VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI – 590018, Karnataka

INTERNSHIP REPORT

ON

"Lip to Speech Synthesis"

Submitted in partial fulfilment for the award of BE degree

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE ENGINEERING

Submitted by:

Aishwarya M S	(1TJ19CS004)
Sai Akshay R	(1TJ19CS060)
Yashaswini D	(1TJ19CS090)
Zaiba	(1TJ19CS091)

Conducted at

VARCONS TECHNOLOGY PVT LTD

Department of Computer Science and Engineering

T JOHN INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University)

No.88/1,Gottigere,Bannerghatta Road,Bengalore-560083

T. JOHN INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University)
No.88/1, Gottigere , Bannerghatta Road,Bengalore-560083

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Internship titled **"Lip to Speech Synthesis"** carried out by **Aishwarya MS(1TJ19CS004), Sai Akshay R(1TJ19CS060), Yashaswini D(1TJ19CS090), Zaiba(1TJ19CS091),** bonafide students of T John Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering**, in **Computer Science** under **Visvesvaraya Technological University,Belgavi** during the year **2022-2023.** This Internship report is approved as it satisfies the academic requirements in respect of internship work pre-scribed for the Bachelor of Engineering Degree.

Signature of Guide	Signature of HOD	Signature of Principal
Sonia Das Assistant Professor Dept.of CSE, TJIT	**Professor Suma R** Head of Department Dept.of CSE, TJIT	**Dr. P Suresh Venugopal** Principal,TJIT

External Viva:

Name of the Examiner — Signature with Date

1. .....
2. **……**

DECLARATION

We **Aishwarya MS(1TJ19CS004), Sai Akshay R(1TJ19CS060), Yashaswini D(1TJ19CS090), Zaiba(1TJ19CS091),** final year student of T john Institute of technology - 560083, declare that the Internship has been successfully completed, in **Varcons Technologies Pvt Ltd** during the academic year **2022-2023**. This report is submitted in partial fulfillment of the requirements for award of Bachelor of Computer Science and Engineering, Bangalore. We further declare to the best of our knowledge and belief, the work reported here is accepted and satisfied, this has not submitted previously for the award of any degree or diploma by us to any other university or institution.

30/09/2022

Bangalore

Aishwarya M S (1TJ19CS004)
Sai Akshay R (1TJ19CS060)
Yashaswini D (1TJ19CS090)
Zaiba (1TJ19CS091)

AKNOWLEDGEMENT

This Internship is a result of accumulated guidance, direction and support of several important persons. We take this opportunity to express our gratitude to all who have helped us to complete the Internship.

We express our sincere thanks to our Principal **Dr Suresh Venugopal** TJIT, for providing us adequate facilities to undertake this Internship.

We would like to thank our Head of Dept Professor Suma R, CSE TJIT, for providing us an opportunity to carry out Internship and for her valuable guidance and support.

We would like to thank our Software Services for guiding us during the period of internship.

We express our deep and profound gratitude to our guide, Sonia Das Assistant Professor CSE TJIT, for her keen interest and encouragement at every step in completing the Internship.

We would like to thank all the faculty members of our department for the support extended during the course of Internship.

We would like to thank the non-teaching members of our dept, for helping us during the Internship.

Last but not the least, we would like to thank our parents and friends without whose constant help, the completion of Internship would have not been possible.

Aishwarya M S (1TJ19CS004)
Sai Akshay R (1TJ19CS060)
Yashaswini D (1TJ19CS090)
Zaiba (1TJ19CS091)

ABSTRACT

Humans involuntarily tend to infer parts of the conversation from lip movements when the speech is absent or corrupted by external noise. In this work, we explore the task of lip to speech synthesis, i.e., learning to generate natural speech given only the lip movements of a speaker. Acknowledging the importance of contextual and speaker specific cues for accurate lip-reading, we take a different path from existing works. We focus on learning accurate lip sequences to speech mappings for individual speakers in unconstrained, large vocabulary settings. To this end, we collect and release a large-scale benchmark data-set, the first of its kind, specifically to train and evaluate the single speaker lip to speech task in natural settings. We propose a novel approach with key design choices to achieve accurate, natural lip to speech synthesis in such unconstrained scenarios for the first time. Extensive evaluation using quantitative, qualitative metrics and human evaluation shows that our method is four times more intelligible than previous works in this space. Please check out our demo video for a quick overview of the paper, method, and qualitative results.

Table of Contents

CHAPTER 1
COMPANY PROFILE

A Brief History of Varcons Technologies

Varcons Technologies, was incorporated with a goal is "To provide high quality and optimal Technological Solutions to business requirements of our clients". Every business is a different and has a unique business model and so are the technological requirements. They understand this and hence the solutions provided to these requirements are different as well. They focus on clients requirements and provide them with tailor made technological solutions. They also understand that Reach of their Product to its targeted market or the automation of the existing process into e-client and simple process are the key features that our clients desire from Technological Solution they are looking for and these are the features that we focus on while designing the solutions for their clients.

Sarvamoola Software Services. is a Technology Organization providing solutions for all web design and development, MYSQL, PYTHON Programming, HTML, CSS, ASP.NET and LINQ. Meeting the ever increasing automation requirements, Sarvamoola Software Services. specialize in ERP, Connectivity, SEO Services, Conference Management, effective web promotion and tailor-made software products, designing solutions best suiting clients requirements.

Varcons Technologies, strive to be the front runner in creativity and innovation in software development through their well-researched expertise and establish it as an out of the box software development company in Bangalore, India. As a software development company, they translate this software development expertise into value for their customers through their professional solutions.

They understand that the best desired output can be achieved only by understanding the clients demand better. Varcons Technologies work with their clients and help them to define their exact solution requirement. Sometimes even they wonder that they have completely redefined their solution or new application requirement during the brainstormingsession, and here they position themselves as an IT solutions consulting group comprising of high caliber consultants.

They believe that Technology when used properly can help any business to scale and achieve new heights of success. It helps Improve its efficiency, profitability, reliability; to put itin one sentence " Technology helps you to Delight your Customers" and that is what we wantto achieve.

CHAPTER 2
ABOUT THE COMPANY

Varcons Technologies is a Technology Organization providing solutions for all web design and development, MYSQL, PYTHON Programming, HTML, CSS, ASP.NET and LINQ. Meeting the ever increasing automation requirements, Varcons Technologies specialize in ERP, Connectivity, SEO Services, Conference Management, effective web promotion and tailor-made software products, designing solutions best suiting clients requirements. The organization where they have a right mix of professionals as a stakeholders to help us serve our clients with best of our capability and with at par industry standards.They have young, enthusiastic, passionate and creative Professionals to develop technological innovations in the field of Mobile technologies, Web applications as well as Business and Enterprise solution. Motto of our organization is to "Collaborate with our clients to provide them with best Technological solution hence creating Good Present and Better Future for our client which will bring a cascading a positive effect in their business shape as well". Providing a Complete suite of technical solutions is not just our tag line, it is Our Vision for Our Clients and for Us, We strive hard to achieve it.

Products of Varcons Technologies.

Android Apps

It is the process by which new applications are created for devices running the Android operating system. Applications are usually developed in Java (and/or Kotlin; or other such option) programming language using the Android software development kit (SDK), but other development environments are also available, some such as Kotlin support the exact same Android APIs (and bytecode), while others such as Go have restricted API access.

The Android software development kit includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and zutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows 7 or later. As of March 2015, the SDK is not available on Android itself, but software development is possible by using specialized Android applications.

Web Application

It is a client–server computer program in which the client (including the user interface and client- side logic) runs in a web browser. Common web applications include web mail, online retail sales, online auctions, wikis, instant messaging services and many other functions. web applications use web documents written in a standard format such as HTML and JavaScript,which are supported by a variety of web browsers. Web applications can be considered as a specific variant of client–server software where the client software is downloaded to the client machine when visiting the relevant web page, using standard procedures such as HTTP. The Client web software updates may happen each time the web page is visited. During the session, the web browser interprets and displays the pages, and acts as the universal client for any web application. The use of web application frameworks can often reduce the number of errors in a program, both by making the code simpler, and by allowing one team to concentrate on the framework while another focuses on a specified use case. In applications which are exposed to constant hacking attempts on the Internet, security-related problems can be caused by errors in the program.

Frameworks can also promote the use of best practices such as GET after POST. There are some who view a web application as a two-tier architecture. This can be a “smart” client that performs all the work and queries a “dumb” server, or a “dumb” client that relies on a “smart” server. The client would handle the presentation tier, the server would have the database (storage tier), and the business logic (application tier) would be on one of them or on both. While this increases the scalability of the applications and separates the display and the database, it still doesn‟t allow for true specialization of layers, so most applications will outgrow this model. An emerging strategy for application software companies is to provide web access to software previously distributed as local applications. Depending on the type of application, it may require the development of an entirely different browser-based interface, or merely adapting an existing application to use different presentation technology. These programs allow the user to pay a monthly or yearly fee for use of a software application without having to install it on a local hard drive. A company which follows this strategy is known as an application service provider (ASP), and ASPs are currently receiving much attention in the software industry.

Security breaches on these kinds of applications are a major concern because it can involve both enterprise information and private customer data. Protecting these assets is an important part of any web application and there are some key operational areas that must be included in the development process. This includes processes for authentication, authorization, asset handling, input, and logging and auditing. Building security into the applications from the beginning can be more effective and less disruptive in the long run.

Web design

It is encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardized code and proprietary software; user experience design.This search engine optimization. The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines. Web design partially overlaps web engineering in the broader scope of web development.

Departments and services offered

Varcons Technologies plays an essential role as an institute, the level of education, development of student's skills are based on their trainers. If you do not have a good mentor then you may lag in many things from others and that is why we at Varcons Technologies gives you the facility of skilled employees so that you do not feel unsecured about the academics. Personality development and academic status are some of those things which lie on mentor's hands. If you are trained well then you can do well in your future and knowing its importance of Varcons Technologies always tries to give you the best.

They have a great team of skilled mentors who are always ready to direct their trainees in the best possible way they can and to ensure the skills of mentors we held many skill development programs as well so that each and every mentor can develop their own skills with the demands of the companies so that they can prepare a complete packaged train

Services provided by Varcons Technologies.

- Core Java and Advanced Java
- Web services and development
- Dot Net Framework
- Python
- Selenium Testing
- Conference / Event Management Service
- Academic Project Guidance
- On The Job Training
- Software Training

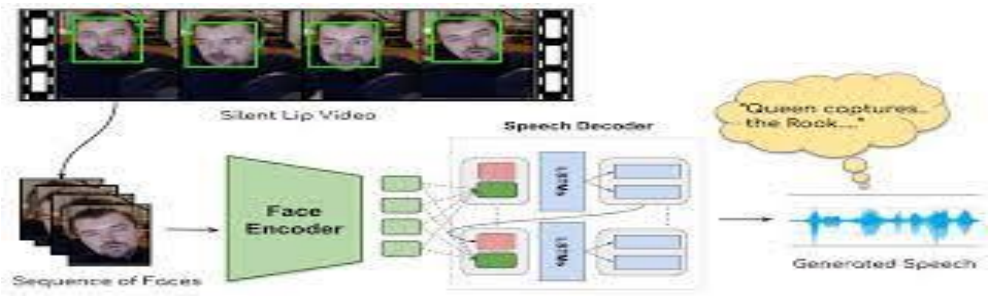
CHAPTER 3

INTRODUCTION

Introduction to ML

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people.

Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.



Introduction to Lip to Speech Synthesis

Babies actively observe the lip movements of people when they start learning to speak. As adults, we pay high attention to lip movements to “visually hear” the speech in highly noisy environments. Facial actions, specifically the lip movements, thus reveal a useful amount of speech information. This fact is also exploited by individuals hard of hearing, who often learn to lip read their close acquaintances over time to engage in more flfluid conversations. Naturally, the question arises as to whether a model can learn to voice the lip movements of a speaker by “observing” him/her speak for an extended period. Learning such a model would only require videos of people talking with no further manual annotation. It also has a variety of practical applications such as

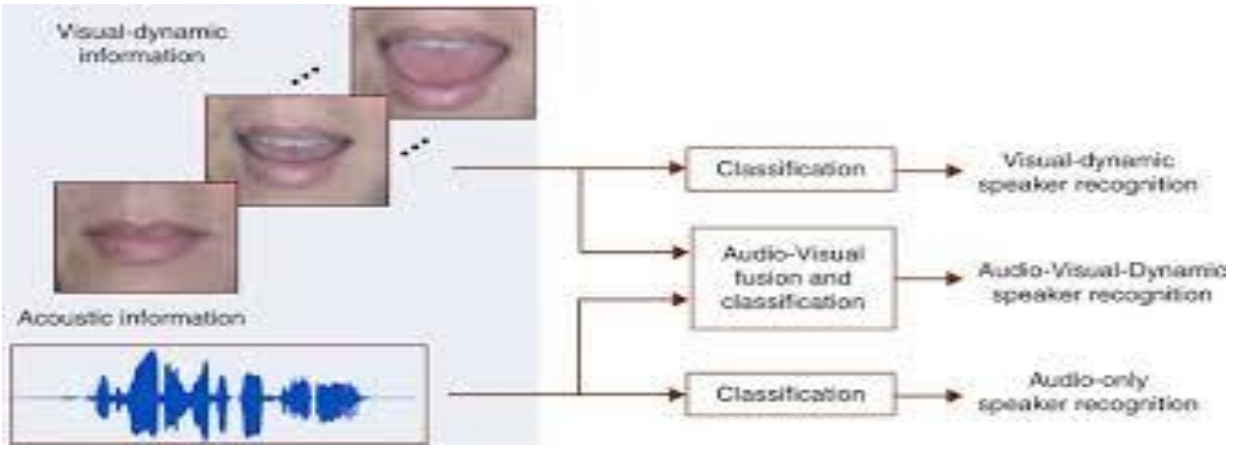
(i) video conferencing in silent environments,

(ii) high-quality speech recovery from background noise ,

(iii) long-range listening for surveillance and

(iv) generating a voice for people who cannot produce voiced sounds (aphonia). Another interesting application would be “voice inpainting” , where the speech generated from lip movements can be used in place of a corrupted speech segment.

In this work, we explore the problem of lip to speech synthesis from a unique perspective. We take inspiration from the fact that deaf individuals or professional lip readers find it easier to lip read people who they frequently interact with. Thus, rather than attempting lip to speech on random speakers in the wild, we focus on learning speech patterns of a specifific speaker by simply observing the person’s speech for an extended period. We explore the following question from a data-driven learning perspective: “How accurately can we infer an individuals speech style and content from his/her lip movements?”.



Problem Statement

I. Unconstrained lip-to-speech is a more challenging task as it aims to generate large vocabulary speech based on real-world videos.

II. To the best of our knowledge, only one prominent work, named Lip2Wav (Prajwal et al. 2020), exists in the current literature. Lip2Wav uses a modifed version of Tacotron2 (Shen et al. 2018) model designed for lip-to-speech. Like Tacotron2, Lip2Wav adopts anautoregressivestructure and takes mel-spectrogram as the generation target.

III. Our method adopts the fow-based non-autoregressive architecture, which can model the frequency correlation effectively and achieve parallel generation.

IV. Our model increases the accuracy of the existing device and generates the vocabulary speech more accurate

CHAPTER 4

SYSTEM ANALYSIS

1. Existing System-Lip to wave.

In this work, we investigate the problem of lip-syncing a talking face video of an arbitrary identity to match a target speech segment. Current works excel at producing accurate lip movements on a static image or videos of specific people seen during the training phase. However, they fail to accurately track the lip movements of arbitrary identities in dynamic, unconstrained talking face videos, resulting in significant parts of the video being out-of-sync with the new audio. We identify key reasons pertaining to this and hence resolve them by learning from a powerful lip-sync discriminator.

2. Proposed System

In our project we have took the source code and increased the accuracy to track the lip movements of arbitary identities in dynamic, by increasing the accuracy we can improve the significant parts of the video and make powerful lip-sync discriminator .

3. Objective of the System

1. They propose lip to wave, which significantly outperforms prior approaches.

2. They introduce a new set of benchmarks/metrics for evaluating the performance of models in this task.

3. They release their own dataset to evaluate the performance of their approach when presented with unseen video-audio content sampled from the wil

CHAPTER 5

REQUIREMENTANALYSIS

Software Requirement Specification

Python

Python offers concise and readable code. While complex algorithms and versatile work flows stand behind machine learning and AI, Python's simplicity allows developers to write reliable systems. Developers get to put all their effort into solving an ML problem instead of focusing on the technical nuances of the language.

PyCharm-IDE

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python,Web and Data science development.

Hardware Requirement Specification

1. RAM: 2GB or higher.
2. HDD: 40GB or higher.
3. Display: 1024 * 768 Resolution Monitor.
4. Keyboard: 104 keys

CHAPTER 6

DESIGN & ANALYSIS

Lip to Speech Generation

While initial approaches to this problem extracted the visual features from sensors or active appearance models, the recent works employ end-to-end approaches. Vid2Speech and Lipper generate low. dimensional LPC (Linear Predictive Coding) features given a short sequence of K frames (K < 15). The facial frames are concatenated channel-wise and a 2D-CNN is used to generate the LPC features. We show that this architecture is very inadequate to model real-world talking face videos that contain significifant head motion, silences and large vocabularies. Further, the low dimensional LPC features used in these works do not contain a significifant amount of speech information leading to robotic, artifificial sounding speech.

A follow-up work of Vid2Speech does away with LPC features and uses high-dimensional mel spectrograms along with optical flows to force the network to explicitly condition on lip motion. While this can be effective for the GRID corpus that has no head movements, optical flow could be a detrimental feature in unconstrained settings due to large head pose changes. Another work strives for improved speech quality by generating raw waveforms using GANs. However, both these works do not make use of the well-studied sequence-to-sequence paradigm that is used for text-to-speech generation thus leaving a large room for improvement in speech quality and correctness.

Lip to Text Generation

In this space as well, several works are limited to narrow vocabularies and small datasets, however, unlike lip to speech, there have been multiple works hat specifically tackle open vocabulary lip to text in-thewild. They employ transformer sequence-to-sequence models to generate sentences given a silent lip movement sequence. These works also highlight multiple issues in the space of lip-reading, particularly the inherent ambiguity and hence the importance of using a language model. Our task at hand is arguably harder, as we not only have to infer the linguistic content, but also generate with rich prosody in the voice of the target speaker. Thus, we focus on extensively analyzing the problem in a *single-speaker* unconstrained setting, and learning precise individual speaking styles.

Text to Speech Generation

Over the recent years, neural text-to-speech models have paved the way to generate high-quality natural speech conditioned on any given text. Using sequenceto-sequence learning with attention mechanisms, they generate melspectrograms in an auto-regressive manner. In our work, we propose Lip2Wav, a modifified version of Tacotron 2 [30] that conditions on face sequences instead of text.

CHAPTER 7

IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigation of the current system and it constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning.

Two major tasks of preparing the implementation are education and training of the users and testing of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required just for implementation.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

TESTING

The testing phase is an important part of software development. It is the Information zed system will help in automate process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. Software testing is carried out in three steps:

1. The first includes unit testing, where in each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether theobjectives have been met. Errors are noted down and corrected immediately.
2. Unit testing is the important and major part of the project. So errors are rectified easily in particular module and program clarity is increased. In this project entire system is divided into several modules and is developed individually. So unit testing is conducted to individual modules.
3. The second step includes Integration testing. It need not be the case, the software whose modules when run individually and showing perfect results, will also show perfect results when run as a whole.

Python Simulation and Code

```
import librosa
import librosa.filters
import numpy as np
import tensorflow as tf
from scipy import signal
from scipy.io import wavfile


def load_wav(path, sr):
    return librosa.core.load(path, sr=sr)[0]

def save_wav(wav, path, sr):
    wav *= 32767 / max(0.01, np.max(np.abs(wav)))
    wavfile.write(path, sr, wav.astype(np.int16))


def save_wavenet_wav(wav, path, sr):
    librosa.output.write_wav(path, wav, sr=sr)


def preemphasis(wav, k, preemphasize=True):
    if preemphasize:
        return signal.lfilter([1, -k], [1], wav)
    return wav


def inv_preemphasis(wav, k, inv_preemphasize=True):
    if inv_preemphasize:
        return signal.lfilter([1], [1, -k], wav)
    return wav
```

```
def start_and_end_indices(quantized, silence_threshold=2):
    for start in range(quantized.size):
        if abs(quantized[start] - 127) > silence_threshold:
            break
    for end in range(quantized.size - 1, 1, -1):
        if abs(quantized[end] - 127) > silence_threshold:
            break

    assert abs(quantized[start] - 127) > silence_threshold
    assert abs(quantized[end] - 127) > silence_threshold

    return start, end


def get_hop_size(hparams):
    hop_size = hparams.hop_size
    if hop_size is None:
        assert hparams.frame_shift_ms is not None
        hop_size = int(hparams.frame_shift_ms / 1000 * hparams.sample_rate)
    return hop_size


def linearspectrogram(wav, hparams):
    D = _stft(preemphasis(wav, hparams.preemphasis, hparams.preemphasize), hparams)
    S = _amp_to_db(np.abs(D), hparams) - hparams.ref_level_db

    if hparams.signal_normalization:
        return _normalize(S, hparams)
    return S


def melspectrogram(wav, hparams):
    D = _stft(preemphasis(wav, hparams.preemphasis, hparams.preemphasize), hparams)
```

```
    S = _amp_to_db(_linear_to_mel(np.abs(D), hparams), hparams) - hparams.ref_level_db

    if hparams.signal_normalization:
        return _normalize(S, hparams)
    return S


def inv_linear_spectrogram(linear_spectrogram, hparams):
    if hparams.signal_normalization:
        D = _denormalize(linear_spectrogram, hparams)
    else:
        D = linear_spectrogram

    S = _db_to_amp(D + hparams.ref_level_db)  # Convert back to linear

    if hparams.use_lws:
        processor = _lws_processor(hparams)
        D = processor.run_lws(S.astype(np.float64).T ** hparams.power)
        y = processor.istft(D).astype(np.float32)
        return inv_preemphasis(y, hparams.preemphasis, hparams.preemphasize)
    else:
        return inv_preemphasis(_griffin_lim(S ** hparams.power, hparams), hparams.preemphasis,
hparams.preemphasize)

    def inv_mel_spectrogram(mel_spectrogram, hparams):
        if hparams.signal_normalization:
            D = _denormalize(mel_spectrogram, hparams)
        else:
            D = mel_spectrogram

        S = _mel_to_linear(_db_to_amp(D + hparams.ref_level_db), hparams)  # Convert back to
linear

        if hparams.use_lws:
```

```
            processor = _lws_processor(hparams)
            D = processor.run_lws(S.astype(np.float64).T ** hparams.power)
            y = processor.istft(D).astype(np.float32)
            return inv_preemphasis(y, hparams.preemphasis, hparams.preemphasize)
        else:
            return inv_preemphasis(_griffin_lim(S ** hparams.power, hparams),
hparams.preemphasis, hparams.preemphasize)

    def _lws_processor(hparams):
        import lws
        return lws.lws(hparams.n_fft, get_hop_size(hparams), fftsize=hparams.win_size,
mode="speech")

    def _griffin_lim(S, hparams):
        angles = np.exp(2j * np.pi * np.random.rand(*S.shape))
        S_complex = np.abs(S).astype(np.complex)
        y = _istft(S_complex * angles, hparams)
        for i in range(hparams.griffin_lim_iters):
            angles = np.exp(1j * np.angle(_stft(y, hparams)))
            y = _istft(S_complex * angles, hparams)
        return y

    def _stft(y, hparams):
        if hparams.use_lws:
            return _lws_processor(hparams).stft(y).T
        else:
            return librosa.stft(y=y, n_fft=hparams.n_fft, hop_length=get_hop_size(hparams),
win_length=hparams.win_size)

    def _istft(y, hparams):
        return librosa.istft(y, hop_length=get_hop_size(hparams), win_length=hparams.win_size)



    def num_frames(length, fsize, fshift):
```

```
    pad = (fsize - fshift)
    if length % fshift == 0:
        M = (length + pad * 2 - fsize) // fshift + 1
    else:
        M = (length + pad * 2 - fsize) // fshift + 2
    return M

def pad_lr(x, fsize, fshift):

    M = num_frames(len(x), fsize, fshift)
    pad = (fsize - fshift)
    T = len(x) + 2 * pad
    r = (M - 1) * fshift + fsize - T
    return pad, pad + r

def librosa_pad_lr(x, fsize, fshift):
    return 0, (x.shape[0] // fshift + 1) * fshift - x.shape[0]

_mel_basis = None
_inv_mel_basis = None

def _linear_to_mel(spectogram, hparams):
    global _mel_basis
    if _mel_basis is None:
        _mel_basis = _build_mel_basis(hparams)
    return np.dot(_mel_basis, spectogram)

def _mel_to_linear(mel_spectrogram, hparams):
    global _inv_mel_basis
    if _inv_mel_basis is None:
        _inv_mel_basis = np.linalg.pinv(_build_mel_basis(hparams))
    return np.maximum(1e-10, np.dot(_inv_mel_basis, mel_spectrogram))

def _build_mel_basis(hparams):
```

```
        assert hparams.fmax <= hparams.sample_rate // 2
        return librosa.filters.mel(hparams.sample_rate, hparams.n_fft, n_mels=hparams.num_mels,
                                   fmin=hparams.fmin, fmax=hparams.fmax)

    def _amp_to_db(x, hparams):
        min_level = np.exp(hparams.min_level_db / 20 * np.log(10))
        return 20 * np.log10(np.maximum(min_level, x))

    def _db_to_amp(x):
        return np.power(10.0, (x) * 0.05)

    def _normalize(S, hparams):
        if hparams.allow_clipping_in_normalization:
            if hparams.symmetric_mels:
                return np.clip((2 * hparams.max_abs_value) * (
                        (S - hparams.min_level_db) / (-hparams.min_level_db)) -
hparams.max_abs_value,
                               -hparams.max_abs_value, hparams.max_abs_value)
            else:
                return np.clip(hparams.max_abs_value * ((S - hparams.min_level_db) / (-
hparams.min_level_db)), 0,
                               hparams.max_abs_value)

        assert S.max() <= 0 and S.min() - hparams.min_level_db >= 0
        if hparams.symmetric_mels:
            return (2 * hparams.max_abs_value) * (
                    (S - hparams.min_level_db) / (-hparams.min_level_db)) - hparams.max_abs_value
        else:
            return hparams.max_abs_value * ((S - hparams.min_level_db) / (-
hparams.min_level_db))

    def _denormalize(D, hparams):
        if hparams.allow_clipping_in_normalization:
            if hparams.symmetric_mels:
```

```
        return (((np.clip(D, -hparams.max_abs_value,
                          hparams.max_abs_value) + hparams.max_abs_value) * -
hparams.min_level_db / (2 * hparams.max_abs_value))
                + hparams.min_level_db)
    else:
        return ((np.clip(D, 0,
                         hparams.max_abs_value) * -hparams.min_level_db /
hparams.max_abs_value) + hparams.min_level_db)
    if not hparams.symmetric_mels:
        return ((D * -hparams.min_level_db / hparams.max_abs_value) +
hparams.min_level_db)
        return (((D + hparams.max_abs_value) * -hparams.min_level_db / (
            2 * hparams.max_abs_value)) + hparams.min_level_db)
```

CHAPTER 8
SNAPSHOTS

CHAPTER 9
CONCLUTION

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project:

- Automation of the entire system improves the efficiency
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating of information becomes so easier
- System security, data security and reliability are the striking features.
- The System has adequate scope for modification in future if it is necessary.

CHAPTER 10

REFERENCE

[1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. The conversation: Deep audio-visual speech enhancement. arXiv preprint arXiv:1804.04121, 2018.

[2] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. Deep lip reading: a comparison of models and an online application. In INTERSPEECH, 2018.

[3] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. Lrs3-ted: a large-scale dataset for visual speech recognition. arXiv preprint arXiv:1809.00496, 2018.

[4] Hassan Akbari, Himani Arora, Liangliang Cao, and NimaMesgarani. Lip2audspec: Speech reconstruction from silent lip movements video. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2516–2520, 2017.

[5] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Lip reading sentences in the wild. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3444–3453. IEEE, 2017.

[6] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In Asian Conference on Computer Vision, pages 87– 103. Springer, 2016.

[7] Martin Cooke, Jon Barker, Stuart Cunningham, and XuShao. An audio-visual corpus for speech perception and automatic speech recognition. The Journal of the Acoustical Society of America, 120(5):2421–2424, 2006.

[8] David A Ebert and Paul S Heckerling. Communication with deaf patients: knowledge, beliefs, and practices of physicians. Jama, 273(3):227–229, 1995.

[9] Ariel Ephrat, Tavi Halperin, and Shmuel Peleg. Improved speech reconstruction from silent video. 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pages 455–462, 2017.

[10] Ariel Ephrat and Shmuel Peleg. Vid2speech: speech reconstruction from silent video. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5095–5099. IEEE, 2017.

[11] Andrew Gibiansky, Sercan Arik, Gregory Diamos, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-speech. In Advances in neural information processing systems, pages 2962–2970, 2017.

[12] Daniel W. Griffifin and Jae S. Lim. Signal estimation from modifified short-time fourier transform. In ICASSP, 1983.

[13] Naomi Harte and Eoin Gillen. Tcd-timit: An audio-visual corpus of continuous speech. IEEE Transactions on Multimedia, 17(5):603–615, 2015.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[15] Lisa I Iezzoni, Bonnie L O'Day, Mary Killeen, and Heather Harker. Communicating about health care: observations from persons who are deaf or hard of hearing. Annals of Internal Medicine, 140(5):356–362, 2004.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal co variate shift. In Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, pages 448–456, 2015.