated lectures
Lip-syncing CGI Characters
Treaty on Franco-German Cooperatic
(Treaty of Aachen)
ONE DOES NOT SIMPLY
STUDY FOR A TEST THE NIGHT BEFORE AND SUCCEED
Lip-syncing live translations in press conferences
Generating missing video call segments
Talking memes, stickers, and GIFs

Figure 1.1: Applications

1.2 Objectives

The goal of this project is to develop an application to take a recorded voice of a speaker and any target video of a speaking person as input, and then to generate a new lip-synchronized video - by changing mouth movements of the target video to match the input . This can promote accessibility to a greater portion of the public. We are extending by making a complete translation system that is able to synch voice from video in language A to a new video in language B.

Time Plan:

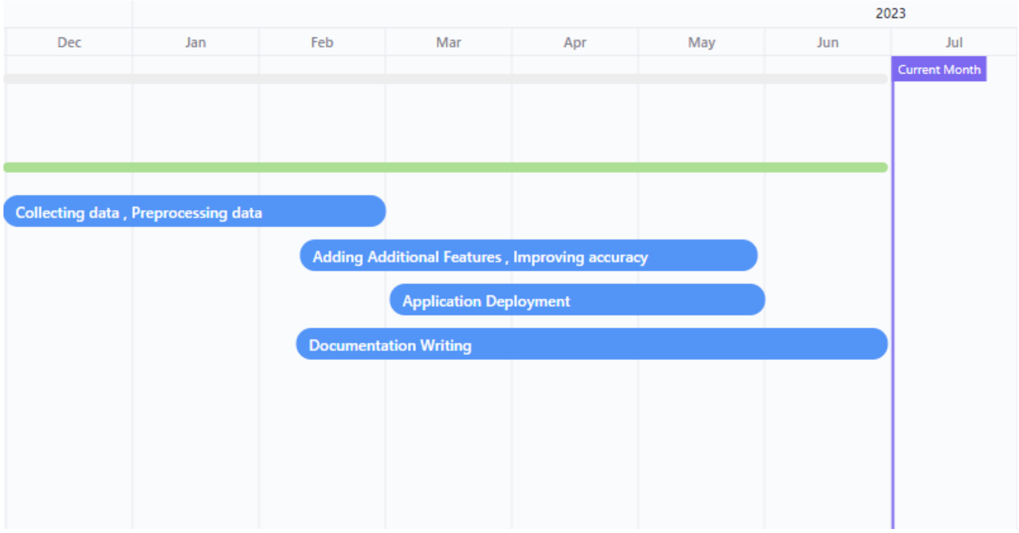


Figure 1.2 Time Table

1.3 Document Organization

In **Chapter 2,** we delve into the background of our research topic, providing a comprehensive overview of key concepts and techniques. We begin by introducing the foundations of deep learning, a powerful subfield of machine learning that has revolutionized various domains. We discuss the Convolutional Neural Network (CNN), a deep learning architecture specifically designed for visual data processing. Additionally, we explore Generative Adversarial Networks (GANs), an innovative framework that enables the generation of realistic synthetic data. Furthermore, we address the crucial topic of face detection, discussing a single specific algorithm that is employed to accurately identify and locate human faces in images and videos. In the realm of audio processing, we examine how to extract features from audio data. Lastly, we provide a comprehensive literature review, summarizing and analyzing the findings of previous research papers and studies in the field. This chapter establishes a solid foundation of knowledge that will inform the subsequent chapters and guide our research methodology.

In **Chapter 3,** we focus on the system architecture and its functioning. This chapter begins by presenting a comprehensive overview of the system architecture, providing detailed descriptions and explanations of its various components and their interactions. To aid in understanding, several diagrams are included to visually represent the architecture and highlight its key aspects. Following the discussion on system architecture, the chapter delves into the features that have been incorporated into the project. These features are carefully designed to enhance the functionality and user experience of the system. Throughout this chapter, we aim to provide a comprehensive understanding of the system architecture, its functioning, and the added features.

In Chapter 4, our focus shifts towards how training is done, dataset preparation, fine-tuning the model and evaluating its performance on different datasets. We begin by describing the various techniques and methodologies employed in fine-tuning, considering aspects such as transfer learning, hyperparameter optimization, and changing loss function. Detailed pseudo-codes of the crucial steps are provided, offering readers a clear understanding of the implementation process. Moreover, we discuss various evaluation metrics and techniques utilized to measure the model's effectiveness. Then, we highlight the notable features that have been incorporated into the model. These features are explained in detail, elucidating their purpose, functionalities, and the added value to our project.

In **chapter 5,** we show how to run the application and use all the different features in it.

In chapter 6, we summarize the findings and our experiments with the ending results after adding the new features and the possible future work.

Chapter Two

Chapter 2: Literature Review

2.1 Background

In this section, the needed background for neural network, face detection and Generative adversarial Network (GANs) will be discussed.

2.1.1 Artificial Neural Networks

Single layer Perceptron

Artificial neural networks are, much like the human brain, made up of several small calculating units. These units can be connected both in series and in parallel to form dense networks. The most basic ANN consists of only input, one calculating unit, and output. This network is called a **perceptron**[1] and can be used to solve problems like linear regression. Though it is almost trivial in its structure, it can be used to show the theory of neural networks.

Figure 2.1 shows a Visualization of a perceptron with input x, bias x0 and, output y.

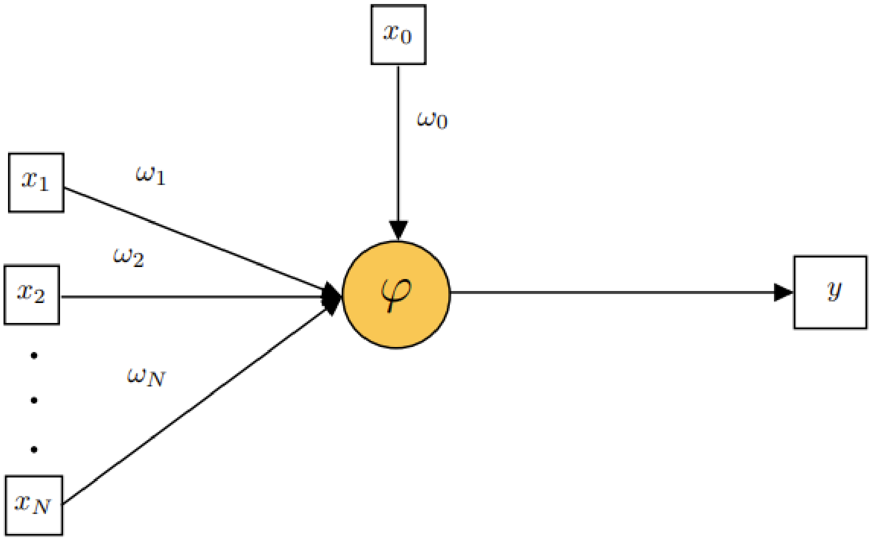


Figure 2.1 Single Layer Perceptron [1]

$$y = \varphi\left(\sum_{n=1}^{N} \omega_n x_n + \omega_0\right)$$ eq(1)

where x1, x2, ..., xN are inputs to the network and x0 = 1 is a bias node, included to prevent the weights from being all zeros. The weights of an ANN are denoted as ωn and signifies the adjustable parameters in the network. As training commences it is the weights that get an updated value and thus represent the experience of the neural network. Lastly, ϕ is known as the activation function [2]. A vast number of activation functions exist and are used in different scenarios, such as the Sigmoid, Tanh, Soft plus and ReLU [2]. The latter is a popular choice in deep learning [2], since it is computational efficient which is important in a dense network. The ReLU, or Rectified Linear Unit, is defined for a scalar input a as ReLU(a) = max (0, a).

Deep learning is a subfield of machine learning that focuses on training deep neural networks with multiple layers. It leverages these deep architectures to automatically learn hierarchical representations of data directly from raw input. The term "deep" refers to the depth of the network, indicating the presence of multiple layers between the input and output layers.

Multi-layer Perceptron:

A single perceptron is most often not desired since it is only capable to perform simple tasks. More complex problems can be solved by increasing either the number of hidden layers or the number of hidden nodes. When the number of hidden layers is larger than one, the network is called a multi-layer perceptron (MLP), and is shown in figure 4: Fully connected multi-layer perceptron with m hidden nodes and n hidden layers, N inputs, and M outputs.

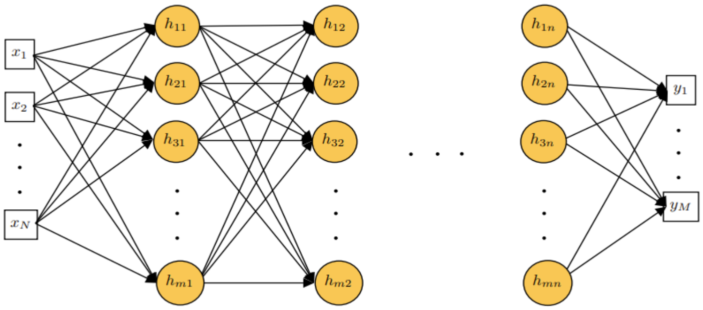


Figure 2.2 MLP Diagram[2]

The power of deep learning lies in its ability to learn complex patterns and features from raw data without the need for explicit feature engineering. This is achieved through a process called backpropagation, which involves iteratively adjusting the network's parameters based on the gradient of a loss function. The backpropagation algorithm calculates the impact of each parameter on the overall prediction error, allowing the network to update its parameters and improve its performance over time.

The availability of large, labeled datasets and advancements in computational hardware, particularly the use of GPUs, have played a significant role in the success of deep learning. The use of large-scale neural networks and training on massive amounts of data has led to breakthroughs in various domains, including computer vision, natural language processing, speech recognition, and many others.

Convolutional neural networks (CNNs)

Are commonly used in computer vision tasks, as they can effectively learn spatial hierarchies of visual features. Transformer models, introduced in recent years, have demonstrated exceptional performance in tasks involving sequential and structured data, such as machine translation and language generation.

While deep learning has achieved remarkable success, it also faces challenges. Deep neural networks require substantial amounts of labeled training data to generalize well, and collecting and annotating such data can be time-consuming and expensive. Deep learning models are also computationally intensive, requiring powerful hardware and significant training times. Additionally, the interpretability of deep learning models can be challenging, as their complex architectures make it difficult to understand the reasoning behind their predictions.

Nonetheless, deep learning continues to be a rapidly evolving field, driving advancements in artificial intelligence and enabling breakthroughs in various applications. Its ability to automatically learn intricate representations from raw data holds great promise for solving complex problems and pushing the boundaries of AI.

The main operation in a convolutional neural network is convolution which is defined, for two dimensions as:

$$\mathcal{H}(i,j) = (X * K)(i,j) = \sum_{m=1}^{M}\sum_{n=1}^{N} X(i,j)K(i-m, j-n)$$

eq(2)

where X is an input matrix, K is the kernel matrix and the output H, is called a feature map. If the convolution is used in the context of images, then X would be an image with dimensions M ×N. Convolution in this case can be viewed as the kernel floating across the image and collecting data in a region combining it into a feature [11]. A visualization of this can be seen in figure 5.

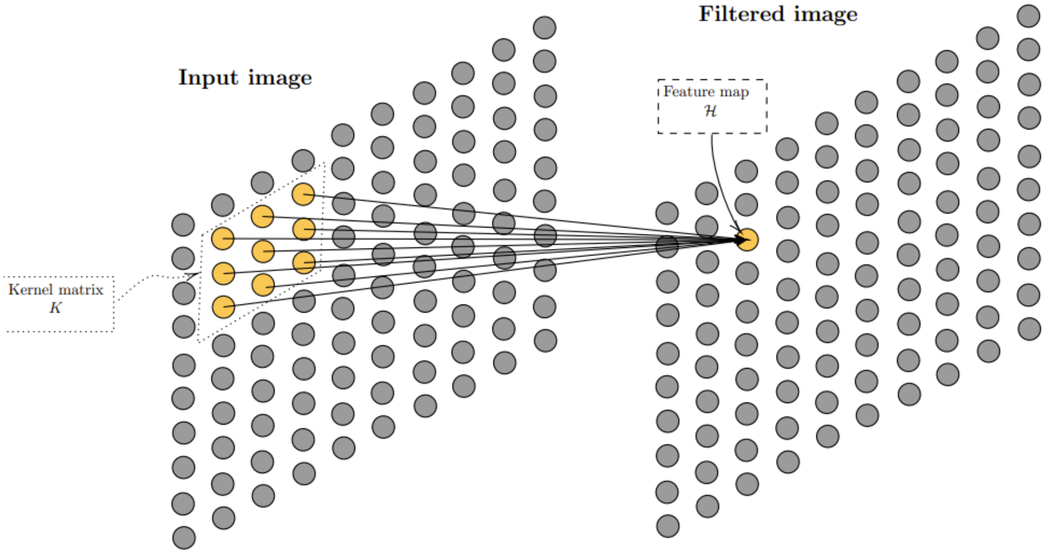


Figure 2.3 : CNN [2]

During a forward pass in neural networks, such as the MLP, the hidden layers use matrix multiplications to traverse from input to output. In this case, all inputs interact with all outputs and thus creates a network with many parameters. If a CNN is used, the kernel matrix K can be made smaller than the input which leads to sparse connectivity. This lowers the number of parameters in the network, while still being able to preserve the features of the data [2]. This enhances the computational complexity of a CNN in comparison to other network types.

The results of a CNN with many hidden layers, a deep convolutional neural network, is a feature map that is compressed to low dimensions. Since the convolution is a linear operation the CNN process can be inversely calculated by using the transpose of the matrix defined by the convolution [2]. The kernel of the CNN produces a down-sample of the input size, and the transpose convolution corresponds to an up-sample of the data size.

2.1.2 Face Detection

Face detection is detecting where the face is in the image and to draw a bounding box around it and it will be needed to crop the face portion of the image to modify lip movements.

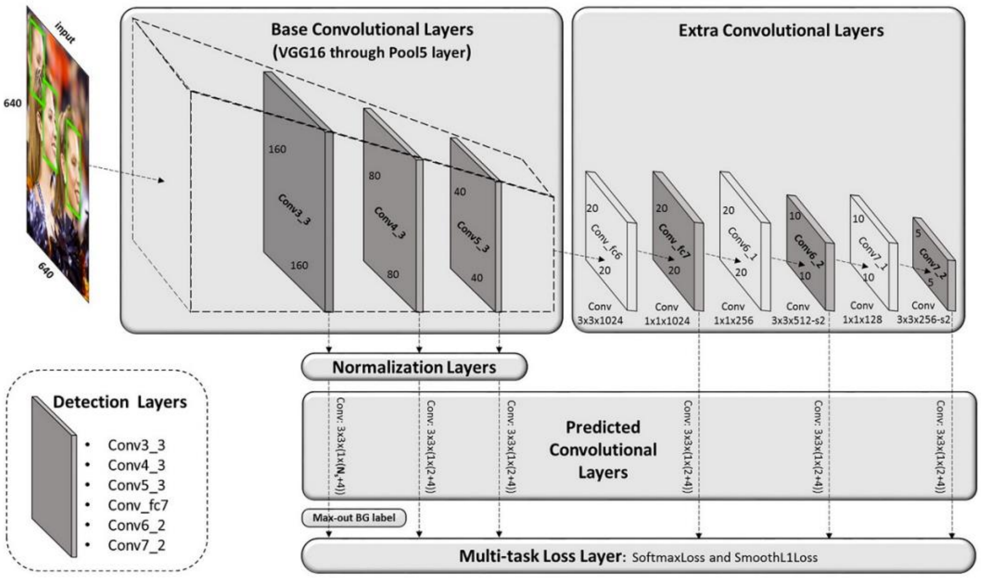


Figure 2.4: S3FD Model

The s3fd face detection model works by using pretrained vgg16 with some extra convolutional layers to predict faces in image by taking features at different levels of convolution and predict faces at every level to detect small faces and large faces with high accuracy.

with pre-trained weight to determine face characteristics instantaneously for the provided input and video pictures. Once a conductive clip of the head, is provided as input, it utilizes the VGG16 architecture of the S3FD model along with extra-convolution layers that construct multi-scale maps, facial feature identification and anchor layers, which are allocated to various anchor measures to anticipate intercepts that establish a sustainable approach of pre-processing magnitude. The input also has higher chances of face detection as the max-out label builds an broad range of anchor points at everylayer, which is the max-out compensation anchor approach.

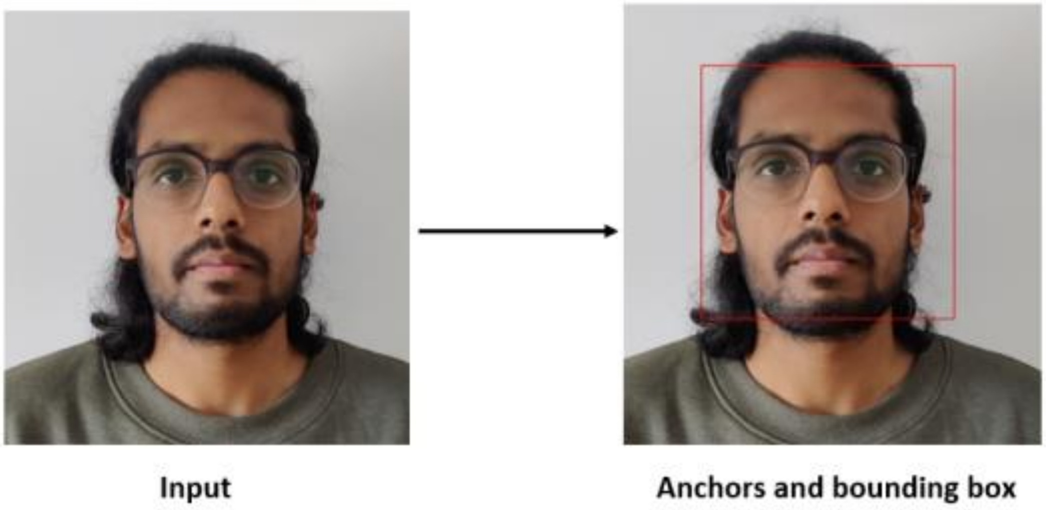


Figure 2.5: face detection example [2]

There is reason for figure 2.5 to predict the bounding box for the input sequence is because the anchor ratio of 1:1 and the recognition layer plays a key function. After mapping the facial expressions to the anchor technique, every face in the bounding box is combined with the best overlapping Jaccard threshold of 0.35 instead of 0.5 to enhance the aggregate quantity of anchors. In general, as compared with the face anchors, the backdrop has more anchors. To minimize the false positive rate of little faces, implement the max-out background labelling for the recognition model process.

2.1.3 Audio representation

Speech is an essential part of lip-synchronization since it determines the lip movements of a person. Human speech is produced by the vocal cords in the form of vibrations that propagates as an acoustic wave, which is known as audio. As audio and speech are time-continuous waves, they need to be quantified into numerical values. This process must preserve the contents as well as the perceptual features. For audio analysis, mel-frequency cepstral coefficients (MFCC) and mel-frequency cepstrum (MFC), more commonly known as mel-spectrogram, have been frequently used to represent speech and audio [21, 22].

These representations have also been widely used in different machine learning projects by, for example, Chen et al. [8] and by Prajwal et al. [9]. Mel frequency cepstral coefficients are simply the coefficients that make up a mel-frequency cepstrum. This method of representing audio was proposed by Mermelstein [23] in the context of speech recognition. The idea of the mel-frequency cepstrum is to give a compact feature preserving representation of audio by creating a spectrogram with the mel-scale on its frequency axis. The mel-scale, or melody scale, is a non-linear scale developed to represent the perceptual scale of pitches. The human audible spectrum ranges from around 20 Hz to 20000 Hz. A specific difference of frequency in the lower side of this spectrum is more clearly audible than a difference in the higher range of the spectrum. Thus, the equal distances between pitches are non-linear and given by the mel-scale [21]. To produce the MFC, and by extension the MFCCs, the audio signal is cropped into T equal spaced time windows of audio that then are transformed to the frequency domain by for example fast Fourier transform (FFT). The frequency spectrum produced is then split into M equally spaced channels, according to the mel-scale.

The MFCCs are obtained by choosing the lowest amplitudes of the spectrum. However, Purwins et al. [21] advises against this due to information and spatial relations being destroyed. Yet, MFCCs does have some merit in creating models when compressed data is required as shown in **figure 2.6**

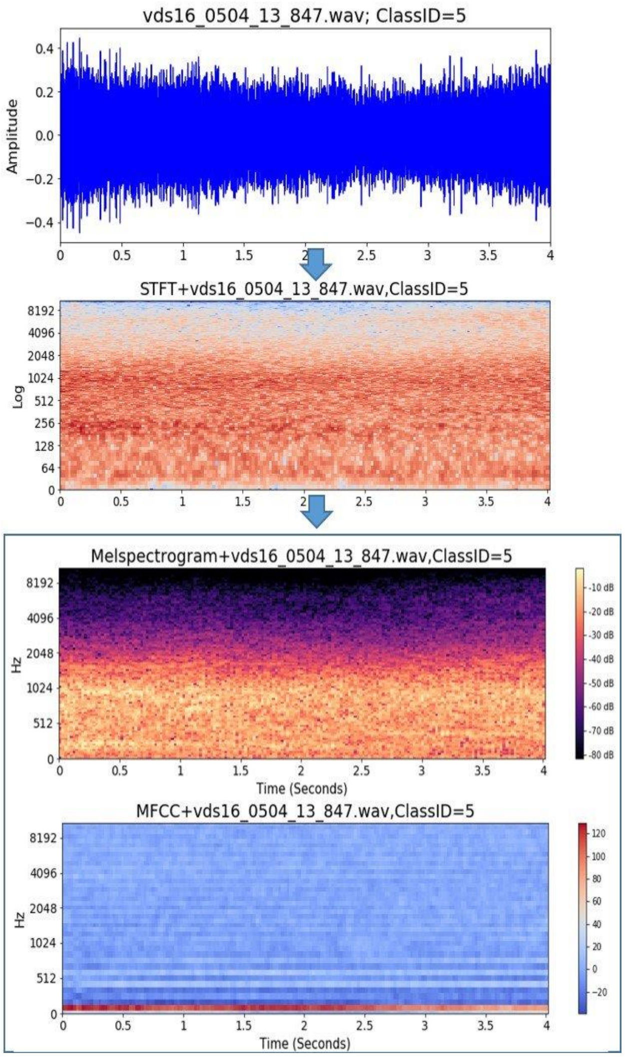


Figure 2.6: MFFC Feature Extraction

2.1.4 Generative adversarial Network (GANs)

Generative Adversarial Networks (GANs) are a type of machine learning framework that consists of two neural networks, a generator and a discriminator, which work in a competitive manner. GANs are used to generate new data that resembles a given training dataset. To understand GANs, let's use the analogy of an art forger and an art critic. The generator network plays the role of the forger, creating fake paintings, while the discriminator network acts as the critic, trying to distinguish between real and fake artwork.

Here's how GANs work: 1. Generator: The generator network takes random noise as input and transforms it into data, such as images or text. In the art analogy, it creates paintings based on its understanding of the style and characteristics of real artwork. 2. Discriminator: The discriminator network takes both real and fake data as input and tries to classify them correctly. In the art analogy, it determines whether a painting is real or fake, trying to identify the work of the forger. 3. Training process: Initially, both the generator and discriminator are not very good at their respective tasks.

The generator produces poor-quality data, and the discriminator makes incorrect classifications. 4. Adversarial learning: The generator and discriminator are trained in an adversarial manner. The generator aims to generate data that the discriminator cannot differentiate from real data. The discriminator, on the other hand, tries to improve its classification accuracy and correctly distinguish real and fake data. 5. Iterative improvement: The generator and discriminator are trained iteratively. In each iteration, the generator creates new fake data, and the discriminator is trained on a mixture of real and fake data. The discriminator's feedback is used to update the generator, and the process continues.

Through this iterative competition, the generator and discriminator learn from each other, gradually improving their performance. Ideally, the generator becomes adept at producing realistic data, while the discriminator becomes skilled at distinguishing real from fake. Once the GAN is trained, the generator can be used independently to create new data that resembles the training dataset. For example, a GAN trained on a collection of cat images could generate new and realistic cat images that were not part of the original dataset. Generative Adversarial Networks have been successfully applied in various domains, including image synthesis, text generation, music composition, and more shown in **figure 2.7** . They offer a powerful framework for generating new and realistic data by leveraging the competitive interplay between the generator and discriminator networks.

When a discriminative model is not separating data by labels and instead returns a scalar value it is known as a critic [25], such as in a Wasserstein Generative Adversarial Network (WGAN).

for illustration. The similarity, or distance, between these distributions, is complicated to measure and thus it is hard to determine when satisfying results are obtained [26] as shown in **figure 2.8** and.

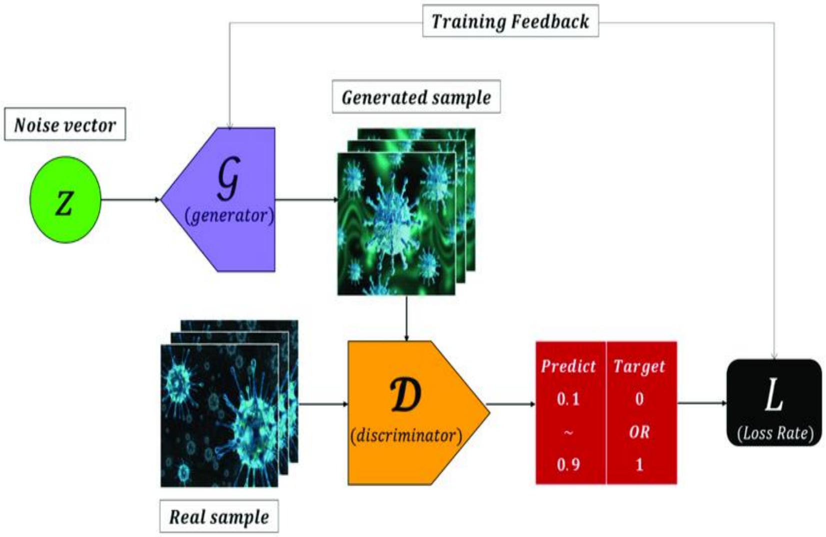


Figure 2.7: Example Of Synethis Images

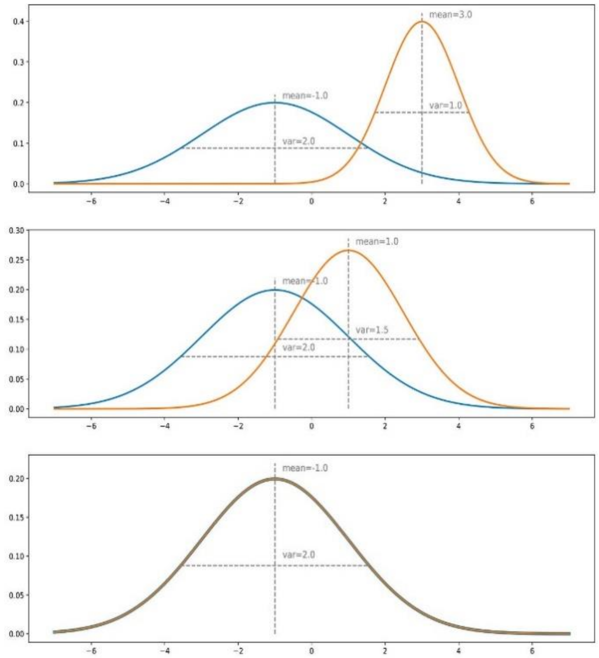


Figure 2.8: Distribution Learning

More formally, the progress of a GAN can be expressed as the following. A noise variable z ~ Pz is used as input into the generator and thus mapped to data space as G(z) where G is some feed-forward ANN. Pz is a random noise distribution, for example, Gaussian noise. A sample x is drawn from either the distribution formed by the generator, Pg or the real distribution Pr. This is used as input into the other network of the GAN, the discriminator, to form D(x). Where D(x) represents the probability of x originating from either Pg or Pr. See **figure 2.9** for a schematic sketch of a GAN.

Wasserstein Generative Adversarial Network

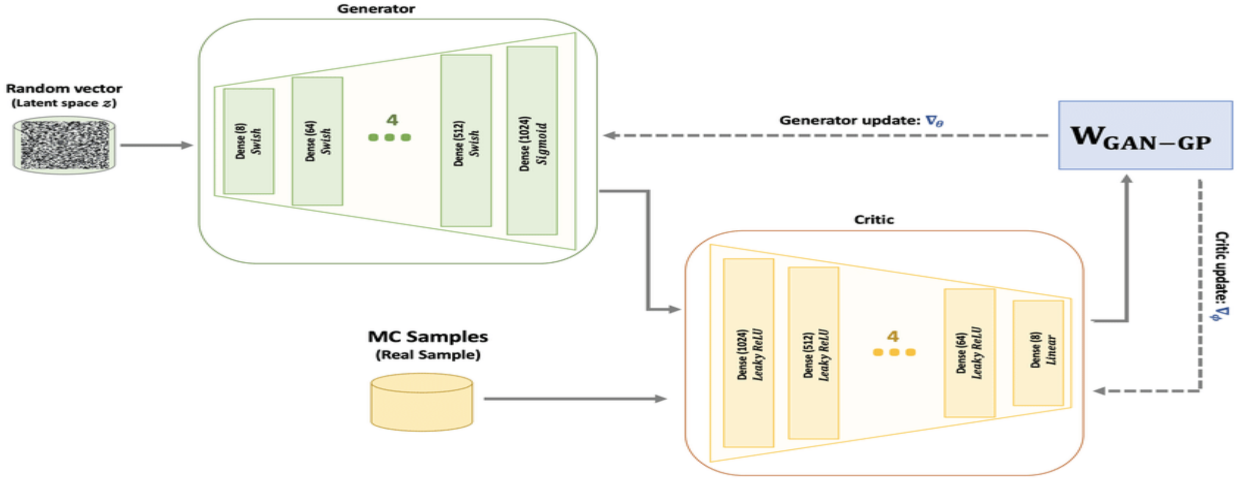


Figure 2.9: WGAN Architecture

GANs are prone to instabilities during training and to mode collapse, and hence, attempts have been made to mitigate this. One of the most notable is the Wasserstein Generative Adversarial Network (WGAN), which was introduced by Arjovsky et al. in 2017. Unlike classical GANs which utilize a discriminatory approach of binary classification, WGANs take another path of trying to minimize the statistical distance between the generated data distribution Pg and the real data distribution Pr . Therefore, a WGAN consists of a generator G and a critic D, and not a discriminator as in the original GAN. However, the two-player zero-sum game from the original GANs persists, with slight modifications to the switching intervals between generator and critic during training . The first remedy to the instabilities is to have a GAN loss function that is continuous everywhere and differentiable almost everywhere. One cost function where this hold is the statistical measurement earth mover’s distance (EMD), also known as the Wasserstein-loss ,Consider the generated data distribution Pg and the real data distribution Pr which forms the marginals for the joint distribution Γ(Pr, Pg). Then the optimal transportation plan to move Pg from its support to Pr and its support, is given by the EMD .

2.2 Related Work

In this section, several related works that discuss the lip sync problem will be reviewed.

In [1], the goal of the authors of SyncNet[1] was to determine the audio-video synchronization between mouth motion and speech in a video by using two-stream ConvNet architecture that enables a joint embedding between the sound and the mouth images to be learnt discriminatively from unlabeled data. Their objective was to develop a language independent and speaker independent solution to the lip-sync problem, using only the video and the audio streams. Architecture: The network consists of two asymmetric streams for audio and video. Sync net's layer architecture based on VGGM, but with modified filter sizes to ingest the inputs of unusual dimensions. The architecture take two inputs which is audio data after extracting MFCC values, and a sequence of mouth regions as grayscale images related to the audio data.

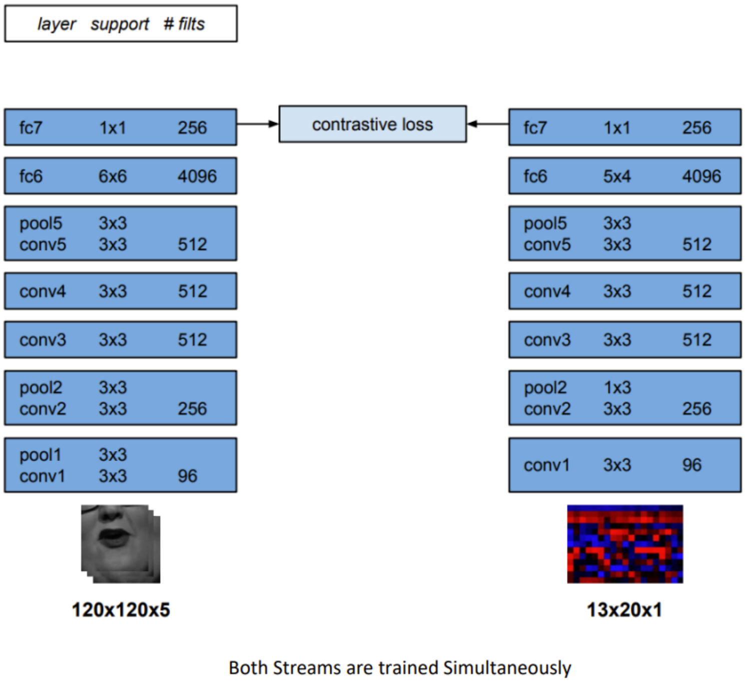


Figure 2.10 Architecture of sync net model[1]

In [4], the challenge of video-to-video translation, specifically focusing on visual speech generation. While image-to-image translation using generative adversarial networks (GANs) has seen remarkable success, there have been limited attempts in the domain of video translation. The goal of the task is to convert an input video of a spoken word into an output video of a different word. This task involves multiple domains, with each word representing a domain of videos uttering that word.

The authors highlight the limitations of adapting the state-of-the-art image-to-image translation model, StarGAN, for this video-to-video translation task, particularly when dealing with a large vocabulary size. Instead, they propose a novel approach called Visual Speech GAN (ViSpGAN) that utilizes character encodings of the words. ViSpGAN is specifically designed for video-to-video translation and addresses the challenge of working with a vocabulary of 500 words. The authors demonstrate that the synthetic samples generated by ViSpGAN exhibit strong visual speech features, comparable to real videos, as evaluated by an auxiliary lipreading network. One key advantage of the proposed model is its ability to generate outputs of any length. By applying the generator sequentially on a long video, it becomes possible to produce multiple words based on their characters. This opens up possibilities for generating videos of full sentences, representing an intriguing avenue for future exploration.

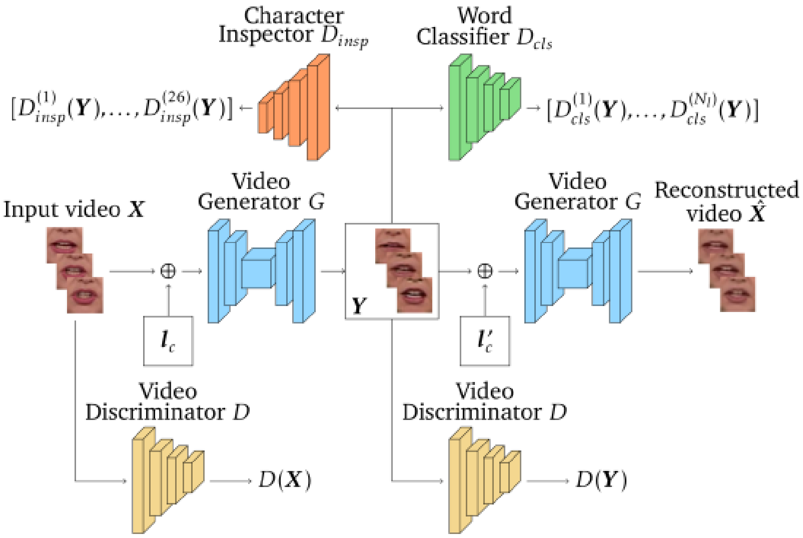

Character Inspector D_{insp}
Word Classifier D_{cls}
$[D^{(1)}_{insp}(\mathbf{Y}),\ldots,D^{(26)}_{insp}(\mathbf{Y})]$
$[D^{(1)}_{cls}(\mathbf{Y}),\ldots,D^{(N_l)}_{cls}(\mathbf{Y})]$
Input video $\mathbf{X}$
Video Generator G
$\mathbf{Y}$
Video Generator G
Reconstructed video $\hat{\mathbf{X}}$
$\mathbf{l}_c$
$\mathbf{l}'_c$
Video Discriminator D
Video Discriminator D
$D(\mathbf{X})$
$D(\mathbf{Y})$

Figure 2.11: VIsSGAN Architecture[4]

In [5], the authors focuse on the challenging task of generating realistic sequences of talking faces that correspond to speech clips. Existing methods either create specific face appearance models for particular subjects or model the relationship between lip motion and speech. However, this work integrates both aspects and achieves arbitrary-subject talking face generation by learning disentangled audio-visual representations. The researchers find that talking face sequences consist of subject-related and speech-related information, which are explicitly disentangled through a novel associative-and-adversarial training process. This disentangled representation allows both audio and video inputs for generation. The approach generates realistic talking face sequences on arbitrary subjects, with clearer lip motion patterns compared to previous methods. The learned audio-visual representation is also useful for automatic lip reading and audio-video retrieval tasks. In this paper, the authors introduce a novel framework called Disentangled Audio-Visual System (DAVS) for generating high-quality talking face videos using disentangled audio-visual representation. This approach involves learning a joint audio-visual embedding space that captures discriminative speech information by utilizing word-ID labels. The authors then disentangle this space from the person-ID space through adversarial learning. DAVS offers several advantages over previous methods including learning a joint audio-visual representation through audio-visual discrimination, leading to improved lip reading performance, unifying audio-visual speech recognition and audio-visual synchronization within an end-to-end framework and Most importantly, this framework enables the generation of high-quality talking face videos for arbitrary subjects, with accurate temporal alignment. Both audio and video speech information can be used as input guidance.

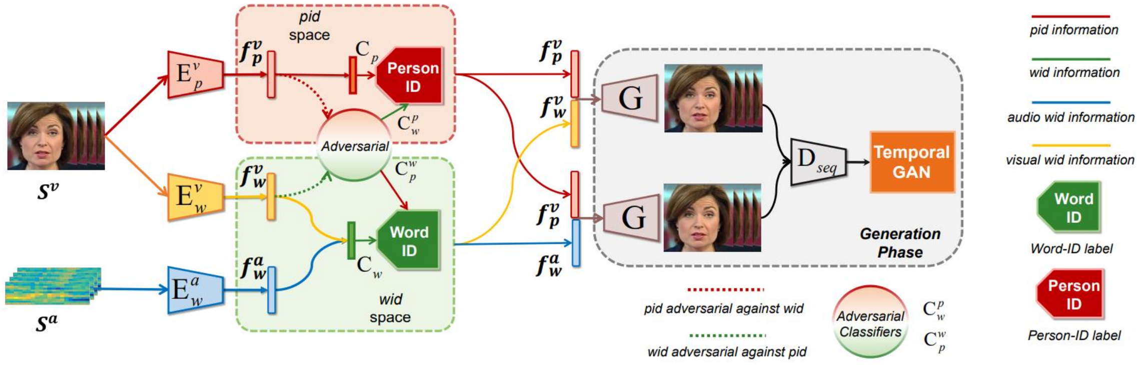


Figure 2.12: DAVS architecture[5]

In [6], the authors present a framework for generating talking faces with accurate lip synchronization and efficient head pose control. Unlike previous methods that rely on pre-estimated structural information, our approach operates on non-aligned raw face images using a single photo as an identity reference. they achieve this by modularizing audio-visual representations and introducing an implicit low-dimensional pose code. By learning the intrinsic synchronization between audio and visual modalities, they defined speech content information. Additionally, they use a modulated convolution-based reconstruction framework to learn a complementary pose code. This method produces talking faces that are both lip-synced and controllable in terms of head pose using other videos. It also exhibits robustness to extreme viewing angles and can perform talking face frontalization.

They introduced the Pose-Controllable Audio-Visual System (PC-AVS) for generating talking faces that exhibit accurate lip synchronization and can be freely controlled in terms of pose using other videos. Several attractive features of our framework that they do not rely on any intermediate structural information but instead devise an implicit pose code, they modularize the audio-visual representations into latent identity, speech content, and pose spaces. that learning procedure ensures more precise lip synchronization compared to previous approaches, thanks to the complementary nature of our modeling. The pose of that generated talking faces can be controlled flexibly using another video as a pose source, which was a challenging task in prior methods. the model demonstrates excellent robustness even under extreme conditions, such as when dealing with large poses and various viewpoints. Extensive experiments validate the effectiveness of our approach.

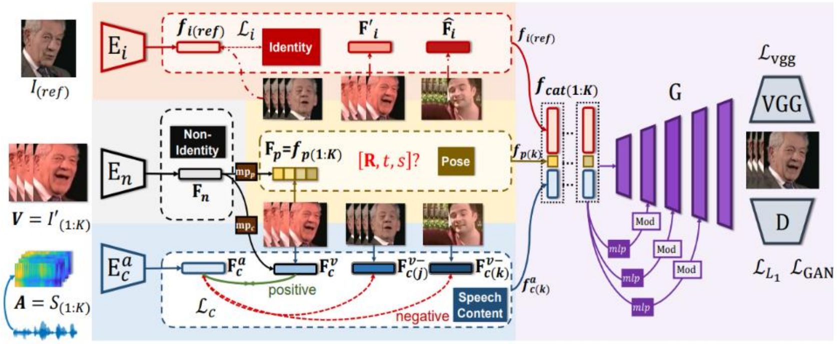


Figure 2.13: PC-AVS model[6]

In [7], the goal of Speech2Vid is to generate videos of a talking face using only an audio speech segment and a face image of the target identity (audio and image to video). They achieve this by using an encoder-decoder CNN model that uses a joint embedding of the face and audio to generate synthesized talking face video frames. The layer configurations are based on AlexNet and VGG-M, but filter sizes are adapted for the unusual input dimensions. The authors showed that re-dubbing videos from a different audio source (independent of the original speaker) is possible.

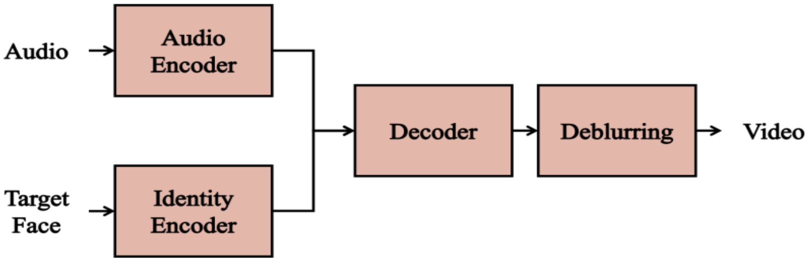


Figure 2.14: Architecture Of Speech2Vid[4]

In [8], The goal of Lip Gan to automatically translate a video of a person speaking in language A into a target language B with realistic lip synchronization. The model take a face frame and a speech from other source and generated a sequence of frames that contains the face speaking the audio with proper lip synchronization. The model is based on GAN network. The model contains two networks, a generator that generates faces by conditioning on audio inputs and a discriminator that tests whether the generated face and the input audio are in sync. The generator network is a modification of Speech2Vid model. The discriminator is the SyncNet Network.

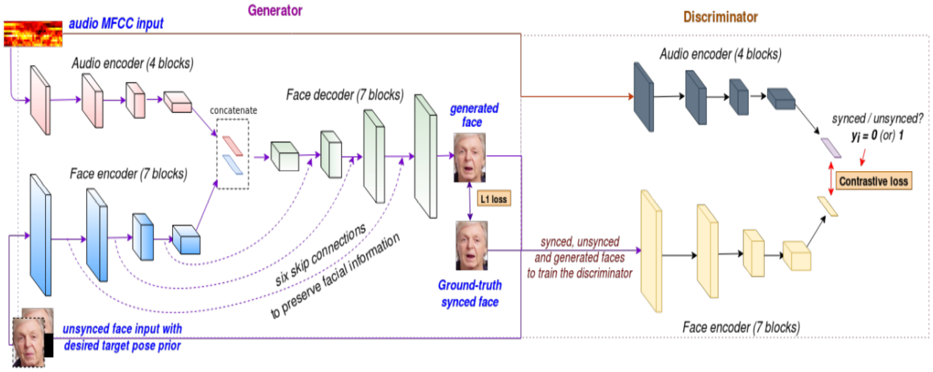
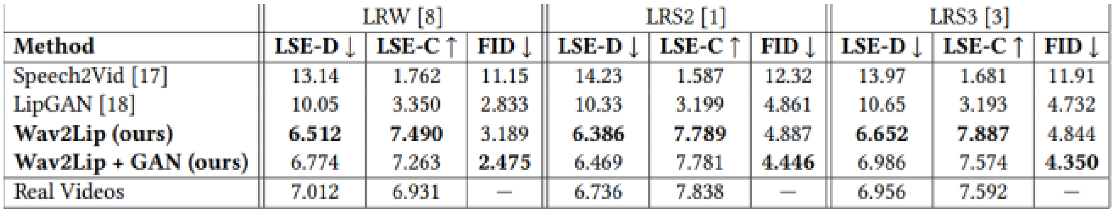


Figure 2.15:Detailed architecture[6]

In [9], the author improved on lip gan model [8] by changing the input in training due to technological advancements. The presentation of audio and video need to be matched or shown synchronously. Datasets used for train are LRS2 and for test are LRS2, LRW, and LRS3.The input for the model is video or an image with the corresponding audio and is processed by this model (The Single Shot Scale-invariant Face Detector (S3FD)) where the shapes and features are identified, mainly concentrating on the lower part of the face. The paper relied on **Resemble.AI** which provide APIS for generating voices, emotions, and themes API features consistent resource oriented URLs, delivers json encoded replies, and employs HTTP response codes, verification, and verbs. The paper did experiments using high resolution input videos and low resolution input videos for low-resolution video the processing time was much larger.

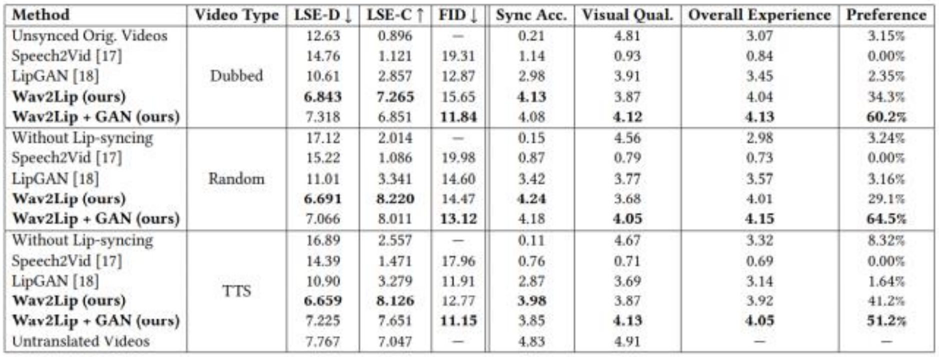
The author used a different loss function cosine-similarity with binary cross-entropy loss. That is, it computes a dot product between the ReLU activated video and speech embeddings v,s to yield a single value between [0, 1] for each sample that indicates the probability that the input audio-video pair is in sync. Training the expert lip-sync discriminator on the LRS2 train split (≈ 29 hours) with a batch size of 64, with Tv = 5 frames using the Adam optimizer with an initial learning rate of 1e −3 . The pre-trained expert lip-sync discriminator is about 91% accurate on the LRS2 test set, while the discriminator used in LipGAN is only 56% accurate on the same test set. The Attn wav2Lip algorithm is an adaptation of Wav2Lip model which is able to pay more attention on the lip region reconstruction. Proposed Wav2Lip model with two improvements on LipGAN with a buffer of T (T=5 in the original article) contiguous frames is utilized by the network to effectively make use of the temporal context information for lip-sync detection. Secondly, Wav2Lip employs a pre-trained lip-sync discriminator that can accurately detect sync in real videos to perform adversarial training of the generator. The reason for this is that lip region reconstruction loss corresponds to less than 4% of the total reconstruction loss (based on the spatial extent), while the discriminator of LipGAN [8] mainly focuses on the visual artifacts instead of the audio-lip correspondence. The output is five face frames matching with input Channel Attention. & Spatial Attention. & Attention Mechanism.

Table 1 : shows performance on new loss metrics[8]



	LRW [8]			LRS2 [1]			LRS3 [3]		
Method	**LSE-D ↓**	**LSE-C ↑**	**FID ↓**	**LSE-D ↓**	**LSE-C ↑**	**FID ↓**	**LSE-D ↓**	**LSE-C ↑**	**FID ↓**
Speech2Vid [17]	13.14	1.762	11.15	14.23	1.587	12.32	13.97	1.681	11.91
LipGAN [18]	10.05	3.350	2.833	10.33	3.199	4.861	10.65	3.193	4.732
Wav2Lip (ours)	**6.512**	**7.490**	3.189	**6.386**	**7.789**	4.887	**6.652**	**7.887**	4.844
Wav2Lip + GAN (ours)	6.774	7.263	**2.475**	6.469	7.781	**4.446**	6.986	7.574	**4.350**
Real Videos	7.012	6.931	–	6.736	7.838	–	6.956	7.592	–

Table 2 : shows the score of the model on real data[8]



Method	**Video Type**	**LSE-D ↓**	**LSE-C ↑**	**FID ↓**	**Sync Acc.**	**Visual Qual.**	**Overall Experience**	**Preference**
Unsynced Orig. Videos	Dubbed	12.63	0.896	–	0.21	4.81	3.07	3.15%
Speech2Vid [17]		14.76	1.121	19.31	1.14	0.93	0.84	0.00%
LipGAN [18]		10.61	2.857	12.87	2.98	3.91	3.45	2.35%
Wav2Lip (ours)		**6.843**	**7.265**	15.65	**4.13**	3.87	4.04	34.3%
Wav2Lip + GAN (ours)		7.318	6.851	**11.84**	4.08	**4.12**	**4.13**	**60.2%**
Without Lip-syncing	Random	17.12	2.014	–	0.15	4.56	2.98	3.24%
Speech2Vid [17]		15.22	1.086	19.98	0.87	0.79	0.73	0.00%
LipGAN [18]		11.01	3.341	14.60	3.42	3.77	3.57	3.16%
Wav2Lip (ours)		**6.691**	**8.220**	14.47	**4.24**	3.68	4.01	29.1%
Wav2Lip + GAN (ours)		7.066	8.011	**13.12**	4.18	**4.05**	**4.15**	**64.5%**
Without Lip-syncing	TTS	16.89	2.557	–	0.11	4.67	3.32	8.32%
Speech2Vid [17]		14.39	1.471	17.96	0.76	0.71	0.69	0.00%
LipGAN [18]		10.90	3.279	11.91	2.87	3.69	3.14	1.64%
Wav2Lip (ours)		**6.659**	**8.126**	12.77	**3.98**	3.87	3.92	41.2%
Wav2Lip + GAN (ours)		7.225	7.651	**11.15**	3.85	**4.13**	**4.05**	**51.2%**
Untranslated Videos		7.767	7.047	–	4.83	4.91	–	–

Difference between LipGan and Wav2lip:

It seems the major improvement in Wav2Lip over LipGAN is that of a better and more robust lip-sync Discriminator. We recall, that the discriminator that has the task of identifying if the generated lip movements match the audio input for a particular frame. In case of Wav2Lip, this is now extended to multiple frames so that there is a temporal context in the mix when deciding how accurate the generated lip movement is. Note that the discriminator in the LipGAN model architecture only has a 56% accuracy at detecting off-sync video-audio content, while the discriminator of Wav2Lip is 91% accurate at distinguishing in-sync content from off-sync content on the same test set.

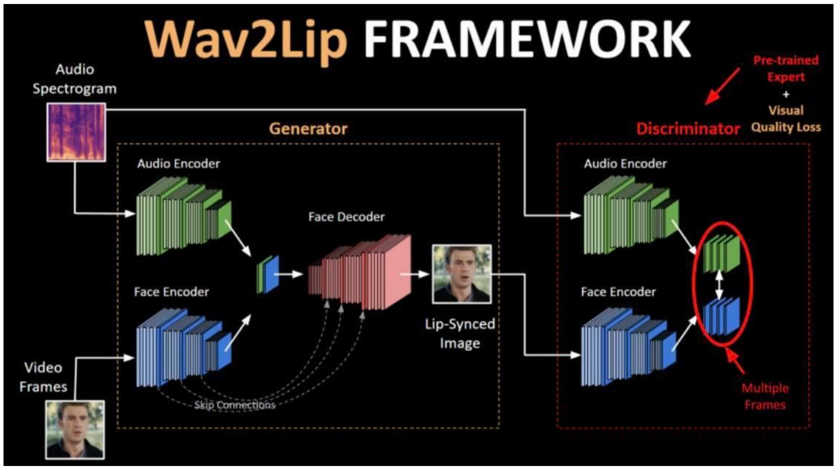


Figure 2.16 Wav2Lip Framework

Moreover, there is now a pre-trained lip-sync expert which is different from the regular sync net model is that it takes colored frames not grey frames discriminator whose weights are frozen during training of this model so that it does not get affected by the visual artifacts of the Generator and can focus solely on the correctness of the generated lip movements. There is an additional loss in the mix here.

A visual quality loss is added to ensure that the overall face in the output frame looks real, thereby minimizing artifacts that were noticed previously in LipGAN.

In table 2.3 summary of related work is provided.

Table 2.3 Summary Of The Related Work

Name	Input	Output	Methodology	Train / Test	LSE-D↓	LSE-C↑	SSIM↑	Year
A Lip Sync Expert Is All You Need for Speech to Lip Generation In The Wild	**Audio and video frames**	**synced video**	**wav2lip**	**LRS2 / LRS2, LRW ,and LRS3**	**6.512**	**7.49**	**0.862**	**2020**
Customising Video Messages using GANS	Audio and video frames	synced video	wav2lip+GAN	LRS2 / LRS2,LRW,and LRS3	6.774	7.263		2021
Towards audio to visual	Audio and video frames	synced video	LipGan	LRS2 / LRW	10.05	3.35		2020
You Said That?	Audio and video frames	synced video	Speech2video	voxCeleb - LRW / LRW	13.14	1.762		
Pose-Controllable Talking Face Generation	person's image with audio source	synced video	PC-AVR	VoxCeleb2 and LRW / LRW			0.861	2021
Talking Face Generation by Adversarially Disentangled	person's image and audio speech or a video	synced video	DAVS	LRW/LRW and some of Voxceleb			0.884	2019
Video-to-Video Translation for Visual Speech Synthesis	Video of a person talking and a label of that word	synced video	VispGan	LRW /LRW				2019

Chapter Three

Chapter 3: System Architecture

3.1 System Overview:

Our abstract system architecture is simply to take a video and a different audio to generate a new synced video as shown in figure 19 .

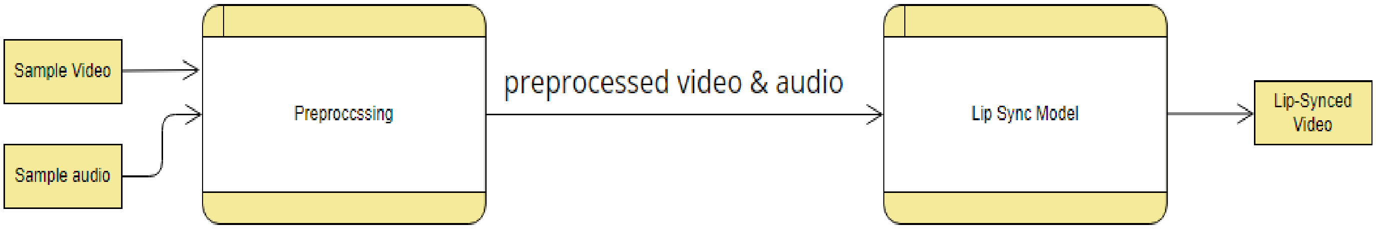


Figure 3.1: overview of architecture

3.1.1 Preprocessing

Videos are converted to frames using "ffmpeg" library, then face frame is extracted using S3FD model and the face region is cropped.

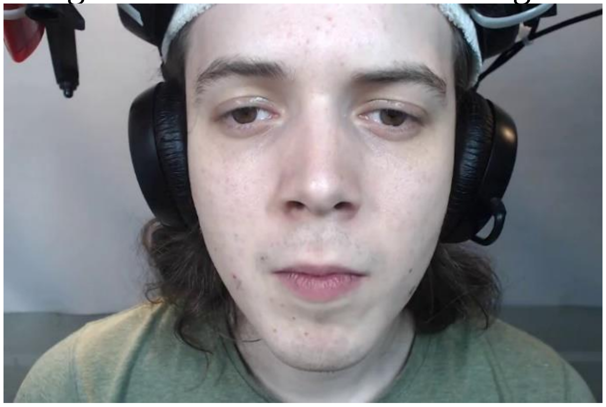


Figure 3.2 : Input Video Of The model

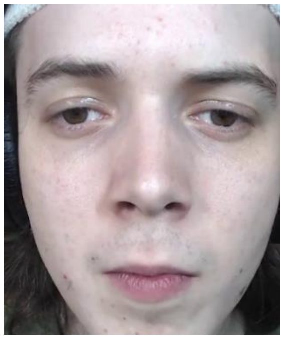


Figure 3.3: After Preprocessing

Figure 3.4 shows the steps of extracting audio from videos and calculating Mel-spectrogram using librosa library.

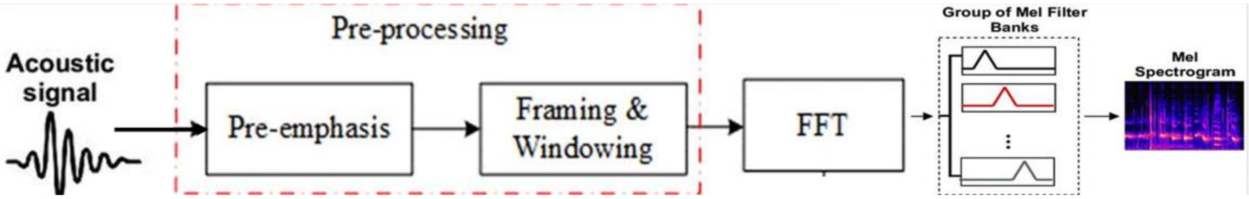


Figure 3.4. Audio preprocessing Steps

These steps are discussed in details as follows:
Pre-emphasis: The goal of pre-emphasis is to compensate the high-frequency part that was suppressed during the sound production mechanism of humans. Moreover, it can also amplify the importance of high-frequency formants. The next example demonstrates the effect of pre-emphasis.

Frame blocking: The input speech signal is segmented into frames of 20~30 ms with optional overlap of 1/3~1/2 of the frame size. Usually the frame size (in terms of sample points) is equal to power of two in order to facilitate the use of FFT. If this is not the case, we need to do zero padding to the nearest length of power of two.

Hamming windowing: Each frame has to be multiplied with a hamming window in order to keep the continuity of the first and the last points in the frame (to be detailed in the next step). If the signal in a frame is denoted by s(n), n = 0,…N-1, then the signal after Hamming windowing is s(n)*w(n), where w(n) is the Hamming window.

Fast Fourier Transform or FFT: Spectral analysis shows that different timbres in speech signals corresponds to different energy distribution over frequencies. Therefore, we usually perform FFT to obtain the magnitude frequency response of each frame. When we perform FFT on a frame, we assume that the signal within a frame is periodic, and continuous when wrapping around. If this is not the case, we can still perform FFT but the in continuity at the frame's first and last points is likely to introduce undesirable effects in the frequency response. To deal with this problem, we have two strategies: Multiply each frame by a Hamming window to increase its continuity at the first and last points.

Triangular Bandpass Filters: We multiple the magnitude frequency response by a set of 20 - 40 triangular bandpass filters to get the log energy of each triangular bandpass filter. The positions of these filters are equally spaced along the Mel frequency, which is related to the common linear frequency. Mel-frequency is proportional to the logarithm of the linear frequency.

3.1.2 Lip Sync Model

We chose **wav2lip model** as the main architecture as it has good accuracy among other models with the type of input, we want the system to handle.
The figure 31 shows the wav2lip model main components consisting of two encoders for video and audio then a decoder for the concatenated features with pre trained lip sync discriminator in addition, Wav2Lip uses an image quality

discriminator to improve the image quality of the generated videos. which will be discussed in the following paragraphs.

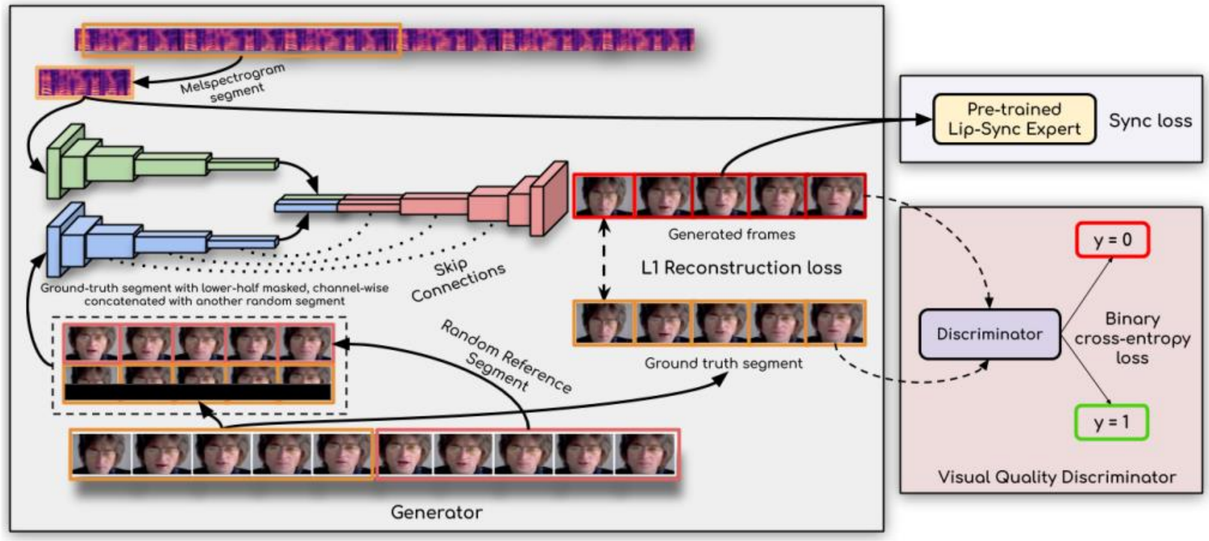

Melspectrogram segment
Pre-trained Lip-Sync Expert
Sync loss
Generated frames
L1 Reconstruction loss
Skip Connections
Ground-truth segment with lower-half masked, channel-wise concatenated with another random segment
Random Reference Segment
Ground truth segment
Discriminator
y = 0
y = 1
Binary cross-entropy loss
Visual Quality Discriminator
Generator

Figure 3.5: Wav2lip architecture[9]

Wav2Lip Generator:

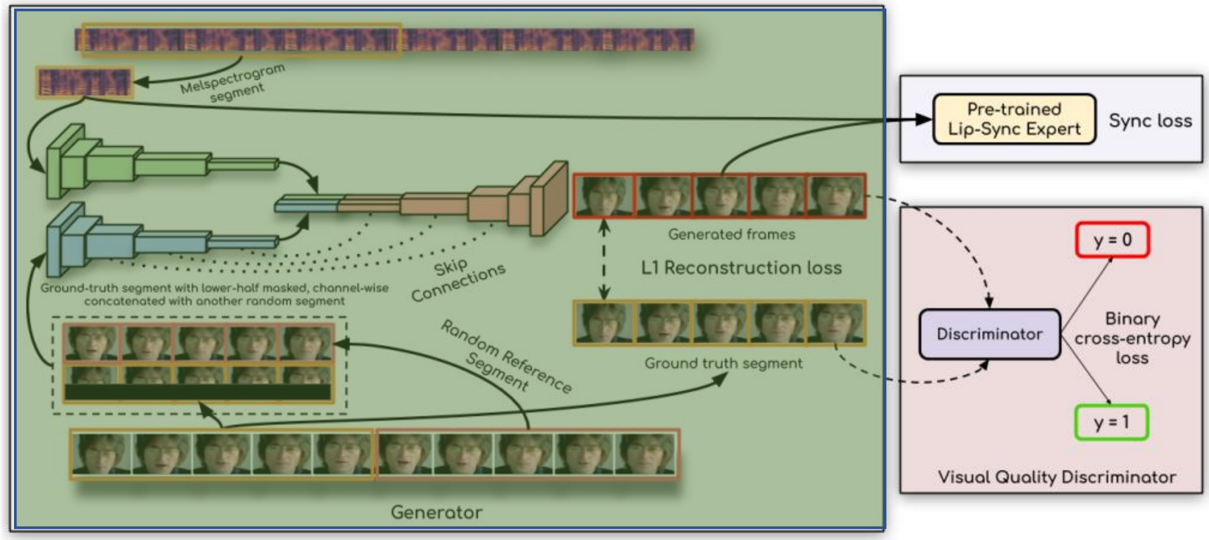

Melspectrogram segment
Pre-trained Lip-Sync Expert
Sync loss
Generated frames
L1 Reconstruction loss
Skip Connections
Ground-truth segment with lower-half masked, channel-wise concatenated with another random segment
Random Reference Segment
Ground truth segment
Discriminator
y = 0
y = 1
Binary cross-entropy loss
Visual Quality Discriminator
Generator

Figure 3.6 Generator Of Wav2Lip

Generator Architecture Details [9]. The author used a similar generator architecture as LipGAN [8]. The key improvement in wav2lip model lies in training this with the expert discriminator. The generator G contains three blocks: (i) Identity Encoder, (ii) Speech Encoder, and a (iii) Face Decoder. The Identity Encoder is a stack of residual convolutional layers that encode a random reference frame R, concatenated with a pose-prior P (target-face with lower-half masked) along the channel axis. The Speech Encoder is also a stack of 2D1convolutions to encode the input speech segment S which is then concatenated with the face representation. The decoder is also a stack of convolutional layers, along with transpose convolutions for up sampling. The generator is trained to minimize L1 reconstruction loss between the generated frames L_g and ground-truth frames LG.

$$L_{\text{recon}} = \frac{1}{N}\sum_{i=1}^{N}||L_g - L_G||_1$$

eq(3)

Thus, the generator is similar to the previous works, a 2D-CNN encoder-decoder network that generates each frame independently. The pre-trained expert lip-sync discriminator that needs a temporal window of Tv = 5 frames as input

Discriminator (Sync Net):

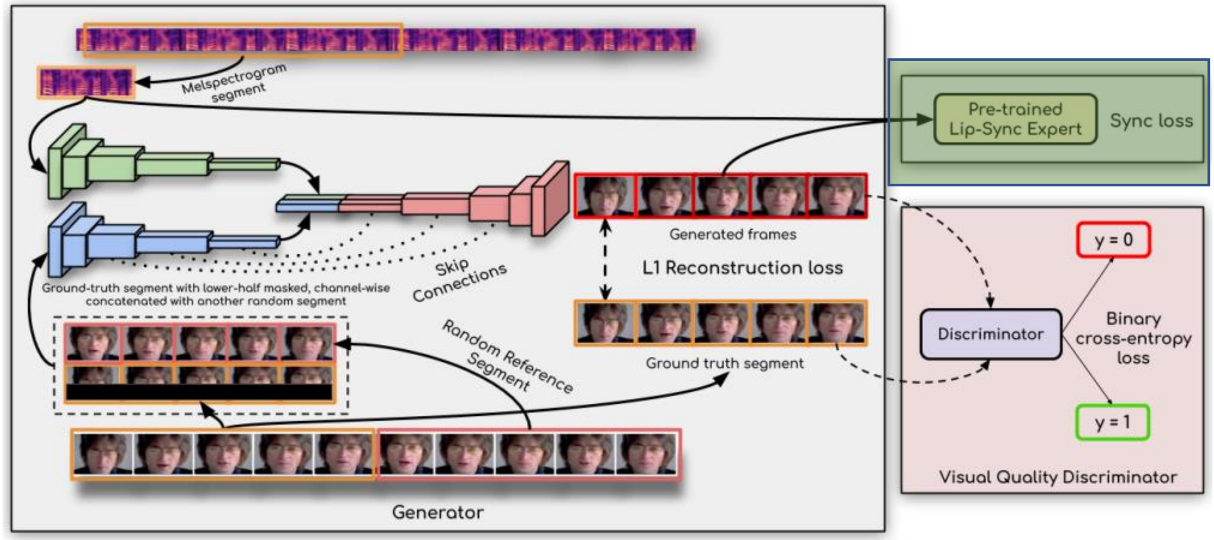

Melspectrogram segment
Pre-trained Lip-Sync Expert
Sync loss
Generated frames
L1 Reconstruction loss
Skip Connections
Ground-truth segment with lower-half masked, channel-wise concatenated with another random segment
Random Reference Segment
Ground truth segment
y = 0
Discriminator
Binary cross-entropy loss
y = 1
Visual Quality Discriminator
Generator

Figure 3.6 Lip-Sync Expert Discriminator

Experts lip-sync discriminator. The following changes were made to SyncNet [3] to train an expert lip-sync discriminator that suits our lip generation task. Firstly, instead of feeding gray scale images concatenated channel-wise as in the original model, colored images are fed to sync net model. Secondly, wav2lip is significantly deeper, with residual skip connections. Thirdly, inspired by this public implementation. The authors in [9] used a different loss function: cosine-similarity with binary cross-entropy loss. That is, a dot product is computed between the ReLU-activated video and speech embeddings v,s to yield a single value between [0, 1] for each sample that indicates the probability that the input audio-video pair is in sync:

$$P_{\text{sync}} = \frac{v \cdot s}{max(\|v\|_2 \cdot \|s\|_2, \epsilon)}$$

eq(4)

Penalizing Inaccurate Lip Generation. During training, as the expert discriminator trained, processes Tv = 5 contiguous frames at a time, the generator G to generate all the Tv = 5 frames. The author samples a random contiguous window for the reference frames, to ensure as much temporal consistency of pose, etc. across the Tv window. As the generator processes each frame independently.

the batch dimension while feeding the reference frames to get an input shape of (N · Tv ,H,W , 3), where N, H, W are the batch-size, height, and width respectively. While feeding the generated frames to the expert discriminator, the time-steps are concatenated along the channel-dimension as was also done during the training of the discriminator. The resulting input shape to the expert discriminator is (N,H/2,W , 3·Tv), where only the lower half of the generated face is used for discrimination. The generator is also trained to minimize the "expert sync-loss" Esync from the expert discriminator [3]

$$E_{\text{sync}} = \frac{1}{N}\sum_{i=1}^{N} -\log(P^{i}_{\text{sync}})$$

eq(5)

where P sync is calculated according to Equation 5. Note that the expert discriminator's weights remain frozen during the training of the generator. This strong discrimination based purely on the lip-sync concept learned from real videos forces the generator to also achieve realistic lip-sync to minimize the lip-sync loss Esync.

Another Discriminator For Quality:

In the conducted experiments in wav2lip paper, it was observed that using a strong lip-sync discriminator forces the generator to produce accurate lip shapes. However, it sometimes results in the morphed regions to be slightly blurry or contain slight artifacts. To mitigate this minor loss in quality, the author trained a simple visual quality discriminator in a GAN setup along with the generator. Thus, the model has two discriminators, one for sync accuracy and another for better visual quality. The lip-sync discriminator is not trained in a GAN setup .On the other hand, since the visual quality discriminator does not perform any checks on lip-sync and only penalizes unrealistic face generations, it is trained on the generated faces. The discriminator D consists of a stack of convolutional blocks. Each block consists of a convolutional layer followed by a Leaky ReLU activation. The discriminator is trained to maximize the objective function Ldisc shown in **figure 3.7**

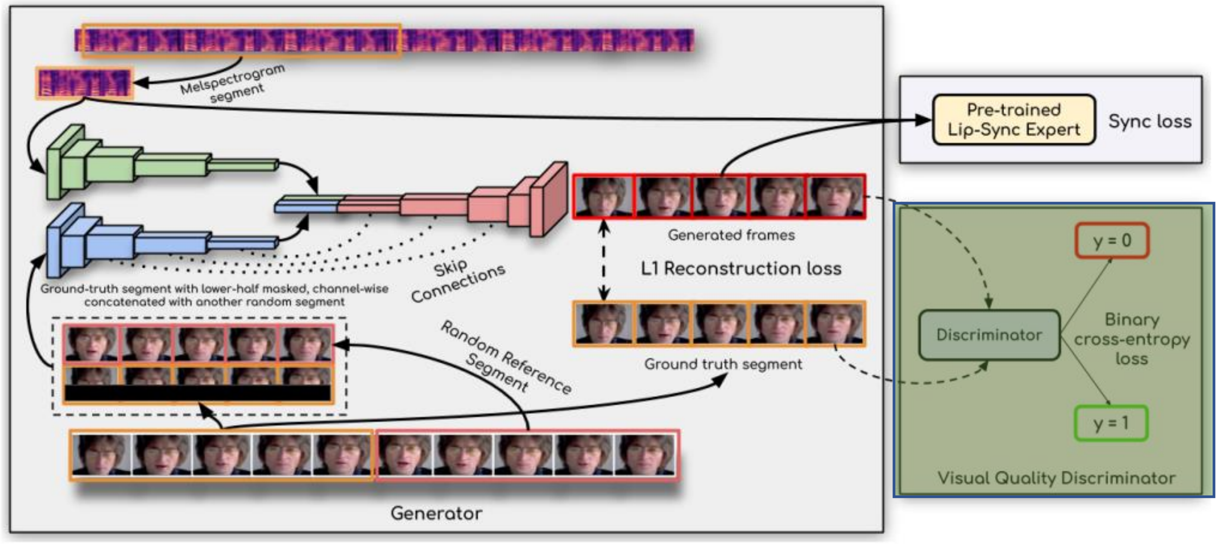

Melspectrogram segment
Pre-trained Lip-Sync Expert
Sync loss
Generated frames
L1 Reconstruction loss
Skip Connections
Ground-truth segment with lower-half masked, channel-wise concatenated with another random segment
Random Reference Segment
Ground truth segment
Discriminator
y = 0
Binary cross-entropy loss
y = 1
Visual Quality Discriminator
Generator

Figure 3.7 Discriminator Of Visual Quality

The model generates a talking face video frame-by-frame. The visual input at each time-step is the current face crop (from the source frame), concatenated with the same current face crop with lower-half masked to be used as a pose prior. Thus, during inference, the model does not need to change the pose, significantly reducing artifacts. The corresponding audio segment is also given as input to the speech sub-network, and the network generates the input face crop, but with the mouth region morphed. All our code and models will be released publicly.

3.2 Additional Features

Our system consists of sub modules to create a full translation system that handles different inputs to make the system more useful and offers more features to the users.

3.2.1 Text to Video

Taking Text and image or video as input ,we start from generating speech from the text input Microsoft API ,then taking the generated speech with the image / video to lip sync model to generate the new synced video.

3.2.2 Video To Video Translation

We take input video with a source language, then extract the audio from the video ,then extract the text from audio using ASR ,then translating the text to the target language , then generating the audio using TTS model ,Finally the lip sync model takes the new audio with the source video as show in **figure 3.8**

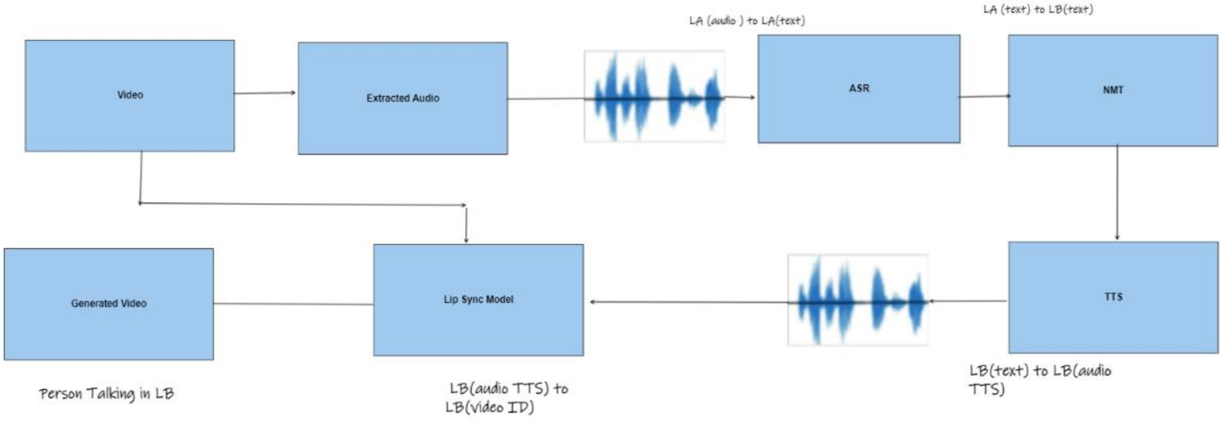


Figure 3.8: Translation Pipline

ASR:

Is Automatic speech recognition which is used to extract a transcript from any audio file from video, maybe the same video but to use transcript in automatic translation of the video in language A to a video in language B.

NMT (Natural machine translation):

This module helps to take the transcript from previous step and translate the language to the other language.

TTS (Text to speech):

This module helps generate a human like voice using most well-known apis and deep learning models.

Gender And Age Detection :

Detect whether the person in the input video is male or female and give information about the approximate age of him to generate the audio based on the gender in the input video to use the TTS module to generate female or male voice.

Lip Sync Model:

This is the **main** module that uses wav2lip to generate the new video based on any arbitrary

Chapter Four

Chapter 4: Implementation and Testing

To implement the target model that sync any talking face video with any arbitrary audio.

We fine-tuned the Wav2Lip model on two datasets, also we changed the loss function and trained two models for gender and age detection. The datasets and experiments will be discussed in the following sections.

4.1 Datasets

Wav2lip model was trained on LRS2 dataset as it’s diverse in the vocabulary and the identities.

The dataset consists of thousands of spoken sentences from BBC television. All videos are approximately 25 frames.

Table 4.4 LRS2 Dataset

Set	# utterances	Vocab
Pre-train	**96,318**	**41,427**
Train	**45,839**	**17,660**
Validation	**1,082**	**1,984**
Test	**1,243**	**1,698**

We used Lombard grid corpus shown in table 4.3, grid corpus shown in table 4.4 that showing the number of speakers and videos in the datasets.

Lombard grid corpus dataset

Table 4.5 Lombard Grid Corpus Dataset

	Train
# of speakers	54
# of videos	5400
# utterances per talker	100
Frame Rate	25

Grid Corpus:

Table 4.6 Grid Corpus Dataset

	Train
# of speakers	34
# of videos	34000
# utterances per talker	1000

We used these two datasets as they have better quality videos.

We used vidtimit dataset for evaluation of the wav2lip model :

Table 4.7 Evaluation Dataset

	Evaluation
# of speakers	43
# of videos	430
# utterances per talker	10

Dataset for gender detection : CelebA is a large-scale face with more than 200k celebrities and 10,177 number of identities

Table 5.8 CelebA Dataset

Train	**Validation**	**Test**
20000	**5000**	**1000**

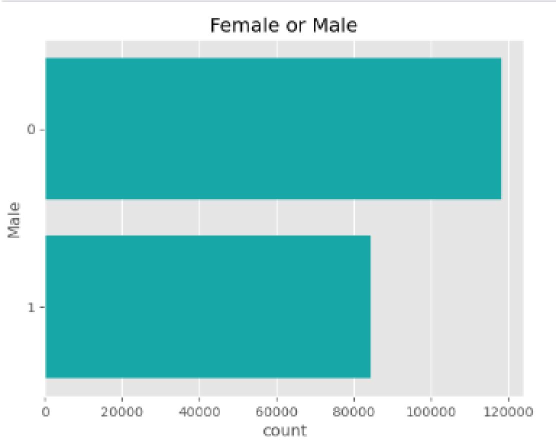


Figure 4.0.1 Distribution Of CelebA

4.2 Evaluation Metrics

LSE-D (“Lip Sync Error - Distance"). A lower LSE-D denotes a higher audio-visual match, i.e., the speech and lip movements are in sync.

LSE-C (Lip Sync Error - Confidence). Higher the confidence, the better the audio-video correlation. A lower confidence score denotes that there are several portions of the video with completely out-of-sync lip movements.

4.3 Experiments and Results

We will discuss the evaluation of the pre-trained wav2lip , how we fine-tuned it on other two datasets ,gender and age detection and video to video translation pipline.

4.3.1 Wav2Lip Model Evaluation

We test the pretrained wav2lip model on a small dataset called (**vidmit)** which contains some videos of people with one color background and has about 30 videos and we evaluate the model by **two metrics (LSE-loss, LSE – confidence)**

We evaluated the model by taking videos from the dataset and calculating the two metrics (LSE-loss, LSE-confidence) and generating the same videos again with the same audio by wav2lip model to compare the results.

LSE-loss: Is calculated based on the minimum distance between the audio and video features every five frames using the sync net model to calculate the contrastive loss between them.

LSE – confidence: Is calculated based on the average distance between the audio and video features every five frames for all segments of the video to get a confidence that most of the video is in sync.

Preprocessing steps for generated fake videos using wav2lip model by extracting cropped frames that contains face only, then extracted Mel spectrogram features from the audio, finally input that to the encoder and decoder to do inference and generate the videos.

Example input:

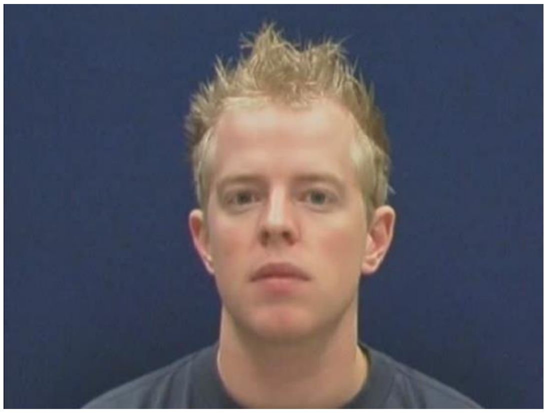


Figure 4.2:video sample from vidtimit

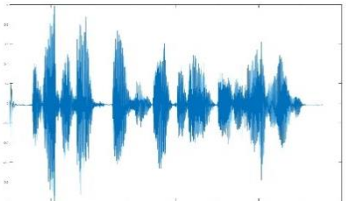


Figure 4.3: audio sample

Example output after preprocessing:

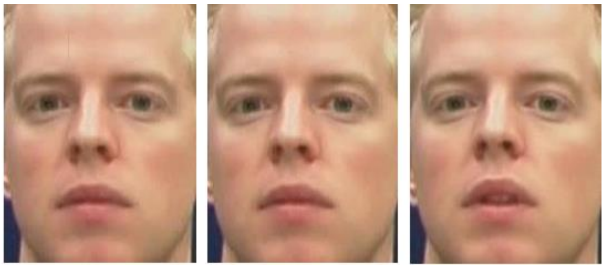


Figure 4.4 Detected Face Frames (Vidtimit)

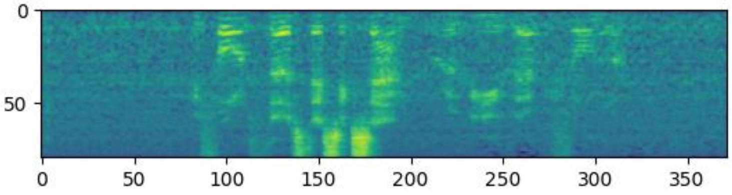


Figure 4.5 MelSpectrogram Extracted from audio.

4.3.2 Wav2Lip Model Enhancement

In order to enhance the model, we applied preprocessing for the Lombard grid corpus dataset which is about 5000 short videos for different 10 persons saying different sentences in each video , we used this data as it has better quality videos than the LRS2 to fine tune the wav2lip model on this new data to get better inference result by finetuning hyper parameters for the new dataset.

Cropped Frame using s3fd model along with the mel spectrogram features from audio:

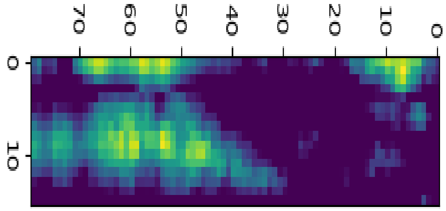


Figure 4.6: mel spectrogram of lombard grid

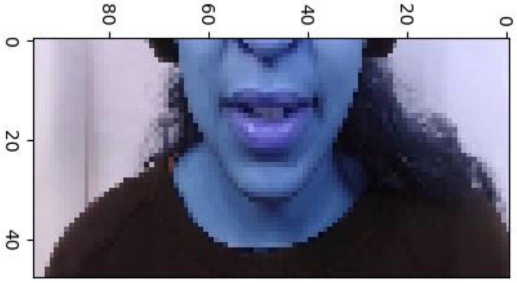


Figure 4.7: cropped frame contains the lip.

After training for 50 epochs:

Table 9 Training Result on lombard dataset

Metrics	L1	Sync	Perception	Fake	Real
Score	0.52638750	0.14395864	10.0308957	0.00966318	0.009603

Example inference of the model after fine tuning is shown in **figure 4.9.**

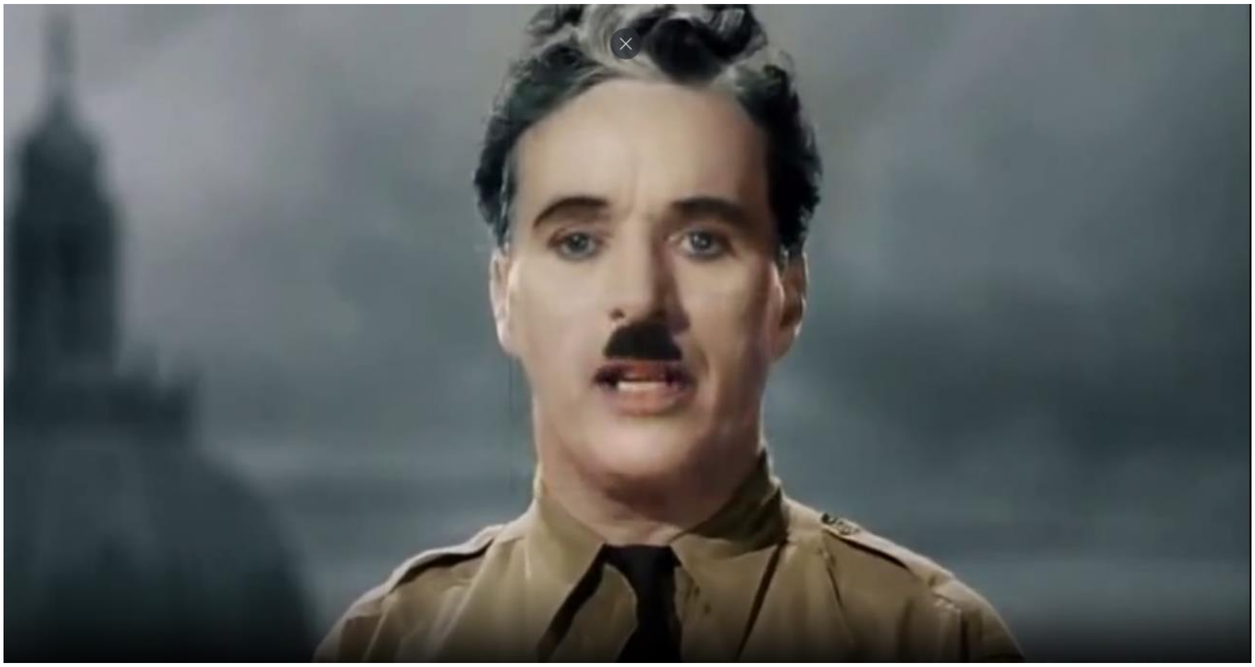


Figure 4.8: example output after finetuning

The quality of the video around the mouth and teeth of the person looks better but the syncing state of the audio is more out of sync, so it is compromise between getting better quality and getting better syncing of the new audio.

4.3.3 Gender and Age Detection

Trained two deep learning models one for detecting gender from a frame in video and the other model to detect the age to use it in producing the sound (producing audio based on age)

Age detection model:

We trained the model on dataset called CelebA. Chosen Model for this task is (Inception model v3) as it has good accuracy in many tasks.

Hyper parameters and activation function used for this task:
Learning rate: 0.001
Batch size: 32
Number of Epochs: 20
Activation function: sigmoid

The loss after we trained the model:

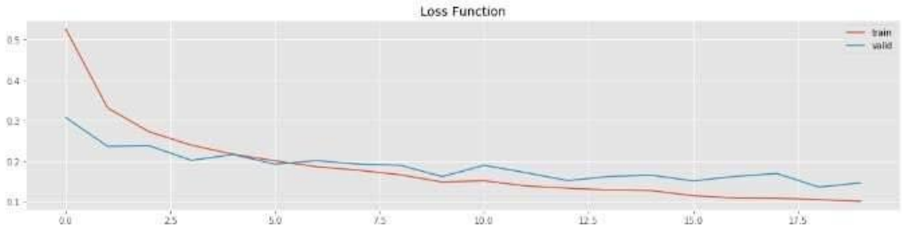


Figure 4.9: Loss Of Gender Detection

The accuracy:

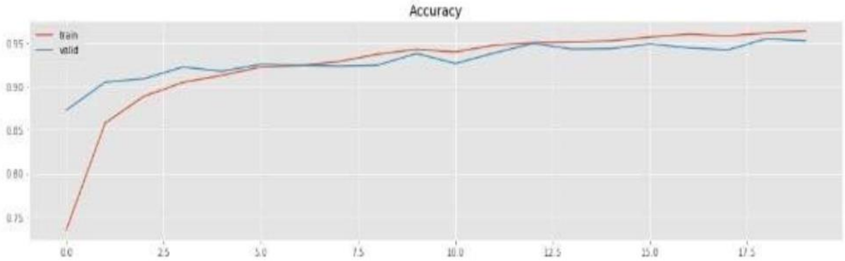


Figure 4.10: Accuracy of Gender Detection

Age detection model:

We trained vgg16 to predict the age as it's regression problem on the dataset **UTK faces.**

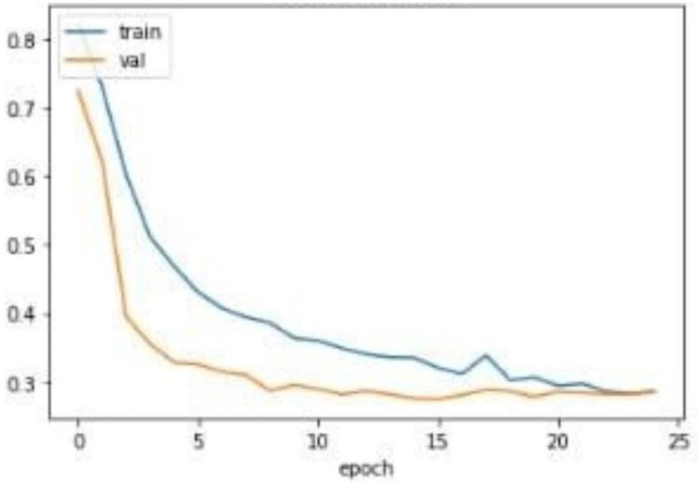
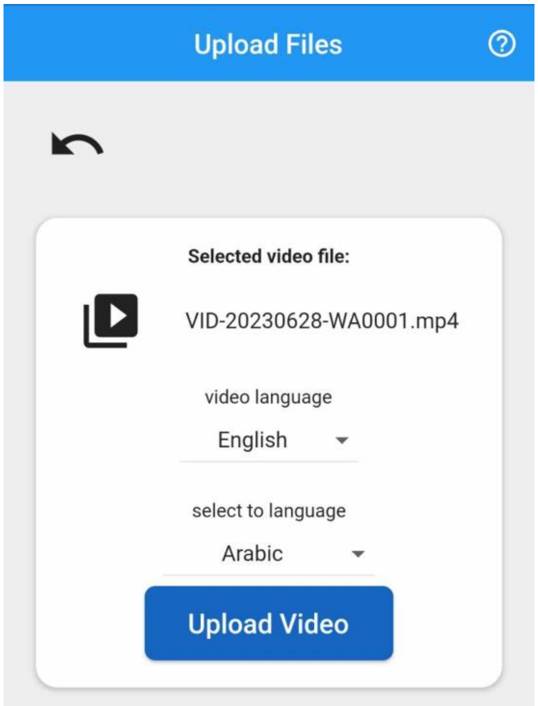


Figure 4.11: loss of age Detection

4.3.4 Video To Video Translation



Selected video file:

VID-20230628-WA0001.mp4

video language

English

select to language

Arabic

Figure 4.12 :Input of the application

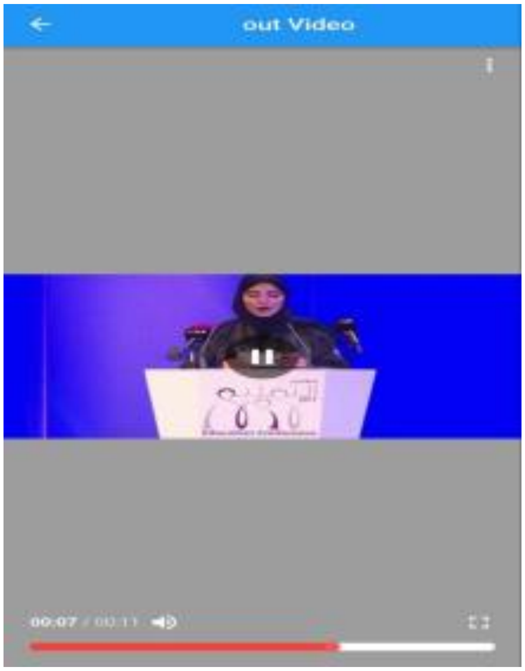


Figure4.13 : The Output Of Application

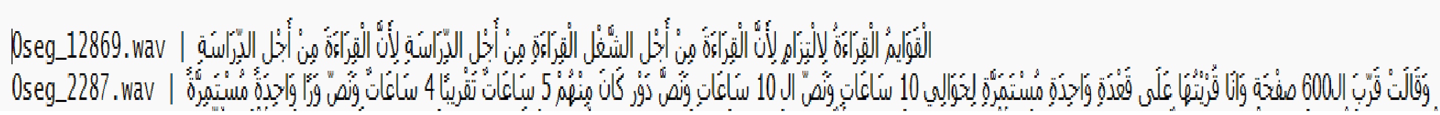
The user can translate from English to the target language (Arabic or German)
Speech to text:
Used Wav2vec model to extract text from English videos to translate to transform the text to another language it later.

Text to text:
We used Google API to translate the language from the input video to a new language.

Text to speech module:
Generating audio based on the gender in the video to make more realistic video.
We Used **salorie** models which contains many models with different speakers in different ages to have options in producing the sound and supports many languages except Arabic.
Used Microsoft azure api which has options to use a dialect in Arabic language to produce high quality audio to the wav2lip.

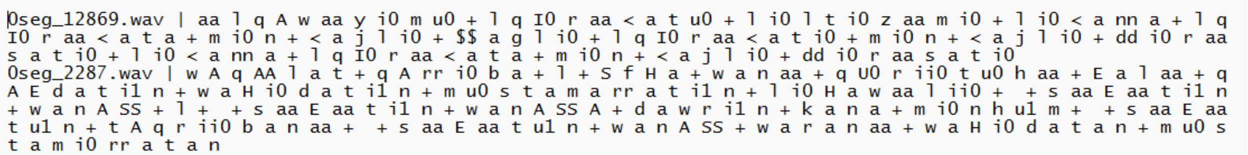
4.3.5 Voice Cloning Experiments

We fine-tuned Tacotron2 text to speech model to produce another voice of a male in Arabic by preprocessing the Qasr dataset which transcribes to a format called Buckwalter to be able to represent the Arabic words as English letters , then feeding the audio along with Buckwalter format to finetune the model .



```
0seg_12869.wav | الْقَوَايِمُ الْقِرَاءَةُ لِالْتِزَامِ لِأَنَّ الْقِرَاءَةَ مِنْ أَجْلِ الشُّغْلِ الْقِرَاءَةِ مِنْ أَجْلِ الدِّرَاسَةِ لِأَنَّ الْقِرَاءَةَ مِنْ أَجْلِ الدِّرَاسَةِ
0seg_2287.wav | وَقَالَتْ قَرِّبْ ال600 صفْحَة وَأنَا قَرَيْتُهَا عَلَى قَعْدَةٍ وَاحِدَةٍ مُسْتَمِرَّةٍ لِحَوَالِي 10 سَاعَاتٍ وَنِصّ ال 10 سَاعَاتٍ وَنِصّ دَوْر كَانَ مِنْهُمْ 5 سَاعَاتٌ تَقْرِيبًا 4 سَاعَاتٌ وَنِصّ وَرَّا وَاحِدَةٌ مُسْتَمِرَّةٌ
```

Figure4.12 : Arabic Text



```
0seg_12869.wav | aa l q A w aa y i0 m u0 + l q I0 r aa < a t u0 + l i0 l t i0 z aa m i0 + l i0 < a nn a + l q
I0 r aa < a t a + m i0 n + < a j l i0 + $$ a g l i0 + l q I0 r aa < a t i0 + m i0 n + < a j l i0 + dd i0 r aa
s a t i0 + l i0 < a nn a + l q I0 r aa < a t a + m i0 n + < a j l i0 + dd i0 r aa s a t i0
0seg_2287.wav | w A q AA l a t + q A rr i0 b a + l + S f H a + w a n aa + q U0 r ii0 t u0 h aa + E a l aa + q
A E d a t i1 n + w a H i0 d a t i1 n + m u0 s t a m a rr a t i1 n + l i0 H a w aa l ii0 +  + s aa E aa t i1 n
+ w a n A SS + l +  + s aa E aa t i1 n + w a n A SS A + d a w r i1 n + k a n a + m i0 n h u1 m +  + s aa E aa
t u1 n + t A q r ii0 b a n aa +  + s aa E aa t u1 n + w a n A SS + w a r a n aa + w a H i0 d a t a n + m u0 s
t a m i0 rr a t a n
```

Figure 4.13 : Buckwalter Format

The results is that the model generate audio in the voice of the new speaker but with some noise.

4.4 Software Tools

Pytorch: Is a library for creating deep learning models and scientific computing ,we used it in Model training and creation of the gender and age detection models

Librosa: Audio processing library contains algorithms for short fourier transform and used to extract Mel spectrogram features from audio

Ffmpeg: video processing library that can handle trimming certain parts and combine generated frames to generate the new video.

Opencv:Image processing library used for face detection and applying filters to the frames of the video.

Chapter Five

Chapter 5: User Interface

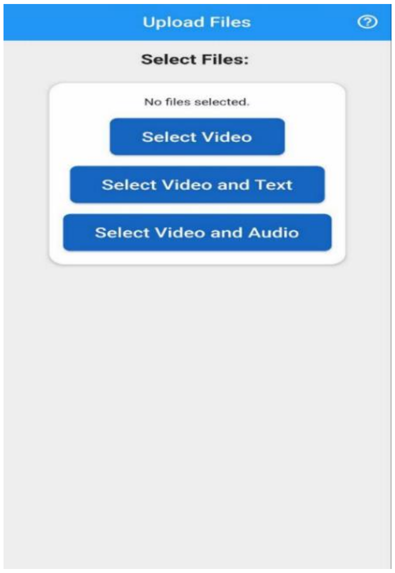
Our application provides several options for the user including taking translation the input automatically by determining the target language (Arabic or German) ,taking the recorded audio from the user along with the input video or generating the audio automatically using input text from user.

 The steps to use each of these options will be explained as follows

1 – Open the application

Figure 5.1 Logo Of The App

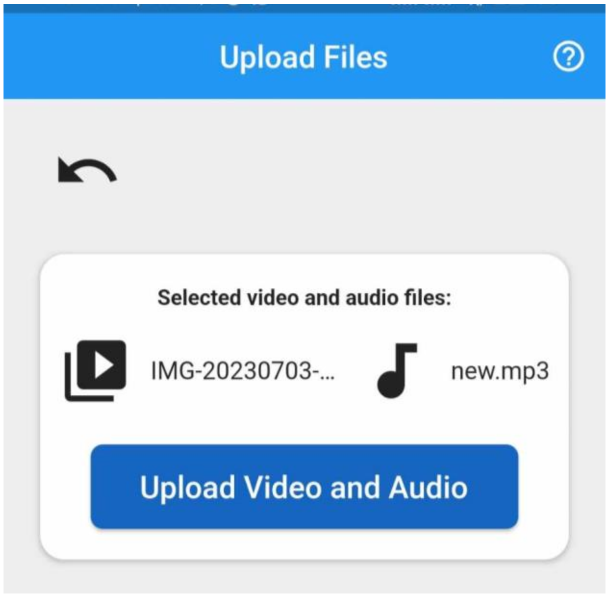
Option 1: Choosing video and audio :
The user selects a video and an audio file from the user device to generate the new video based on the input audio.



No files selected.

Figure 5.2 Interface Of The App

3 - After choosing video and audio, the user press upload video and audio



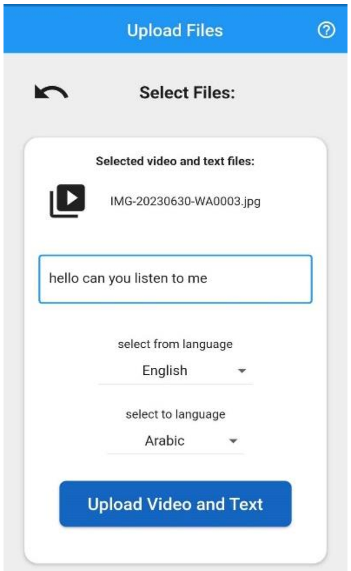
Selected video and audio files:

IMG-20230703-... new.mp3

Figure 5.3 After Uploading

Option 2 : Choosing text and video or image

The user input the text in the text box and selecting the language to translate to it.



Selected video and text files:

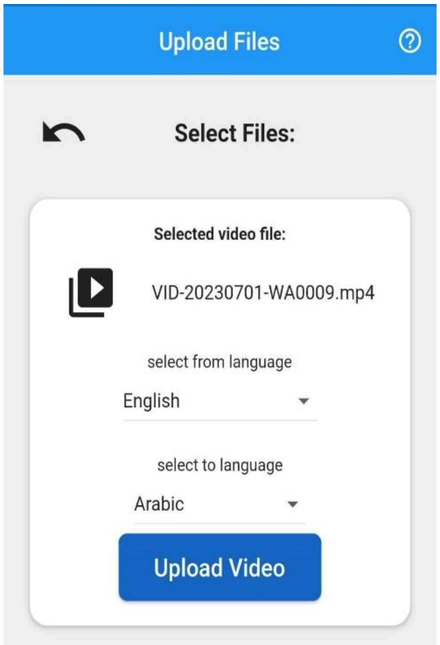
IMG-20230630-WA0003.jpg

hello can you listen to me

select from language: English

select to language: Arabic

Figure 5.4 After Entering Input Text



Selected video file:

VID-20230701-WA0009.mp4

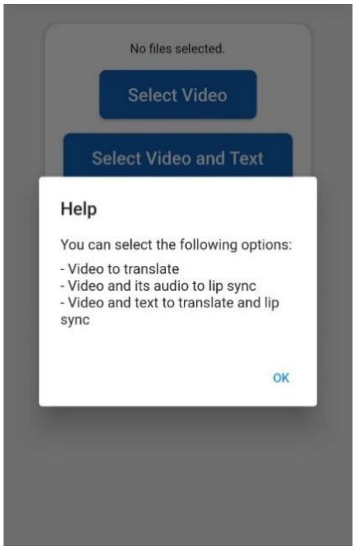
select from language

English

select to language

Arabic

Figure 5.5 : After Choosing The Language



No files selected.

Help

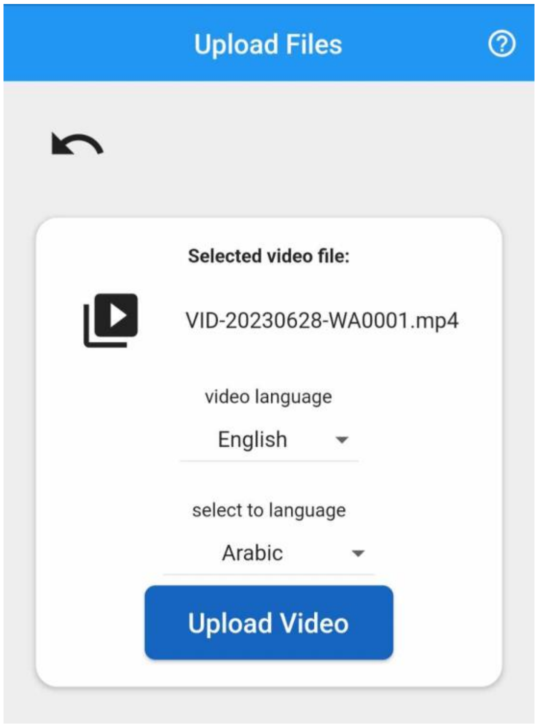
You can select the following options:

- Video to translate
- Video and its audio to lip sync
- Video and text to translate and lip sync

Figure 5.6 Help Instructions

Option 3: Video to Video Translation

The user can choose to translate from English to Arabic or German



Selected video file:

VID-20230628-WA0001.mp4

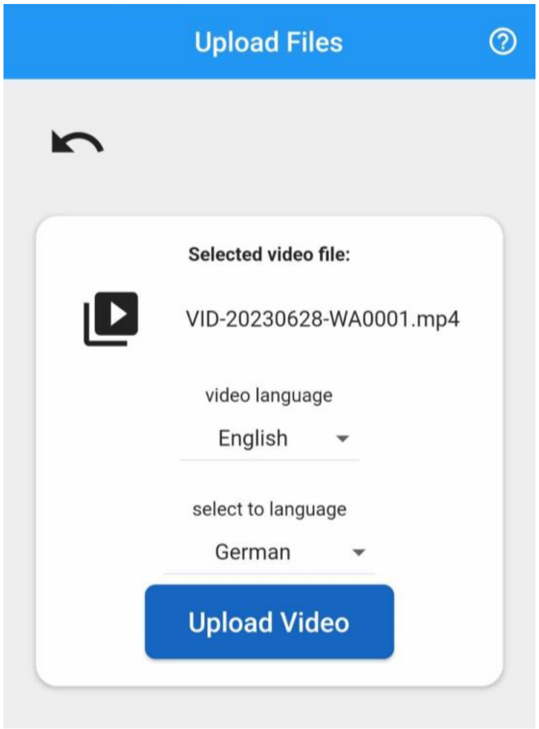
video language

English

select to language

Arabic

Figure 5.7 English To Arabic



Selected video file:

VID-20230628-WA0001.mp4

video language

English

select to language

German

Figure 5.8 From English To German

The output of all options will be Synced video

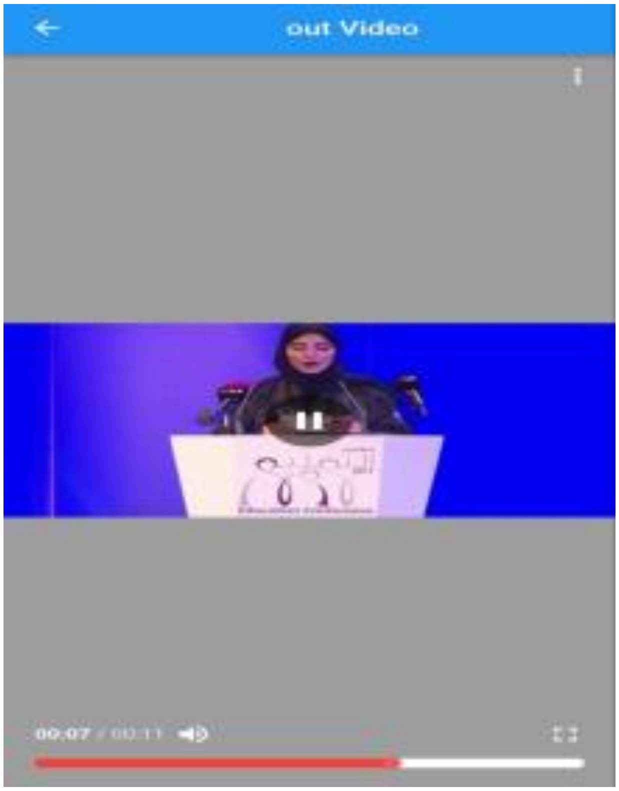


Figure 5.9 Output

Chapter 6

Chapter 6: Conclusion and Future Work

6.1 Conclusion

In this work we utilized the Wav2Lip model that can generate accurate lip-synced videos in the wild. The authors of Wav2Lip [9] highlighted two major reasons why current approaches are inaccurate while lip-syncing unconstrained talking face videos. Based on this, we investigated that a pretrained, accurate lip-sync “expert" can actually enforce accurate, natural lip motion generation. Before evaluating the model, we used the new evaluation benchmarks and metrics proposed in wav2lip paper. The Wav2Lip model [9] outperforms the current approaches by a large margin in both quantitative metrics and human evaluations and we enhanced the model by fine tuning it on another two datasets.

We extend the problem of automatic machine translation to face to face translation with a focus on audio-visual content, i.e., where input and output are talking face videos. Beyond demonstrating the feasibility of a Face-to-Face translation pipeline. We also contribute towards several language processing tasks (such as textual machine translation).

6.2 Future Work

We believe our efforts and ideas in this problem can lead to new directions such as

- Making the model run in real time will be important for video calls, live conferences, and online meetings.
- Voice Cloning for Multilingual Videos: Extending the model to generate videos in a new language while retaining the speech from the source language (voice cloning) would be an interesting direction. This could enable the creation of multilingual videos with.
- Summarizing large videos with a small of version of it by summarizing the transcript and generate new summarized version.
- Enhancing Realism with Facial Dynamics: Exploring techniques to incorporate facial dynamics, such as subtle muscle movements and micro expressions, can further enhance the realism of lip-synced videos.

REFERENCES

[1] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd. USA: Prentice Hall Press, 2009, isbn: 0136042597.

[2] Liljegren, Johan, and Pontus Nordqvist. "Generative Adversarial Networks in Lip-Synchronized Deepfakes for Personalized Video Messages." *Master's Theses in Mathematical Sciences* (2021).

[3]Joon Son Chung and Andrew Zisserman. 2016. Out of time: automated lip sync in the wild. In Workshop on Multi-view Lip-reading, ACCV.

[4] Doukas, Michail C., Viktoriia Sharmanska, and Stefanos Zafeiriou. "Video-to-Video Translation for Visual Speech Synthesis." arXiv preprint arXiv:1905.12043 (2019).

[5] Wang, Ganglai, et al. "Attention-Based Lip Audio-Visual Synthesis for Talking Face Generation in the Wild." arXiv preprint arXiv:2203.03984 (2022).

[6] Zhou, Hang, et al. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021. p. 4176-4186.

[7] Joon Son Chung, Amir Jamaludin, and Andrew Zisserman. 2017. You said that? arXiv preprint arXiv:1705.02966 (2017).

[8] KR, P., Mukhopadhyay, R., Philip, J., Jha, A., Namboodiri, V., & Jawahar, C. V. (2019, October). Towards automatic face-to-face translation. In Proceedings of the 27th ACM international conference on multimedia (pp. 1428-1436).

[9] Prajwal, K. R., Mukhopadhyay, R., Namboodiri, V. P., & Jawahar, C. V. (2020, October). A lip sync expert is all you need for speech to lip generation in the wild. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 484-492).

[10] Mira, Rodrigo, et al. "End-to-end video-to-speech synthesis using generative adversarial networks." IEEE Transactions on Cybernetics (2022).

[11]Zhang, Shifeng, et al. "S3fd: Single shot scale-invariant face detector." *Proceedings of the IEEE international conference on computer vision*. 2017.

[12] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. 2017. Synthesizing obama: learning lip sync from audio. ACM Transactions on Graphics (TOG) 36, 4 (2017), 95.

[13] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. 2019. Neural Voice Puppetry: Audio-driven Facial Reenactment. arXiv preprint arXiv:1912.05566 (2019).

[14] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. 2020. DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection. arXiv:2001.00179 [cs.CV] .

[15] Eleanor Tursman, Marilyn George, Seny Kamara, and James Tompkin. 2020. Towards Untrusted Social Video Verification to Combat Deepfakes via Face Geometry Consistency. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

[16] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. 2019. Realistic speech-driven facial animation with gans. International Journal of Computer Vision (2019), 1–16.

[17] Rithesh Kumar, Jose Sotelo, Kundan Kumar, Alexandre de Brébisson, and Yoshua Bengio. 2017. Obamanet: Photo-realistic lip-sync from text. arXiv preprint arXiv:1801.01442 (2017).

[18] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic mod(els. In Proc. icml, Vol. 30.3.

[19] NPD. 2016. 52 Percent of Millennial Smartphone Owners Use their Device for Video Calling, According to The NPD Group. https://www.npd.com/wps/portal/npd/us/ news/press-releases/2016/52-percent-of-millennial-smartphone-owners-usetheir-device-for-video-calling-according-to-the-npd-group/