Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Customising Video Messages using GANS

by

Darshan Umapathi

Supervisor Prof. Gerard Lacey

August 30, 2021

A dissertation submitted in partial fulfilment

of the requirements for the degree of

M.Sc (Computer Science)

Declaration

I hereby declare that this dissertation is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

Signed: _____ Date: _____

Abstract

With an exponential rise in video messages on the internet, creating videos and customizing them quickly has become an essential need. Short videos are growing, and they tend to be engaging, for example, reels on Instagram, Facebook, TikTok etc. However, customizing pre-recorded video is challenging, and modifications require either re-shooting the video with new phrases or using complex editing systems such as Adobe Photoshop or ProCreate to achieve desired results. These methods are expensive and time-consuming. This study proposes a novel ensemble model which helps to automatically translate text input to video to generate realistic lip synchronization output.

The proposed model takes an image or a video as input that is processed using a facial recognition system, and simultaneously the real-time text-to-speech model converts the transcripts to the actor's voice. Both the outputs are fed to Wav2Lip plus GANs model to incorporate lip-sync synthesizes. The output generates a new synthesized video from the text input and customizes messages by altering the text using the same driving video. The proposed approach is effective at incorporating low-resolution images or videos of talking head up to 45 seconds, and it is evaluated using the start-of-the-art lip synchronization model-SyncNet to produce a lip synchronized output video with Lip-Sync Error-Distance of 8.14 and Lip-Sync Error-Confidence of 5.65. The obtained results indicate an accurate and effective model compared to previously proposed techniques.

Acknowledgements

I am overwhelmed and grateful to all those who have supported me to put these ideas into action and helped me making in a concrete everlasting project.

I like to offer my heartfelt appreciation and gratitude to my research supervisor, Prof.Gerard Lacey, for guiding me with selecting essential publications for the stimulating interactive question and answering sessions on GANS and the implementations on GANS for videos. Due to regular meetings and discussions, I was able to look outside the box and have different viewpoints to construct a complete and impartial critique. To be honest, I was incredibly moved by his energy, his insight, passion, and motivation. He has instructed me on how to research and how to represent the results of my study in an effective manner. His mentorship was a tremendous privilege and honor. The gift he has given me is priceless, and I thank him profusely.

I am thankful to Trinity College Dublin, School of Computer Science and Statistics, for their assistance during my research work. I am grateful that the institution not only encouraged and supported me but also provided relevant resources and environment to conduct my experiments.

Any attempt at any level cannot be satisfactorily completed without the support and blessings of my mother, Mrs Radhamma. Special thanks to my mother for believing in me and encouraging me to achieve this milestone. Despite their hectic schedules, my friends provided me with new ideas to make my project distinctive. They aided me to acquire diverse information, collect statistics, and guide me from time to time as I worked on this project. Thus, I would like to exclusively thank my friends.

Thanks, Mom!

Contents

List of Figures

List of Acronym

List of Acronym	
GAN	Generative Adversarial Network
TTS	Text-to-Speech
FFmpeg	Fast Forward MPEG
AV	Audio Visual
PSNR	Peak Single to Noise Ratio
SSIM	Structural Similarity Index
LSE-C	Lip-Sync Error-Confidence
LSE-D	Lip-Sync Error-Distance
HD	High Definition
S^3FD	Single Shot Scale-Invariant Face Detector
IOU	Intersection over union
API	Application Programming Interface
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
LRW	Lip Reading in the Wild Dataset
TIMIT	Texas Instruments/Massachusetts Institute of Technology

1 Introduction

1.1 Overview

Due to technological advancements, customising videos have proliferated at a rapid stage. Personalized video messages, TV, ads, YouTube clips and online courses all contain speech videos that focus on the face and mouth of a presenter. It is incredibly tough to modify pre-recorded footage: however particular themes requires to be amplified. Professional editors typically use contemporary tools such as Adobe Premiere or Camtasia to sweep through raw video data. Even with these tools, they have to analyze meticulously where to place cuts to assure that the audio-visual flow is not interrupted. Therefore some modifications require either the contents to be re-shot of the existing video with a new dialogue to be or the sequence to be dubbed. Both techniques are expensive as they require fresh findings and over-dubbing often leads to lip movements and audio malfunctions.

In this study, a talking head video containing phonemes, facial anchors, expressions and lip synchronization for every frame is dynamically annotated. In order to modify a clip, users must simply change the transcript to generate the desired video message. The suggested model receives a picture or video as input, extracts and analyses facial features by virtue of the advanced facial recognition technique, especially lower half of the lip. Simultaneously, a real-time API based technique, text-to-speech conversion is employed by resemble.ai and transforms transcripts into voices. This is fed to a GAN model with an advanced pre-trained discriminator to produce realistic lip synchronized generated video. The obtained output video is evaluated against a ground truth video utilizing the SyncNet model, and the quality

is assessed using PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index Measure).

1.2 Motivation

A few of the vital industry applications that inspired this research are Personalized Messages, for example, doctor's appointment in the form of a video rather than an email or text message with various languages features, Real-time markerless face capture in the CG character animation, and Face capture and tracking techniques, which are the cornerstone for many VR and AR applications. Marketing and Product walk-through with the help of personal digital assistance, Learning and Development. To accomplish visual contents that may be substituted with texts, this research aims to duplicate a photo-realistic face animation technology employed in a digital visual assistant. The production of new audio-visual speech content cannot be made using conventional editing tools or a text-based direct manipulation method. This project presents methodologies that include utilizing the input transcript to fabricate a new video of the speaker who speaks the exact text provided in the transcript.

1.3 Thesis Structure

The thesis is structured accordingly: The introduction containing the overview and inspiration is given in Section 1. In the literature survey in Section 2, previous work relevant to this topic is discussed. Section 3 and Section 4 details the technique used to implement the model. Section 5 provides the evaluation of the model results. Finally, the document is concluded in Section 6, which has further discussions and future work related to this project.

2 Literature Survey

2.1 Previous work

Audio-Driven Facial Reenactment

Audio-driven face regeneration aims to create photo-real films which are synchronized with the voice input stream. Several techniques for facial audio reconstruction (1) were researched, but only a few produce full photographic pictures, natural photo-realism. (2) utilizes President Barack Obama's voice stream to synthesis his talk in elevated videos. For several hours of his speaking a persistent Neural Network is trained to recognize audio mouth shape. The mouth is then composed with the correct 3D match for photo-real reanimation of an original video. It does not apply to the other target actors because of the enormous quantity of training data utilized (17h). On the contrary, this method needs a 2-3-minute video of a target sequence.

(3) Introduced a way to move a still image's lips in order to follow an audio utterance. First, a deep encoder projects the image and audio into a latent space. A decoder then uses the face and audio ligament to fabricate the speaker. The method is trained for 10 hours of data. (4) Provided an additional 2D picture-based approach. They employ a temporary GAN to generate a video with a speaking face and an audio stream as input. The generator feeds the still picture and audio into a structure for encoder decoding using an RNN in order to record temporal connections better. It employs per-frame and series work discriminators to increase temporal cohesion. As a conditioner, the audio signal is also used as an input to compel the synthetic mouth to synchronize with the audio. In (4), the outcomes are improved using a

specialized mouth-audio sync indicator. The 2D picture-based techniques are, in contradiction to this method, confined to a normalized picture space for cut and faced pictures. The drawback encountered is that they are not appropriate to the generation of complete 3D pictures.

Text-Based Video Editing

The video of talking-head is framed to concentrate on a speaker's face is omnipresent in films, TV programs, advertisements, YouTube video logs and online courses. It is difficult to modify pre-recorded videos, but some contents may be hassled, padding words deleted, errors corrected or more widely in accordance with the editor's objective. (5) introduced a text-based video editing method that allows a wide range of changes, like adding, removing, changing words, and persuasive translations and complete phrase fabrication. The analyzed video has a face model (6) and a Viseme search discovers video sections with mouth motions comparable to the altered text. The respective facets of the relevant video segment are combined with the initial photo-realistic sequential specifications accompanied by an in-depth renderer. The strategy is person-specific and needs a one-hour workout from the target player and hence does not apply to brief Online videos. This approach has multiple systems in tandem to interact and pre-processing step take adds a lag in the execution. The major shortcoming of this approach is that the viseme search is sluggish and does not permit interactive outcomes (five minutes for three terms).

Generation of Constrained Talking Face from Speech

This study initially examines the work in the context of speaking faces that are either restricted by their variety of identities or confined to their range of speech. Several recent works (7) (8) on Barack Obama's films have produced a realistic creation of speak face videos. Learn to map the input audio with the associated lip markings. These cannot synthesize for different identities or voices as they are educated on just a particular speaker. They require a great deal of data, usually a few hours, from a given speaker. In a recent piece, (5) it is suggested that an individual actor can be easily edited by adding or deleting sentences. Very recently,

another project (9) aims at minimizing these overhead data with a two-step method, where the model learns speaker-independent characteristics first and then understand maps with the chosen speaker's 5-minute rendering time. They nonetheless train their language-independent network over a much narrower range and incur an in another for each target speaker to provide clean dataset training. The vocabulary is another restriction of current works. Several research works (1) (10) (4) are trained in small data set like GRID (11), LRW (12) and TIMIT (13), which restricts a model substantially by learning about the vast diversity of phoneme-viseme maps in actual films (14). This discussed method primarily focuses on unrestricted lip-syncing facial films that match all target languages, not only identity, voice or lexicon.

Generation of Unconstrained Talking Face from Speech

Despite the increasing number of works on speech led facial creation, only a few works were unexpectedly intended to synchronize films of random personas, accents and dialects. It is not a tiny number of identities or vocabularies that are trained. This enables them to create random identities for any utterance at the time of the exam. As far as it is known, in today's literature, two famous works exist (15) and the more extended version of this method is (3). The works (3) (14) formulation of the challenge of lip synchronization in the wild: Given a brief S-speech segment and a random R-speaker, a lip-synced LD variant of the input faces matching the auditory is the network's goal. Further, with a bottom-half masked to function as a position prior, the LipGAN model also inserts the target face. This was essential as it enabled the produced faces to be reassembled without any further post-processing flawlessly into the actual clip. This trains the discriminator against in-sync or out-of-sync audio-video pairings together with the generator. These two methods are, however, significantly restricted: they perform exceptionally well on static photos of arbitrary identity but create inaccessible lip generation in wild video lip syncs. A prepared, accurate lip-sync discrimination system is utilized, which has not been further trained with the generator, as opposed to the GAN configuration used in LipGAN (14)). This is a crucial decision to obtain significantly better lip synchronization.

Target Models Speech Animation

Several approaches are linked to the development of animated speech patterns (16)(17))(18). They are intended particularly for animated 3D models and not pictures or videos that requires a feature rig and a rig correlation provided by the artist. This technique, in contrast, "encourages" a genuine person who speaks based on text and a monocular record. Audio editing and text-based video. Several audio and video editing programs have been created predicated on time-aligned summaries. Those tools help editors to reduce and re-order audio podcasts speech (19)(20), review video annotation (21), audio descriptive of B-roll segment video material (22), and organized reading video synopsis (23) utilize the time framework to effectively alter numerous script-level images of a scripted scene depending on the editor's higher film idioms. The conversational tone, a video editing tool of (24) is comparable to this approach by cutting, copying and pasting texts for the conversational interviews. Re-order the video by cutting, copying and pasting text, which allows synthesizing new videos only by inputting the new content into transcripts, unlike other prior text-based altering instruments - Synthesis Audio. The synthesis action of new video clips can frequently organically include audio synthesis in transcribed video editing. This approach to video is audio-independent and hence may be utilized in other ways from text to speech (TTS). Two main techniques have been investigated by Traditionally TTS: parametric methods (for example (25)) provide text-based acoustic characteristics and synthesis with the waveform. They sound robotic because of over-simplified acoustic models. Unit selection, however, is a data-driven technique, which builds new waveforms by combining tiny sounds (or units) discovered somewhere in the transcript (26). Influenced by the latter, the (27) VoCo conducts a search for small audio ranges, which may be combined smoothly in the environment from surrounding insertion position, in an existent recording. The video provides several instances of how to synthesize new words in video with the usage of VoCo for the synthesis of the respective audio. The latest TTS methods are dependent on deep learning (28) (10)). These approaches, however, necessitate the target speaker to get many hours of preparation.

Iterative Text-based Editing of Talking-heads Using Neural Retargeting

An improvement to its predecessor, "Text-based editing of talking-head video"(5), is a quick and sophisticated tool for altering talking-head video that uses text and allows for repeated editing. Iteratively, users may modify the speech's text, fine-tune mouth motions to eliminate artifacts, and alter non-verbal parts of the performance by introducing mouth gestures (such as smiling) or adjusting the overall performance style (e.g. energetic, mumble). Their phoneme search method identifies phoneme-level sub-sequences of the original database video, which accurately describe a specified edit rapidly and efficiently. This allows for a rapid iteration loop to be created. To translate the mouth gestures of the original actor to the targeted actor, this approach makes use of an extensive library of video footage of the source actor. This is a brief movie about a target actor that was created. Users may edit a brief video with the target actor, and this program can create a transcript of what they've just edited. To further fine-tune language, lip motions and/or mouth gestures, the user may examine the provided feedback, make changes, and immediately observe how those changes influence the synthetic video. An hour-long recording of a source actor speaking the TIMIT corpus is pre-recorded. While this method improves on (5) in numerous ways, the pre-possessing, which captures the phenomes alignment and 3D face geometry registration, stays unchanged. Finally, neural re-targeting and rendering where the sequencing of full-facial attributes for the target actor and first creates a synthesis picture where the bottom face area is a rendering of the 3D head model using GANS. In this approach, it is required to keep a copy of the phenome in an external repository and then compare changes in text with the repository to identify the best match. In addition, this instrument has a loop feedback time that cannot be changed in real-time, making it ineffective. The demerit of this particular methodology is that it is necessary for the video output to run through the whole pipeline before it can produce a lip-sync.

Neural Voice Puppetry:Audio-driven Facial Reenactment

The incorporation of digital voice assistants into numerous commodities such as smartphones,

televisions, automobiles, and so on are now ubiquitous. Businesses are employing data mining techniques to operate service bots to communicate with their consumers. The goal of these virtual agents is to provide a convenient man-machine interaction while reducing maintenance expenses to a minimum. Consequently, appealing to people by presenting knowledge in a manner that is familiar and comfortable to them would be a big issue.

(29) therefore propose a novel method for audio-driven face visual synthesizing. The goal of this study is to fill the gap in the digital medium by developing Neural Voice Puppetry. This approach utilizes a photo-realism facial graphics approach in the context of a graphical virtual assistant.

Researchers produce photo-realistic visual footage of a human face that is already in synchronization with the audio of the source input provided with an aural sequence of a source person or digital assistant. A neural network using a latent 3d facial supermodel drives an audio-driven facial recreation. The algorithm achieves temporal stability intrinsically through the underpinning 3D structure and then uses neural projections to create realistic result frames. The method extrapolates across individuals, enabling to synthesize recordings of a targeted actor with either the voices of an unknown source person or a synthetic voice created using traditional text-to-speech techniques.

The proposed method was performed admirably on a variety of audio channels and target recordings. However, the approach failed when there were various voices in the source audio. To enable excellent visual tagging, there are various assumptions like: first, the subject films must be clutter-free, just like most reenactment techniques. Second, the audio-visual synchronization of the source recordings must be precise since it influences the efficiency of the reenactment. Furthermore, throughout the subject sequence, it is assumed that the subject actor maintains a consistent speaking style.

3 Methodology

3.1 Framework

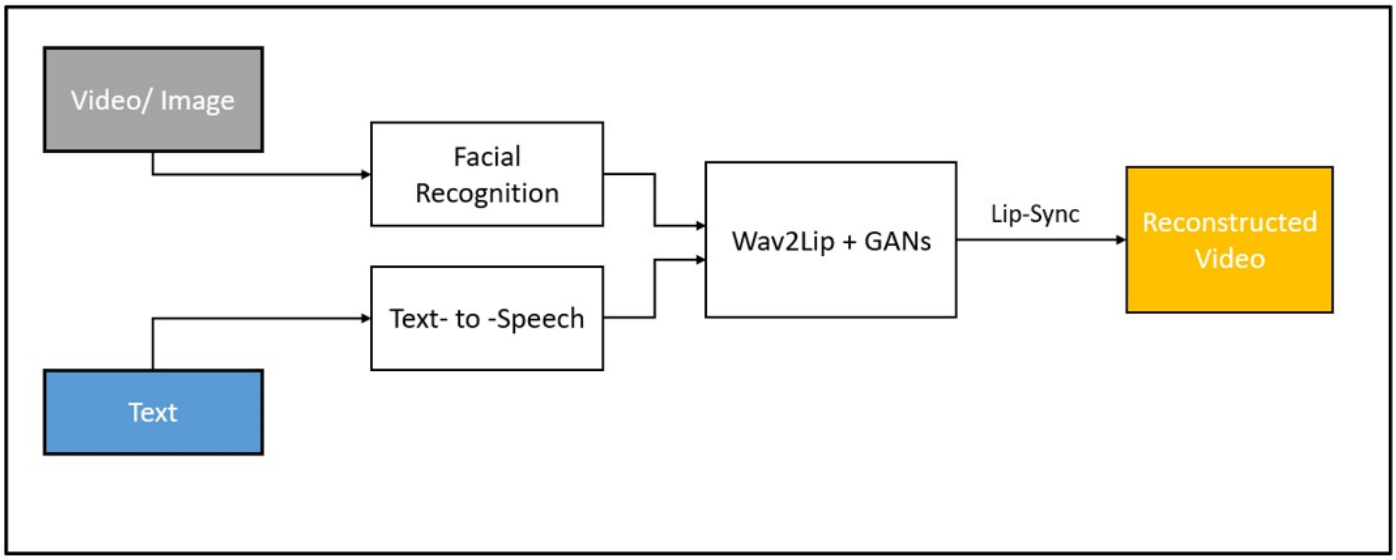

Figure 3.1: Proposed Framework

The methodology for the proposed framework is described below.

Input: The user inputs a Video or an Image, and simultaneously the desired speech is given in the form of text.

Facial Recognition model: The input video or image is processed by this model where the shapes and features are identified, mainly concentrating on the lower part of the face.

Text-to-Speech Conversion Model: This pipeline converts the user input that is in the text format to speech using resemble.ai.

Wav2Lip+GANS Model: The output from the facial recognition model and text-to-speech conversion model is fed to an ensemble model which uses Wav2Lip and GANS combination to perform lip synchronization and produces the desired output.

Each of these models will be discussed in more detail in the below section.

3.1.1 Facial Recognition Model

The Single Shot Scale-invariant Face Detector (S^3FD)(30) is a real-time face detector that performs better on different sizes of faces with a similar deep neural network, notably for tiny faces. To be more specific, trying to tackle the difficulty of anchor-based detectors degrade substantially as entities get smaller. As a result, this study's key contributions can be summarized as follows:

- To handle faces of varied sizes, a face detection framework with a variety of anchor-associated stages and a sequence of appropriate anchor scales is proposed.
- To increase the memory rate of tiny faces, a scale compensated anchor matching approach is leveraged during the implementation.
- To decrease the false - positives rate of tiny faces, a backdrop label with a max-out value has been introduced.
- With concrete results on WIDER FACE, PASCAL face, AFW and FDDB.

3.1.2 Text-to-Speech Conversion Model

This project model was tested with a variety of approaches in order to reach this stage of the process. IBM Watson and Amazon's real-time text to speech conversion API, as well as API of Text to Speech by Google, yields comparable results. Instead of utilizing synthetic voice for the model, a real-time speech-to-text translation model named resemble.ai is employed.

Resemble.AI: All Resemble capabilities, notably generating voices, emotions, and themes, may be accessed via the API. API features consistent resource-oriented URLs, delivers JSON-encoded replies, and employs HTTP response codes, verification, and verbs. Utilizing resem-

ble.ai's real-time technology, the user can register on their website and build a voice dataset of the performer's voice of over 100 recordings. The deep learning algorithm, powered by AI, analyzes the audio and replicates it into the performer's voice.

3.1.3 Wav2Lip + GANS

In this study, lip-sync is applied to synchronize an exogenous identity's conversing facial video to a specific speech clip in real-time. Current work excels at creating precise lip movements on a static image or recordings of individuals viewed throughout the training stage of the study. As a corollary, the video is out of sync with the .wav file in significant portions. A robust lip-sync classifier is used to identify the primary reasons behind this and how to overcome it. Investigating current person speaking techniques for speech to lip creation to arrive at a speaker-independent strategy that does not require any extra speaker-specific data Lip-sync classifier that can compel the generator to continually generate precise, realistic lip motion is adapted to the generator's requirements. Noticed that by applying a powerful lip-sync classifier, the synthesizer is forced to create correct lip movement. However, the morphing portions can occasionally be somewhat fuzzy or include minor artifacts as a result. For this reason, this study trains a primary image quality classifier with the generator in a GAN configuration. Two different discriminators are needed for sync precision and visual quality, respectively. For lip-sync, use one, however for developing feasible faces, use the other:

- When it comes to lip-syncing arbitrary conversing face clips in the environment with undefined speech, Wav2Lip(31) is much more precise than prior attempts.
- Wav2Lip is first-person speaking model to produce videos with lip-sync reliability that equals genuine synchronized footage.

Internal working of the model: Utilization of a customized version of SyncNet(32) - trained lip-sync classifier that is very effective in predicting lip-sync in actual footage. A lip-sync specialist on staff is present. In order to train an elite lip-sync classifier that is suitable for a lip generation task,the following modifications to SyncNet are implemented. First and foremost, color photos are sent into the system, rather than grey-scale frames. Second, the proposed model is much

heavier includes residual convolution layers at the Generators. The third step is to apply a unique gradient descent: cosine-similarity loss plus binary cross-entropy loss. That is, calculate a dot product between the ReLU-activated video and voice embedding v,s to produce a single value between [0, 1]. Every block has a convolutional layer followed by activation of Leaky ReLU. The discriminator is instructed to make the loss or function maximum.The generator minimizes the "expert sync-loss"[1].This powerful classifier or discrimination based purely on the lip-synchronization concept learned from real-world example videos ensures the generated video from the generator produces accurate lip-sync to reduce the lip-synchronization loss. The generator reduces its loss, which is the weighted amount of the recovery loss.

3.2 Technical Requirement

This section discusses the basic technical requirement for the implementation of the project along with the steps to implement from the end user's point of view.

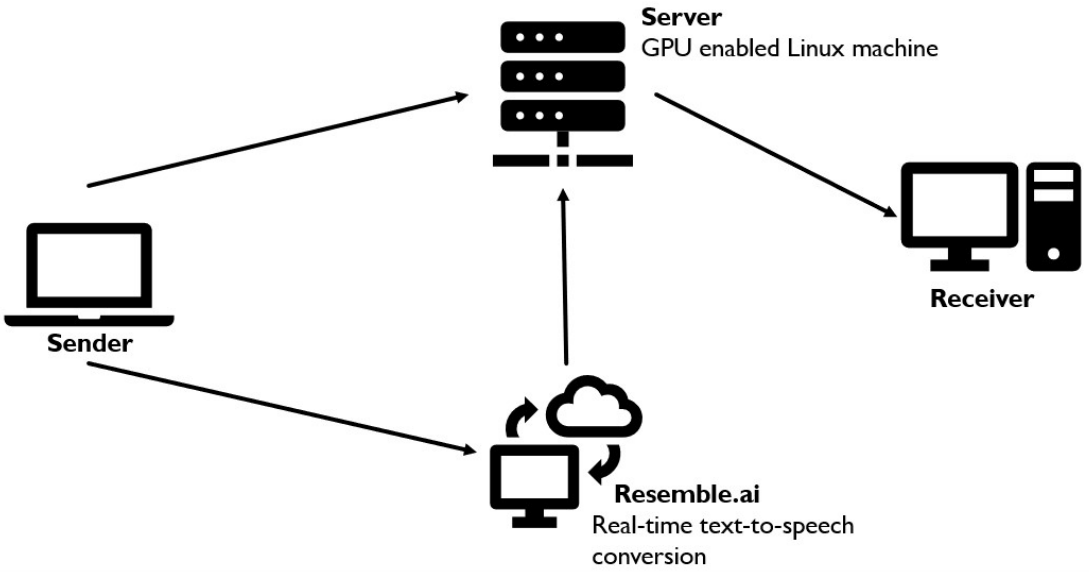

Figure 3.2: Technical Requirement Diagram

3.2.1 Hardware Requirements

Figure 3.1 depicts the hardware requirement and it consists of 3 important components.

[1]Note :that the expert discriminator's weights remain frozen during the training of the generator.

- Server - Server is a cloud or on-premise virtual machine with GPU capability. It is the host machine where the models (face detection and wav2Lip are loaded) and the processing of the video for facial recognition and lip-synchronisation is done. This experiment was conducted with GPU - NVIDIA GeForce 1650 and GPU cloud machine with eight vCpu's. With the accelerated GPU, real-time streaming at 30fps can be achieved.
- Sender - Sender is an end system that initiates the process from the end-user. This system can be either CPU or GPU enabled.
- Receiver - Receiver is the system when the output is played. It needs to have the ability to stream the real-time output video[2].
- Resemble.ai - It is an ingenious real-time API based text-to-speech system that is hosted by a third-party user.

Note: The systems in this experiment are operating on a Linux OS environment.

3.2.2 Software Requirements

Setting up the necessary prerequisites on all workstations (sender, server, and receiver): The entire implementation is scripted using Python3 along with the flask framework.

Creating a conda environment

conda create –name <env> python=3.7

conda activate <env>

It should be noted that Python 3.7 or higher is required for the streaming to function, despite the fact that the wav2lip repo requires 3.6. It is significant for audio-video synchronization (AV sync) in video content since both video and audio are managed independently. <env> is indeed the name identifier of the environment to maintain.

Ascertain that FFmpeg is available.

[2]Note: The sender and receiver can be the same system. These systems need to be connected under the same network.

sudo apt-get install ffmpeg For macOS users: *brew install ffmpeg*

Python dependencies should be installed. The dependencies are defined in the appendix

Pip install -r requirements.txt

Setting up the Callback Server:

By utilizing this server to collect the voice outcome produced by Resemble in order for the Resemble Text-To-Speech to function through API. The next step is to launch the callback server where the decryption takes place

cd resemble_tts
export FLASK_APP=tts_callback_file.py
python -m flask run -p <callback_port>

These ports are connected to the flask application that is developed. The above code executes a callback server on the default port (8080) on the server. Whereas if the server's port is available publicly, the corresponding callback may be accessed at *http://localhost:callback <port>*

But if the server is located within a system, then it must offer a publically available port for such resemble to communicate voice data. Using *https://ngrok.com* or *https://www.heroku.com/* to create an HTTP connection might be one of the options. Next, establish a tunnel to the local port in which the server is receiving by executing:

./ngrok http <callback_port >

The port transmits a publicly available connection to the server. The link is provided as the outcome where the ngrok command was executed in the Forwarding section.

«link» -> http://localhost:«callback_port»

The *<link>* the necessary publicly available callback sessions are passed to the —**resemble** callback URL argument for the respective scripts utilizing Resemble as Text-To-Speech. This connects the external component to the core model as an input.

4 Implementation

4.0.1 Facial Recognition Model

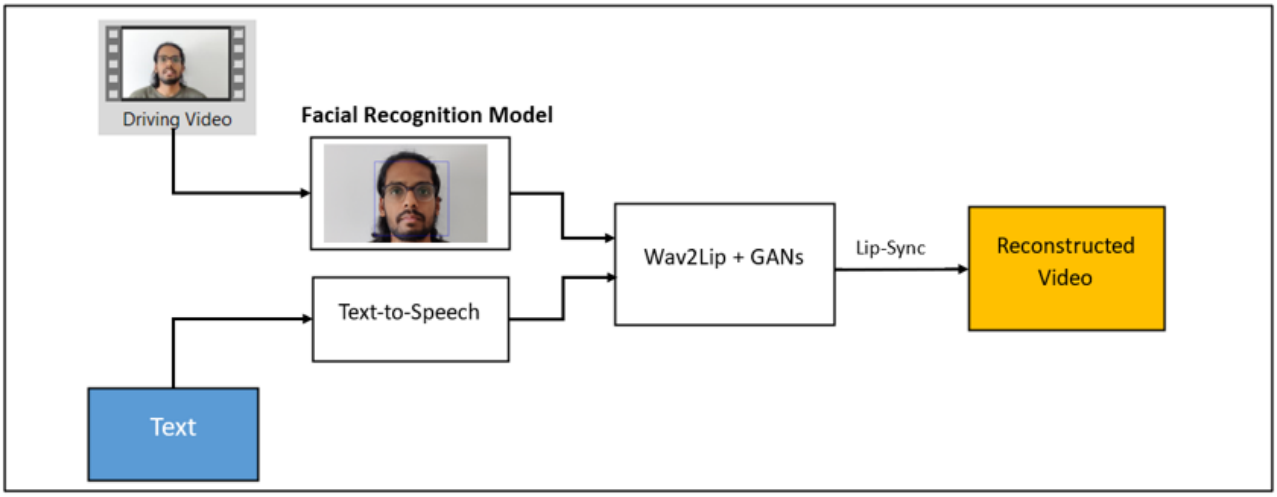

Figure 4.1: Face Recognition system

This is the preliminary stage in the deployment of the S^{3}FD framework (30)[1] 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 S^{3}FD 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 every layer, which is the max-out compensation anchor ap-

[1]The model is trained on 12,800 pictures using several techniques for image augmentation. Script and Comment were obtained and executed the S^{3}FD model in the wav2Lip block.

proach. The framework therefore, guarantees that all facade sizes have appropriate detection characteristics at their respective anchor layers. In a video sequence, it is also beneficial to identify several faces. This also ensures that anchor size corresponds well with the appropriate efficient convolution layer and the varied anchor sizes have the same density.

Figure 4.2: Face Detection block

There reason for Figure 4.2 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.

Load the S^3FD pre-trained weights in the Wav2Lip model by creating a Face Detector folder and place the .pth obtained from the downloads. This model gets triggered by the system when an input image of the face and facial video is provided. The above features of the model can be accessed using **–face** command that activates the model and internally runs the face segmentation by drawing a bounding box around the face. This is then passed to the Wav2Lip+GANS model.

4.0.2 Text-to-Speech Conversion Model

Resemble.AI: All features are accessible using the API to generate voices and text inputs. API provides standardized resources, JSON-encoded responses, and HTTP response codes, authentication. Register on their website and create a speech dataset with the voice of the performance of more than 100 recordings, using the revolutionary innovation. AI-powered profound learning system scans the sound and duplicates it into accent of the actor.

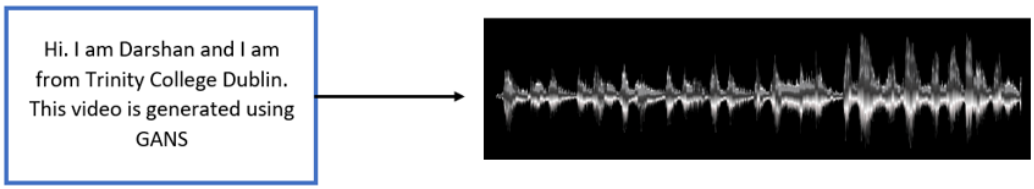

Figure 4.3: Text to Audio Conversion block

Furthermore, the use of the supporting SSML components. The element of speech is necessary. Speak, seems like anguish, sad or happy feeling. Use a emotion to the word or adjust the pitch, amplitude and speed of the phrase in a granular. Phonemes - Demonstrates phonemes in which the actor mimic several dialects and break text included in it. Inserts a pause among words. It can be accessed with an app that processes separately on the service using python in the method.

It used a configuration file with the appropriate details and an API call back using a server code hosted online. Here the app reads the file stored in the path and connects to the resemble.ai using the socket connection that has been established when the flask app exposes the API to be accessed over the internet. Once the text file is read and the text is send to the API to retrieve the appropriate clone voice of the actor and is re-sampled using librosa. This process takes forty seconds to convert a thirty word sentence from text to speech and is significantly lower when it comes to real-time streaming.

Resemble Text-to-Speech Setup:

To access the Resemble API, the following requirements should be met:

Register a profile on the resemble.ai site to upload 60 -100 snippets of audio resources to develop a voice. Resemble permits to train a free voice replica per account. Begin by creating a new project.

Make a file called resemble tts/resemble config.json and fill this with the data. The structure of this json is as follows:

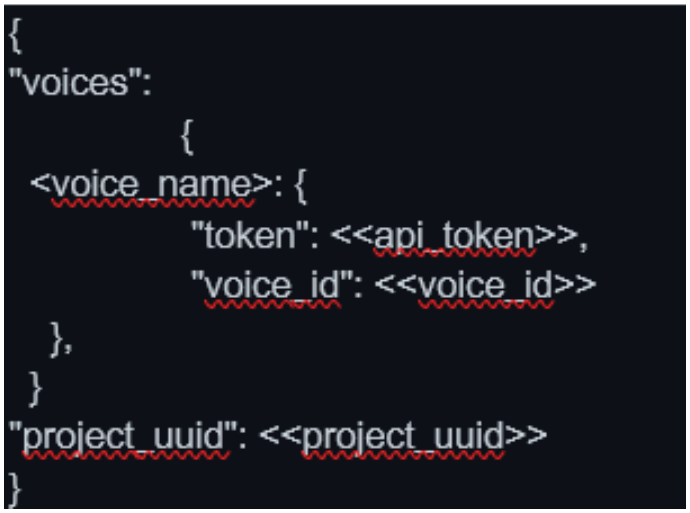

Figure 4.4: Json structure

Here, all variables are strings. The token <api token> is utilized for API access. The voice id is an 8-bit string that resembles the voice ID. In the case of default resemble voices, the voice ID is just like the preset voice's rather than a distinct 8-bit ID. The voice name could be any identifying string for the speech. By providing the voice name to the voice argument for the respective script executions that use Resemble as TTS. The projects uuid is an 8-bit ID for which the speech will be produced utilizing the API.

4.0.3 Wav2Lip with GAN Model

Once the facial features extraction and text-to-speech blocks have been executed the output of these blocks are passed as inputs to the Wav2Lip+GAN models. As discussed in the

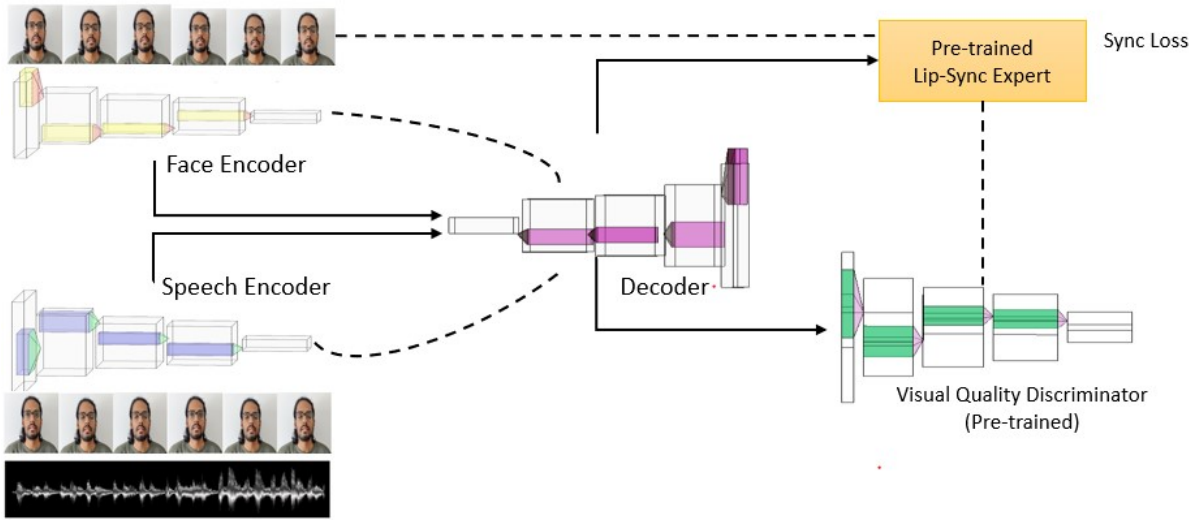

Figure 4.5: Wav2Lip Block

methodology the Wav2Lip model is built on GANS architecture[2].Generator setup resembles Encoder-Decoder block and the Encoder has three elements.

- Face Encoder
- Speech Encoder
- Decoder

Face Encoder is a 2D convolutions network that tracks the lower half of the lip to get the features and this segment is the facial expression P.This is also known as Face Encoder. Another 2D convolutions stack, the Voice Encoder, encodes the speech signals segments S, which is then synthesized with a face that was recognized using the first 2D network. For up sampling, it uses the decoder is likewise a stack of convolutional layers. The features of the face and audio is passed to the decoder using a feature forwarding technique for the up-sampling that helps in ideal reconstruction.

As discussed in the methodology, the proposed approach is equipped with two powerful discriminators one checks the quality of the output video and the second one for the lip-sync loss. The Discriminator is not trained on the input images or video hence makes it independent and can be used to detect the in-sync and out-of-sync video with higher accuracy.

[2]Refer appendix for understanding more about GANS

When the generated video is passed to the sync discriminator for verification and to penalize inaccuracies with lip generation.The output of the generator in penalized to the lower half of the face specifically the lip region. In this way, the model does not need to alter its position, decreasing artifacts substantially. The matching audio segment will likewise be provided as a voice system input and the framework will create the face images crop, however the lower face will become morphic.

In the pipeline, the wav2Lip weights are utilized and placed in the checkpoints folder.[3]. The wav2Lip is loaded after the face detection and text to voice conversion is done. Then for every five frames, the model is executed. It does the lip-sync and produces output and the output is written to an output file of output.mp4. The model is accessed using the inference streaming pipeline and –**checkpoint_path**: the output is stored using the command –**video_file_out**. The interference pipeline in the system also considers various inputs such as –**refactor_size** and **padding**.

Wav2Lip Setup:[4]

Ensure that the model files first from the wav2lip repository is obtained and placed in the respective directory on the system in which the code decryption will execute (Linux server).

GAN model: Wav2Lip/checkpoints should contain the GAN model wav2lip gan.pth. The pre-trained model could be obtained from the URL. (Refer to appendix for the links)

Model for face detection: Wav2Lip/face detection/s3fd.pth should include s3fd.pth. The links to download the pre-trained models is added in the links available in the appendix section.

Once the models are set up, this will complete the server part of the implementation. The flask app is started and the callback server is set up using the commands.(33)
export FLASK_APP=tts_callback_file.py && python -m flask run -p <callback_port>
this opens the local port to talk to server.

[3]the weights are for low resolution video and can be accessed using the link in appendix
[4]https://github.com/Rudrabha/Wav2Lip

Prior to this host, the resemble.ai data will be present on http server so it can be available to the model to access on the server. This is done by hosting on ngrok. ngrok should be download and in the same folder the command should be executed.
./ngrok http <callback_port>
output of the above command is a http URL for resemble.ai http://url-for-resemble.

Changes are made in the inference_streaming_pipeline.py file in the Wav2Lip folder to have real-time access of audio file and when the audio file is ended to the start loading the modelling for lip-synchronization. The command is:

python interface_streaming_pipeline.py -it text –TextToSpeech Resemble –chechpoint_path checkpoints/wav2lip_gan.pth –face /dataset/driving_video.mp4 –tif file –video_file_out generated_video.mp4 -resemble_data /resemble_tts/config.json -v driver_voice_name_on_resemble –url_resemble_network

Custom input is provided using the hyphen - symbol to the wav2lip model. From the above command it can be inferred that the model is loaded from the checkpoints folder. To connect to resemble network -url field needs to populated with the ngrok URL and for fetching of data from resemble network set up the right config.json file, -v is the name give to audio file on the resemble network. -face is providing the input (driving) video and the corresponding generated video is stored with the help of -video_file_out tag.

These model are not trained on the faces in video or voice of the actor in the experiment.

4.1 Experiments

This pipeline is designed on an iterative approach, it started with images, and then videos with various resolution. There are experiments conducted with padding and refactor size (dimension reduction of the video file) to identify the system behaviour and performance.

Here are the ways that these experiments are approached.

4.1.1 Experimentation using Images

Images are the static frame from a video clip, the best place to start the implementation process and the process of face detection is faster even when complied on CPU based machine. The image was placed in the Wav2Lip folder data folder and access using the command **–face** before the model is loaded.

The image was picked and transcripts were converted using the deep learning resemble.ai to give the actor voice and these were combined to produce an output where only the lip was detected by the wav2Lip model and was moved according to the input audio signal. There were no other facial motions such as blinking of eyes, this made the entire set up look non-realistic, like a picture talking.

The experiment was tested on various random images with faces and they produced identical results.

4.1.2 Experimentation using Videos

Video files of various length which includes driving clips of eight seconds and thirty seconds long were considered for this experiment. Along with this text of various lengths were used from a single word, a sentence and concluded the experiment with implementing three sentences. The video loaded used the tag **–face** in the inference pipeline.

Case-1:High resolution Input Videos

- Recording of a High Definition video of eight seconds and thirty seconds was used in this experiment. Placed the file in the dataset folder for the model to access. The text file was placed in the dataset folder in a .txt file.
- Started the call back server using ngrok server on a specific port to convert text to speech.
- The video clip of eight seconds duration with 1080p consumed one minute thirty seconds

for face detection and the lip-synchronization output was produced after two minutes. The entire process was three minutes thirty seconds.

- The video clip of thirty seconds duration with 1080p errored out due to shortage of memory in the GPU based Linux system.

Case-2: Low resolution Input Videos

- Recording of a low resolution video (360p and 240p) of eight seconds and thirty seconds were used in this experiment. Placed the file in the dataset folder for the model to access. The text file was placed in the dataset folder in a .txt file.
- Started the call back server using ngrok server on the specific port to convert text to speech.
- The video clip of eight seconds duration with 240p and 360p was processed within 20 seconds for face detection and the lip-synchronization output was produced inside one min from the time of execution..
- The video clip of thirty seconds duration with (360p and 240p) took two minutes from the start of the server to produce an output.

Results of the HD video and low resolution video are presented in the output section. The CPU based machine errored out due to shortage of processing memory of the HD videos and the for low resolution video the processing time was much larger.

4.1.3 Experimentation using various text lengths

Use video files of low resolution, in this section experiment was conducted to check the process time and response of the video for varied text length. The input text was passed through the command **–it** where it stands for input text.

Case-1: With text length of one line [eleven words]

- Recording of low resolution video of eight seconds and thirty seconds were used in this

experiment. The Text file consisted of 9 words "Hi I am Darshan and I am from Trinity College Dublin". The .txt file in the dataset folder is modified.

- Started the call back server using ngrok server on the specific port to convert text to speech on port 5000.
- Connected to the external call back server hosted on ngrok using the – **Resemble** tag extended from the model file. This will activate the function the input resemble_tts_callback function designed to connect to resemble.ai and open the .txt file, read the text data and pass it to the resemble platform.
- The connection is configured using the JSON file that has the details mentioned above such as project id, voice id. If this connection was successful the text is converted and sent back to the callback function.
- This returned file is now process as saved a audio bit file in .wav format. This is then used by the model as an audio input.
- This above process takes 15-20 seconds to execute using a server that is GPU enabled. This is used to generate a final generated video.

Case-2: With text length of three line.[31 words]

- Recording of low resolution video of eight seconds and thirty seconds was used in this experiment. The text file consisted of the words "Hi I am Darshan and I am from Trinity College Dublin. This video was generated using GANS. I would like to thank professor Gerard Lacey for giving this project to me.". The driving video remained the same. Modifications were done to the .txt file in the dataset folder.
- Steps 2, 3 and 4 are similar to Case-1 of Experimentation using various text lengths.
- This returned file is now processed as saved an audio bit file in .wav format. Library called librosa and pyaudio are used for conversion and processing purposes. This is then used by the model as an audio input. This file is stored in the temp folder and is deleted after the process is completed.

- The above process takes forty five seconds to execute using a server that is GPU enabled. This is used to generate a final generated video. The previous method required the driving video and audio to be of the same length but that is not the case in this experiment.

The noticeable feature while using video of random size and text length is that the loop back of the video when the video ended. For Example, if the video file was of eight seconds and the sentence given was twenty words or longer needed the video to played back once the eight seconds duration was complete. This was automatically handled in the implementation. Irrespective of the video and text length the video adjusted itself to the audio received from the resemble network

5 Results

5.1 Output

- Output [Image as an Input]

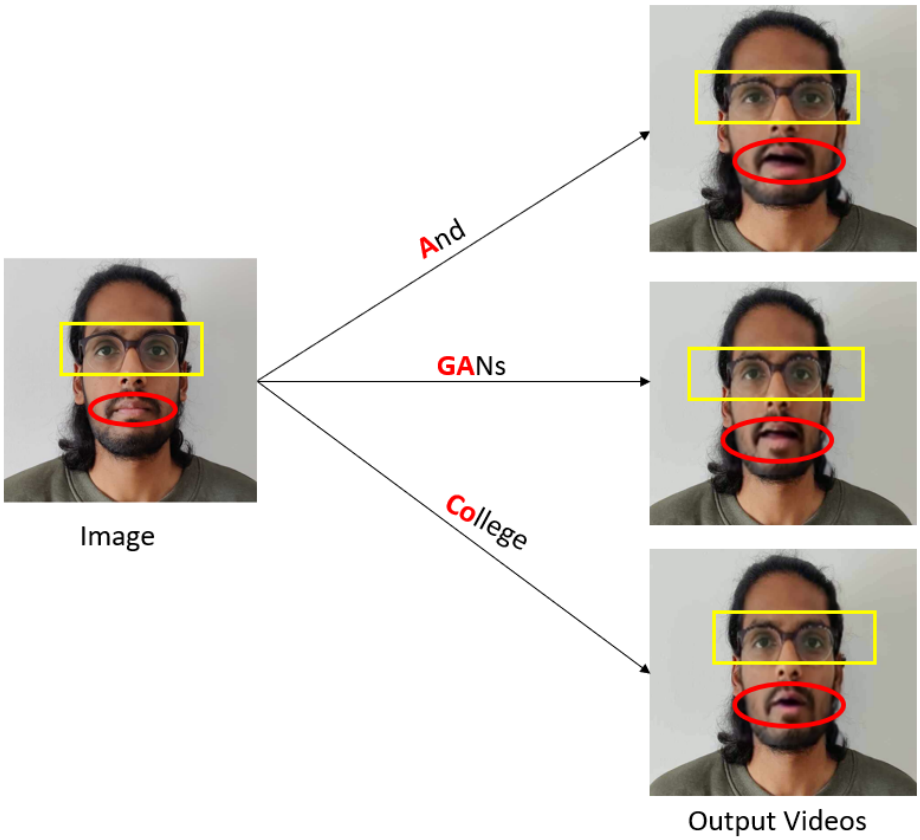

Figure 5.1: Lip-Synced for Images

The generated video file is for an image input and it can be noted that there are no natural movements such as blinking of eye or head movement. There is movement in the Lips and the same is indicated with red highlights is noticed.

- Output [Low Resolution Video as an Input - Input Video - 360p]

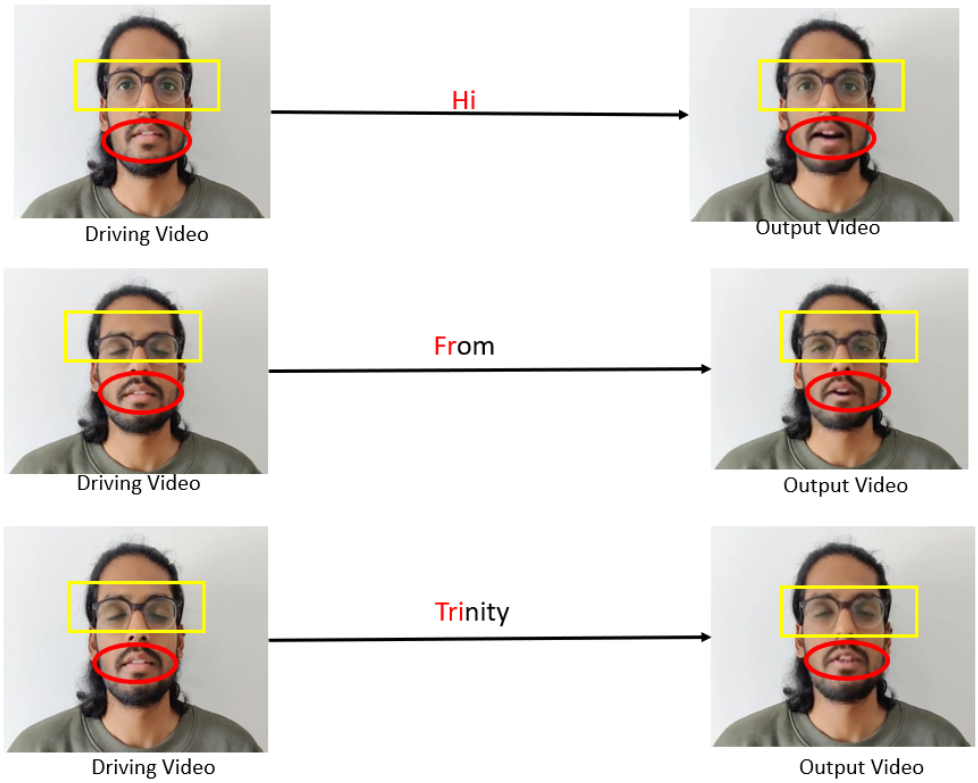

Figure 5.2: Lip-Synced video for input resolution 360p Video

The video file generated using driving video as an input rather than an image tends to give a realistic feel with eyes blinks or head movements along with lip movement. It is noted during the pronunciation of the word **Fr**om the eyes are closed in driving video the generated video mimic this along with lip movement.

- Output [High Resolution Video as an Input]

Figures 5.3 and 5.4 are generated using High Definition input videos. The results from these videos are fuzzy and noticeably blur in the detected face region, In this example, there is a detectable difference that the glasses and it looks blurred and also the a region below the chin is not merged during lip movement process.

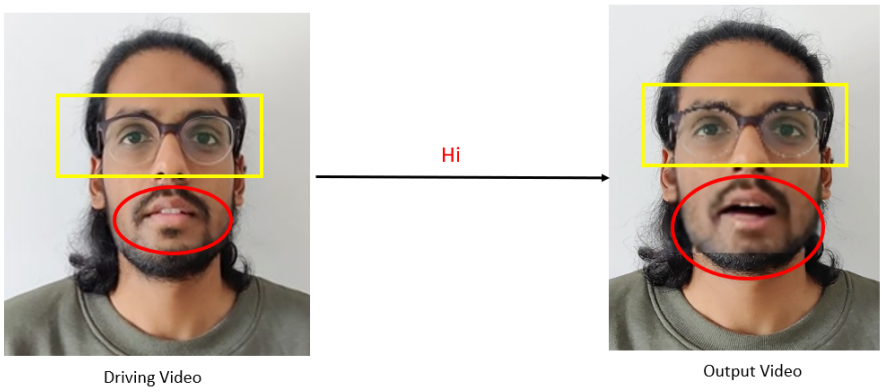

Figure 5.3: Lip-Synced video for HD input video -1080p

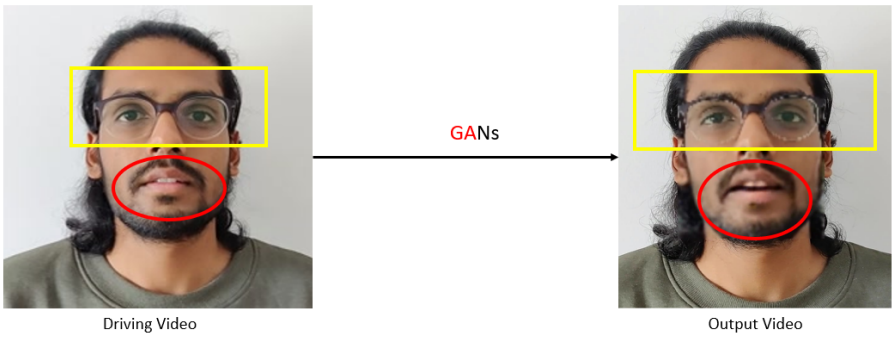

Figure 5.4: Lip-Synced video for HD input video -1080p

5.2 Quantitative Analysis

Detecting video being synchronized with audio is a problem that it is encounter on OTT(Over The Top) mediums for the receiver. A lip-synchronization error is common because it is too small it is unnoticed most of the time. If the reason for the mistake is in the transmission, the video typically delays in accordance with the audio.[1]

SyncNet(32) was primarily used for determining the lip-sync error in videos, and lip movements in faces. This is achieved by separating the audio from the video and factoring them independently. The video of the face specifically the lower half is considered and is extracted from the video using the Histogram Of Gradient method across frames. For the audio file Mel-frequency cepstral coefficient (MFCC) is used to extract its characteristics.

Leveraging the pre-trained SyncNet model (python set up is available) is deployed to measure

[1]These errors are often noticeable – the threshold of an average viewer is around -125ms (the audio signals trails the movie content) to +45ms (the audio signal dominates the film)

measure the lip-sync leads and lags (error), a speech segment that is either aligned with the video (in-sync) or from a different time-step (out-of-sync). The accuracy of SyncNet averaged over a video clip is over 99%(32)[2]. Therefore, this is an automatic technique used for analyzing the accuracy of lip-syncing. **"Lip-Sync Error-Distance"** (minimum values determine better synchronization) and **"Lip-Sync Error-Confidence"** (maximum value suggests a high level of synchronization), that can reliably measure the lip-sync accuracy in videos. The Lip-Sync Error-Distance (LSE-D) is computed between one 5-segment of clip and audio attributes in the $\pm$ 1 second range. Lip-Sync Error-Confidence (LSE-C) high confidence value evaluate to a better the audio-visual binding. A lower confidence value depicts an inaccurate lip movement to voice. The SyncNet takes care temporal coherency of the videotapes(31).

Video	LSE-C	LSE-D	AV Offset
LipGAN	2.33	10.85	7
Generated Video	5.65	8.14	2
Real Video	6.9	7.9	1

Figure 5.5: Quantitative Analysis

The significance of the Audio Visual Offset, LSE-D and LSE-C are discussed in depth in the next section - Discussion

5.3 Qualitative Analysis

The metrics such as SSIM (Structural Similarity Index Measure) is a measure that quantifies image quality generally used for verification of the image or video data used in for transmission and PSNR (Peak Signal-to-Noise Ratio) in image processing refers to the ratio between the highest possible transmission value of any image and the power of noise that intrudes the quality over transmission or representation.

[2]https://github.com/joonson/syncnet_python

For instance, two images I_1 and I_2 and the Mean Square Error (MSE) gives the mean squared distance between the two images and this is in turn used to measure the PSNR which is the peak error.The MSE distance should be small to get better picture quality.

$$MSE = (\frac{1}{N}) \sum_{i=1}^{n} (I_1^i - I_2^i)^2 \tag{1}$$

PSNR is calculated using the MSE.

$$PSNR = 10 \log_{10} (\frac{(MAX)^2}{MSE} \tag{2}$$

where MAX is 255 the max values of an Image pixel. The general accept values of PSNR is in the range of 30 to 50.

Similarly, SSIM actually measures the perceptual variance between two identical images. SSIM is calculated by

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3}$$

where x and y are two windows in the images. Mu is the averages.

SSIM is then a weighted combination luminance, contrast, and structure.

Below table shows the results of PSNR and SSIM:

Sl. No	Ground Truth	Generated video	PSNR	SSIM
1	Input video - 360p	Generated Video - 360p	32.71	U-0.92
2	Input video - 360p	Generated Video - 360p with padding	33.81	U - 0.92
3	Input video - 360p	Generated Video - 360p with resize_factor -1	34.01	U - 0.94
4	Input video - 240p	Generated Video - 240p	33.87	U-0.92
5	Input video - 240p	Generated Video - 240p with padding	34	U-0.92
6	Input video - 240p	Generated Video - 240p with resize_factor -1	34.1	U-0.92
7	Input video - 1080p	Generated Video -1080p	27.4	U-0.88

Figure 5.6: Qualitative Analysis

The values of PSNR and SSIM for the original video compared with the generated video is calculated using FFmpeg and the command used for calculation is:

ffmpeg -i "Generated-Video.mp4" -i "Ground-Truth.mp4" -lavfi "ssim;[0:v][1:v]psnr" -f null -

The above table in Figure 5.2 depicts the PSNR values are greater than 30 and SSIM is greater than 92% for various input resolutions. This justified the quality of the original video and generated video are identical in nature and can for transmission. The next section gives an insights into values presented here.

6 Discussion and Future work

6.1 Discussion

This section delves deeper into the findings, results and their significance in evaluating the project. Analysing the quantitative and qualitative results it can be inferred that:

- The Audio-Visual Offset (AV Offset) provides an intuition if the video is lagging or leading the audio. If it is 0 it means they are in sync and the error - distance is minimum at the Offset 0. The Generated video has a lag of 2 which is negligible when compared to the previous method for lip-synchronization LipGAN having a lag of 7. This means that the video leads the audio.

- Lip-Synchronization Error- Distance is the typical distance that between lip during the video analysis. This checks if the lips are moved and to what extend when the audio is played. The lower value of error-distance played implies that the models' output is having higher lip-synchronization and the generated video has the value of 8.14 when compared to LipGAN with a higher value at 10.85 with a larger AV Offset.

- Lip-Synchronization Error- Confidence from the SyncNet associates a number with the video and if this number is higher it conveys that the model portrays that there is a better lip-synchronization whereas a lower number or number close to 0 shows that the audio and video are out-of-sync. The experiment infers that the video generated using this pipeline produces confidence of 5.65 to that of ground truth video being 6.9. The LipGAN on the other hand has a lower confidence level.

- Finally, the quality of the output video by comparing the PSNR and SSIM of the generated video against the original video is measured. It is evident that when the resolutions were low it produces a higher PSNR of 34.1 and compared to 32.71 when the resolution is 360p. But all the values are in the transmittable range of 30 to 50. When the same with high resolution is compared with the generated video, it depicts that the PSNR value is 27.4. The SSIM values show that both the ground truth video and the generated video are identical with over 92% similarity in reference to lower resolution video and 88% similarity with HD video.

From the resulting frames, the high resolution images tend to provide fuzzy lip and face orientation and this can be seen in Figures 5.3 and 5.4, respectively. Whereas the lower resolution input video sequence produces accurate results seen in Figure 5.2.

The highlights of the discussion with respect to this experiment are

- The user can provide an input in the format of both image and video
- Low resolution video clips or images provide more authentic results compared to high resolution videos
- This model supports user input in the form of text up to 30 words and performs lip synchronization under 40 seconds when complied on a GPU enabled server.

This experiment mainly aims to create an automated system that produces lip-synced output for customising personal messages, which receives user input in the form of a text file along with a video. It is successful in demonstrating an accurate lip-synced video when compared with previous models such as LipGAN(14).

6.1.1 Real-time Streaming

Real-time streaming was tested using a similar setup and the system is designed to consider real-time streaming where the sender transmits the text input in real-time and the video aligns the lip accordingly to produce seamless uninterrupted streaming of low resolution video. The flask app is equipped with this feature to incorporate. A python script for streaming is created

that opened connections to the external port from the sender system and the server receives these values and then forwards it to the resemble network to fetch the appropriate speech syllable to create a lip-sync video. The command is

python input_stream_socket.py -it text -tif terminal –HOST server –PORT 50007 where HOST and port are the external ip address and port of the Server. The Host should white-list the external IP or they need to be on the same network the streaming to successfully compile. It was seen that this was achievable and open up opportunities for further development

6.2 Ethical considerations:

The implemented text-based editing method sets the groundwork for improved customization options for personalized messages and cinematic digital effects. Due to tiny script rewrites, it is also possible to acclimate audio-visual media subject matter to targeted viewing audiences using this editing procedure: for illustration, a fiction video can be refined to viewers of various contexts, or instructional video could be tailored to students of all ages using only textual story line redactions. The accessibility of such an innovation is at a level that some may find indistinct from the original sources. This poses serious and legitimate considerations about the abuse possibilities. However, when utilized to a medium of interaction that is generally regarded to be authentic proof of ideas and intentions, the possibilities of misuse are magnified. The fact that there are unscrupulous actors, who may tend to employ such technology to fabricate personal claims and smear notable persons, which is the main concern to be acknowledged. The output video must be differentiable from the original video when observed carefully. This sheds light on the fact that the original clip is alerted. Before a video is widely circulated, on-screen characters must express their endorsement to any adjustment. Consequently, rules and legislation that penalize misuse while permitting inventive and consensual use cases may be required to address these challenges. A pipeline may be set up to generate realistic video material of an individual using a character-specific audio synthesizer. It is ideal for those who enjoy the content, altering, and generating new films.

6.3 Future work

To exemplify the performance of the proposed generating technique, a prototype pipeline is now being constructed. Using this method on low-frequency devices, which tend to be situated in locations with a weak connection, is hampered by the high computational cost of lip sync and the GPU needs that are required. Text to Speech relies heavily on cloud-based APIs, which is incongruent with the larger objective of decreasing bandwidth use. To allow interactive practical applications such as custom messaging, the video streaming pipeline's latency should be minimized.

There is scope for enhancements with respect to encoding and reconstructing quality, in addition to the technological constraints listed above. Due to its restricted design, it may alter linguistic content (e.g. tone or tempo) or lose non-verbal cues (eg. nodding, raising eyebrows) as well as non-verbal noises (such as laughing) and facial expressions. Required to train the new models to accommodate this non-verbal sustainability will be required in the future and will be retained for future research. This pipeline can create some privacy concerns, and which can be exploited, is typically linked to Deepfakes. This may be prevented by using a technique that restricts the use of the training algorithm to the length of calls.

For users to constantly be aware that they are interacting with created video, new approaches are needed to differentiate between genuine audio/video and synthesized acoustic at the human and cognitive levels. For the broad application of this technology, the government, businesses, and academics must work together to establish protective measures and solve these problems at the legal, technological, and social levels.

A noticeable blob is formed at the lower lip creating a non-realistic feel of the transmitted output is synthetically generated. This is caused by the limited resolution handling capacity of the wav2Lip model. It can be addressed by training the generators and discriminator set up with high resolution video dataset.

6.4 Conclusion

As our social interactions and information consumption grow more audio-visual, large-scale video translations and production are urgently needed. For example, a birthday message can be lip-synced or a doctor's appointment in English to a voice rendered in another original tongue (automatically). To create realistic films, this study presents a novel ensemble approach that accepts user input as text and transforms into voice before fabricating it into a pragmatic video. Face detection algorithms and text-to-speech methods are used to handle the user's video or picture and text inputs, respectively. In order to recreate the video, Wav2Lip and GANS are deployed. In the context of lip-syncing and candid filming, the current technique is feasible. A pre-trained group of lip sync "experts" has been touted as being capable of inducing precise and constant lip synchronization. The pipeline takes care of the intricate details and penalizes wrong lip-synchronized output. The acquired results signify that the lip-synchronization was accurate in comparison with the ground truth video. Highly confident SyncNet network is used for evaluation purposes, there was minimal less lag or lead that was induced using this method. The PSNR and SSIM values depict that the quality of the video sequence can be used for viewing and also transmission.

7 Appendix A

List of additional information with respect to various techniques that provides better understanding of Lip-synchronization.

Let's know GANS

Generative Adversarial Networks (GANs) are discursive models that use deep learning techniques, like convolutional neural networks, to produce the model's output. By outlining the issue as a supervised learning technique, GANs are a brilliant method to train a feature representation. One component is termed the Generator G(x) and the other is the Discriminator D(x). To learn and train complex data like speech, video clips and picture files they both function concurrently. Arbitrary noise is analyzed employing a symmetrical or normal distribution, and then supplied via the generator, then creates a picture as a result of the sample. To train the discriminator, the original pictures are fed into the classifier. Both the Generator and Discriminator go back and forth to produce a realistic output. It is similar to a situation where the Generator is trying to outsmart and Discriminator is trying to catch the generator in the act. GANs produce realistic results in a variety of areas, most prominently in image-to-image translations tasks such as converting summertime to wintertime or daytime to nighttime pictures. In addition, GANS can create lifelike images of items, settings, and individuals that even humans are unable to discern as false.

Facial Reenactment:

The domain of study in which facial video reproduction is prevalent (23)(6)(34). The video

recreation is shown by (35). (17) allows the absolute authority of a targeted actor's head posture, expression and eye gaze based on recent improvements in image-to-image learning (36). Some new techniques allow for controlled face motions of single photos to be synthesized (37)(38)(39) recently demonstrated how well a single picture might be measured to provide a controlled avatar. The Facial reinstatement technique is utilized to visualize the texts-based editing findings and demonstrate how neural face reinstatement can be handled. This paper demonstrates the basics of the facial reattachment technique that has been include as a part of the GAN architecture in various subsequent papers in the implementation of FSGANS, LIPGAN and one-shot Facial reenactment.

Dubbing visual:

Facial recreation provides a framework for visual dubbing, since it enables a target actor to change his expression in a foreign language, in order to complement a dubbing actor's action. Some dubbing techniques are vocational (Speech-driven)(40)(41)(42). Appropriate lip-synchronic methods(2) have been proven. Although this technique can synthesize realistic lip synchronized videos, the new audio must sound the same as the actual speaker while allowing fresh video synthesis with text-based changes. In a controlled setting with a unified backdrop shows outcomes without head movements. By comparison, our 3D and neural rendering method can create delicate phenomena like lip rolling and operates more generally. This technique is the cornerstone of various state of the art lip-synchronization models such a Wav2Lip and LipGANS.

The models of the Deep Generative:

Very lately, academics have suggested the synthesis of pictures and videos via Deep Generative Adversarial Networking (GANs)(43). Approaches produce fresh scratch pictures (1)(31) or condition the input photos summary (36). Recent proof has been given of high-resolution contingent video synthesis (28). In addition, unpaired video-to-video interpretation algorithms require just two training, which necessitate a partnered training dataset. In several applica-

tions, video-to-video interpretation was employed. The re-enactment of the human head, head and upper body (9) and the entire human body (18)(44), for example, has demonstrated outstanding findings.

Synthesis of video-driven, voice-driven and text-driven speech-head:

One typical technique to the synthesis of a video conversational to a chat is to utilize the "driving" videos of a separate actor who has the necessary movement, emotion and voice. Early trials employed video facial markers to find and play the frames of someone else (34) or after warping (45). If choose to have less information required, several techniques may be used to synthesize videos, either by framing and mixing (37) or utilizing the neural networks (38)(46). These approaches produce brief films but are less persuasive for complete words. Several techniques utilize a traced head model to separate characteristics in order to provide convincing results (e.g., posture, identity, expressions) (47)(29). A recorded head model to isolate these characteristics was employed. All these earlier processes require a video to indicate the intended output head movement and emotion. These attributes using text, a usual simpler, less expensive interface.

Another way to synthesize speech is to speak by voice (40) pioneer's work produces a mixture of alignment and mix and has been enhanced in different follow-up works (41)(42). Others utilized non-human dub synthesizing with human voice-driven (48). In conjunction with the audio track (1)(3), some techniques synthesis a speech-head, which is one to a few videos sequence. The outcome is nonetheless a fixed frame or narrowly cropped head plus a rotating inner area and may readily be seen in realistic footage. (8) show that it is possible to generate convincing syntheses utilizing the huge video collection (17hours). (9) produces video of the speech in work parallel with our own. None of these voice drive techniques offer the most refined and efficiency controls necessary for recurrent editing which is a huge demerit.

The approaches that conduct video editing and reconstruction are most closely linked to our work. The synthesis and control of facial emotions are done by (49), however the head is floated in space and not the subject of a picturesque movie. (50) synthesis audio-visual words

from the text, but they are rendered unrealistic by the results of the movies. By clipping, copying and pasting transcribed texts, (19) may modify video conversation. However, they cannot alter phrases or correct flubbed lines by synthesizing unfamiliar phrases. ObamaNet (7) is employing a wide dataset of 17 hours of President's speeches, which reconstructs both visual and voice from the text.(9) exhibit text-based findings as well by adding a text to speech system, while mainly an audio technique. It takes over 1 hour of target footage and consumes hours to create a result, while this technology takes 2-3 minutes of target video and delivers results in around 40 seconds, which is the closest work of (5).This approach provides sophisticated and efficient monitoring which in all prior text-oriented synthesis tools is lacking, but which is important for a good approach for video editing.

Jaccard Overlap

To determine which elements of two categories are alike which are different, the Jaccard similarity index (also known as the Intersection over Union (IOU)) compares their components. The measure of resemblance between two data sets; spectrum is from 0% to 100%. Less dissimilar the two populations are, the greater the proportion. Although it's straightforward to understand, it's very susceptible to small datasets and can produce erroneous findings, particularly with very tiny samples or data sets with insufficient records.

Requirements.txt

Please find the below list of software as a part of the model implementation in the requirements.txt file.

librosa==0.7.0

numpy==1.17.1

opencv-contrib-python>=4.2.0.34

opencv-python==4.1.0.25

torch==1.1.0

torchvision==0.3.0

tqdm==4.45.0

numba==0.48

pyaudio==0.2.11

imageio==2.3.0

imageio-ffmpeg==0.4.3

pandas==0.23.4

PyYAML==5.1

scikit-image==0.14.0

ffmpeg-python==0.2.0

google-cloud-texttospeech==2.2.0

google-cloud-speech==2.0.1

six==1.16.0

requests==2.25.1

Flask==1.1.2

Execution steps:

Step1 – Provide a video or an Image as the driving actor reference to the pipeline.

Step2 – Enter the text (data) that needs to incorporate into the video/Image.

Step3- The Face detection model is employed to detect facial features, specifically lips.

Step4 – As an internal process the text file is converted to speech using a deep learning model employed using resemble.ai

Step5 – Output of step3 and step4 is fed to the model (Wav2Lip+GAN).

Step6 – The reconstructed video is obtained as a result.

Bibliography

[1] L. Chen, R. K. Maddox, Z. Duan, and C. Xu, "Hierarchical cross-modal talking face generation with dynamic pixel-wise loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7832–7841.

[2] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: learning lip sync from audio," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[3] J. S. Chung, A. Jamaludin, and A. Zisserman, "You said that?" *arXiv preprint arXiv:1705.02966*, 2017.

[4] K. Vougioukas, S. Petridis, and M. Pantic, "End-to-end speech-driven realistic facial animation with temporal gans." in *CVPR Workshops*, 2019, pp. 37–40.

[5] O. Fried, A. Tewari, M. Zollhöfer, A. Finkelstein, E. Shechtman, D. B. Goldman, K. Genova, Z. Jin, C. Theobalt, and M. Agrawala, "Text-based editing of talking-head video," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.

[6] P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt, "Reconstruction of personalized 3d face rigs from monocular video," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 3, pp. 1–15, 2016.

[7] R. Kumar, J. Sotelo, K. Kumar, A. de Brébisson, and Y. Bengio, "Obamanet: Photo-realistic lip-sync from text," *arXiv preprint arXiv:1801.01442*, 2017.

[8] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: learning lip sync from audio," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[9] J. Thies, M. Elgharib, A. Tewari, C. Theobalt, and M. Nießner, "Neural voice puppetry: Audio-driven facial reenactment," in *European Conference on Computer Vision*. Springer, 2020, pp. 716–731.

[10] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[11] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006.

[12] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Asian conference on computer vision*. Springer, 2016, pp. 87–103.

[13] N. Harte and E. Gillen, "Tcd-timit: An audio-visual corpus of continuous speech," *IEEE Transactions on Multimedia*, vol. 17, no. 5, pp. 603–615, 2015.

[14] P. KR, R. Mukhopadhyay, J. Philip, A. Jha, V. Namboodiri, and C. Jawahar, "Towards automatic face-to-face translation," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1428–1436.

[15] A. Jamaludin, J. S. Chung, and A. Zisserman, "You said that?: Synthesising talking faces from audio," *International Journal of Computer Vision*, vol. 127, no. 11, pp. 1767–1779, 2019.

[16] P. Edwards, C. Landreth, E. Fiume, and K. Singh, "Jali: an animator-centric viseme model for expressive lip synchronization," *ACM Transactions on graphics (TOG)*, vol. 35, no. 4, pp. 1–11, 2016.

[17] S. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A. G. Rodriguez, J. Hodgins, and I. Matthews, "A deep learning approach for generalized speech animation," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–11, 2017.

[18] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5933–5942.

[19] S. Rubin, F. Berthouzoz, G. J. Mysore, W. Li, and M. Agrawala, "Content-based tools for editing audio stories," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, 2013, pp. 113–122.

[20] H. V. Shin, W. Li, and F. Durand, "Dynamic authoring of audio with linked scripts," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016, pp. 509–516.

[21] A. Pavel, D. B. Goldman, B. Hartmann, and M. Agrawala, "Vidcrit: video-based asynchronous video review," in *Proceedings of the 29th annual symposium on user interface software and technology*, 2016, pp. 517–528.

[22] M. Leake, A. Davis, A. Truong, and M. Agrawala, "Computational video editing for dialogue-driven scenes." *ACM Trans. Graph.*, vol. 36, no. 4, pp. 130–1, 2017.

[23] A. Pavel, C. Reed, B. Hartmann, and M. Agrawala, "Video digests: a browsable, skimmable format for informational lecture videos." in *UIST*, vol. 10. Citeseer, 2014, pp. 2 642 918–2 647 400.

[24] F. Berthouzoz, W. Li, and M. Agrawala, "Tools for placing cuts and transitions in interview video," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–8, 2012.

[25] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *speech communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[26] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *1996 IEEE International Conference on Acoustics,*

Speech, and Signal Processing Conference Proceedings, vol. 1. IEEE, 1996, pp. 373–376.

[27] Z. Jin, G. J. Mysore, S. Diverdi, J. Lu, and A. Finkelstein, "Voco: Text-based insertion and replacement in audio narration," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–13, 2017.

[28] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[29] H. Kim, M. Elgharib, M. Zollhöfer, H.-P. Seidel, T. Beeler, C. Richardt, and C. Theobalt, "Neural style-preserving visual dubbing," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–13, 2019.

[30] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "S3fd: Single shot scale-invariant face detector," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 192–201.

[31] K. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. Jawahar, "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*, 2020, pp. 484–492.

[32] J. S. Chung and A. Zisserman, "Out of time: automated lip sync in the wild," in *Asian conference on computer vision*. Springer, 2016, pp. 251–263.

[33] P. Tandon, S. Chandak, P. Pataranutaporn, Y. Liu, A. M. Mapuranga, P. Maes, T. Weissman, and M. Sra, "Txt2vid: Ultra-low bitrate compression of talking-head videos via text," *arXiv preprint arXiv:2106.14014*, 2021.

[34] I. Kemelmacher-Shlizerman, A. Sankar, E. Shechtman, and S. M. Seitz, "Being john malkovich," in *European Conference on Computer Vision*. Springer, 2010, pp. 341–353.

[35] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2387–2395.

[36] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[37] H. Averbuch-Elor, D. Cohen-Or, J. Kopf, and M. F. Cohen, "Bringing portraits to life," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, pp. 1–13, 2017.

[38] J. Geng, T. Shao, Y. Zheng, Y. Weng, and K. Zhou, "Warp-guided gans for single-photo facial animation," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–12, 2018.

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

[40] C. Bregler, M. Covell, and M. Slaney, "Video rewrite: Driving visual speech with audio," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, pp. 353–360.

[41] Y.-J. Chang and T. Ezzat, "Transferable videorealistic speech animation," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2005, pp. 143–151.

[42] T. Ezzat, G. Geiger, and T. Poggio, "Trainable videorealistic speech animation," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, pp. 388–398, 2002.

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

[44] Z. Liu, Y. Shan, and Z. Zhang, "Expressive expression mapping with ratio images,"

in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 271–276.

[45] P. Garrido, L. Valgaerts, O. Rehmsen, T. Thormahlen, P. Perez, and C. Theobalt, "Automatic face reenactment," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 4217–4224.

[46] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "Ganimation: Anatomically-aware facial animation from a single image," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 818–833.

[47] P. Garrido, L. Valgaerts, H. Sarmadi, I. Steiner, K. Varanasi, P. Perez, and C. Theobalt, "Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track," in *Computer graphics forum*, vol. 34, no. 2. Wiley Online Library, 2015, pp. 193–204.

[48] O. Fried and M. Agrawala, "Puppet dubbing," *arXiv preprint arXiv:1902.04285*, 2019.

[49] H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang, "Talking face generation by adversarially disentangled audio-visual representation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 9299–9306.

[50] W. Mattheyses, L. Latacz, and W. Verhelst, "Optimized photorealistic audiovisual speech synthesis using active appearance modeling," in *Auditory-Visual Speech Processing 2010*, 2010.