TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING

Khwopa College Of Engineering
Libali, Bhaktapur
Department of Computer Engineering

A REPORT ON

LIP READING USING CONVOLUTION NEURAL NETWORK

Submitted in partial fulfillment of the requirements for the degree

BACHELOR OF COMPUTER ENGINEERING

Submitted by

Chirag Khatiwada	KCE077BCT001
Bishesh Pokharel	KCE077BCT014
Mahim Rawal	KCE077BCT019
Rowel Maharjan	KCE077BCT027

Under the Supervision of
Er.Mukesh Kumar Pokharel
Department Of Computer Engineering

Khwopa College Of Engineering
Libali, Bhaktapur
2023-24

TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING

Khwopa College Of Engineering
Libali, Bhaktapur
Department of Computer Engineering

A REPORT ON

LIP READING USING CONVOLUTION NEURAL NETWORK

Submitted in partial fulfillment of the requirements for the degree

BACHELOR OF COMPUTER ENGINEERING

Submitted by

Chirag Khatiwada	KCE077BCT001
Bishesh Pokharel	KCE077BCT014
Mahim Rawal	KCE077BCT019
Rowel Maharjan	KCE077BCT027

Under the Supervision of
Er.Mukesh Kumar Pokharel
Department Of Computer Engineering

Khwopa College Of Engineering
Libali, Bhaktapur
2023-24

CERTIFICATE OF APPROVAL

This is to certify that this minor project work entitled **"LIP READING USING CONVOLUTIONAL NEURAL NETWORKS"** submitted by Chirag Khatiwada (KCE077BCT001), Bishesh Pokharel (KCE076BCT014), Mahim Rawal (KCE077BCT019) and Rowel Maharjan (KCE077BCT027), has been examined and accepted as the partial fulfillment of the requirements for the degree of Bachelor in Computer.

.....
Er. Kishor Kumar Adhikari, PhD
External Examiner,
Associate Professor,
Department of Electronics and
Computer Engineering,
National College of Engineering

.....
Er.Mukesh Kumar Pokharel
Project Supervisor
Assistant Lecturer,
Department of Computer Engineering,
Khwopa College of Engineering

.....
Er. Dinesh Gothe
Head of Department,
Department of Computer Engineering
Khwopa College of Engineering

Copyright

Acknowledgement

We would like to express our heartfelt gratitude to our HOD, **Er. Dinesh Gothe**, for providing us with the opportunity and the confidence to act on our will to work on this project and proposal. Also, not forgetting the great help of our supervisor, **Er. Mukesh Kumar Pokharel**, for providing us with the vision to work on our project.

Chirag Khatiwada	KCE077BCT001
Bishesh Pokharel	KCE077BCT014
Mahim Rawal	KCE077BCT019
Rowel Maharjan	KCE077BCT027

Abstract

Decoding text from a speaker's facial movements, or lip reading, is a skill that has great potential for people with speech or hearing impairments. In addition to improving communication, it makes captioning easier in difficult audio situations, such as crowded spaces or far-off speakers. Automated lip reading systems have been made possible by recent advances in computer vision and machine learning. In order to improve the accuracy of visual voice recognition, this research project suggests a hybrid architecture that combines Bidirectional Long Short-Term Memory(BiLSTM) and Convolutional Neural Networks(CNN), taking advantage of their respective advantages. To ensure robustness and generalization, the hybrid model will be trained on an extensive dataset that spans a variety of speakers, languages, and environmental situations. Model parameters will be optimized using fine-tuning processes to provide adaptability in a variety of lip reading circumstances. In order to simplify the implementation process, the project is limited only to visual data.

Keywords: Lip reading, Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (BiLSTM) networks

Contents

List of Tables

List of Figures

List of Symbols and Abbreviation

ADAM	Adaptive Moment Estimation
AL	Automatic Lipreading
ASR	Automatic Speech Recognition
Bi-LSTM	Bi-directional Long Short Term Memory
CNN	Convolutional Neural Networks
GB	Giga Byte
GCP	Google Cloud Platform
GPU	Graphics Processing Unit
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
VM	Virtual Machine
SAT	Speaker Adaptive Training

Chapter 1

Introduction

1.1 Background Introduction

People often communicate through hearing and vision, that is, through voice signals and visual signals. Speech signals often contain more information than visual signals, so many studies have focused on Automatic Speech Recognition (ASR). Although automatic speech recognition (ASR) technology is mature, there are still some unsolved problems, such as how to accurately identify what the speaker is saying in a noisy environment. Lipreading is a visual speech recognition technology that recognizes the speech content based on the motion characteristics of the speaker's lips without speech signals. Therefore, lipreading can detect the speaker's content in a noisy environment, even without a voice signal. Machine learning methods have a great impact on social progress in recent years, which promoted the rapid development of artificial intelligence technology and solved many practical problems. Automatic lip-reading technology is one of the important components of human–computer interaction technology and virtual reality (VR) technology. It plays a vital role in human language communication and visual perception. This project investigates the task of speech recognition from video without audio. The input data to our algorithm is sequences of still images taken from frames of video. We use models to output one of 10 words that are spoken by a face in the input images. We explore and combine a number of different models including CNNs, RNNs and existing publicly available pretrained models to assist in mouth recognition.

1.2 Motivation

The primary objective of developing lip reading software is to increase accessibility by utilizing technology as a tool to remove obstacles to communication for those who have speech or hearing problems. By creating software that can identify and convert lip movements into text, we hope to make it easier for those who are hard of hearing to understand spoken language. In addition, those with speech impairments can now communicate clearly and effectively thanks to this technology. Not only may lip reading increase the accessibility of media for those who have hearing loss, but it can also be used to automatically caption videos, which is useful when the audio quality is not good. The primary goal is to employ technology to enhance the overall standard of living for those with speech or hearing impairments.

1.3 Problem Definition

At present, the ASR can reach a very high recognition rate without severely damaging the speech signal and also can be used in many practical fields. Visual speech recognition is a technology that recognizes the speech content by lip movement characteristics on no speech signal. The information received by the voice channel is two dimensional. Compared with the one-dimensional voice information received by the voice channel, the visual information often contains more redundant information. So visual speech recognition has always been a difficult problem to solve. Visual speech technology is also known as Automatic Lipreading (ALR), which infers the speech content according to the movement of lips in the process of speaking. In real world, there are people with hearing impairment. They communicate through sign language or observing through people's lip movements. But gesture language has problems such as being difficult to learn and understand, and inadequate expression skills. Therefore, ALR technology can help people with hearing impairment communicate with others better to some extent. Also in noisy environments, the speech signal is easily interfered with by the surrounding noise, resulting in the reduction of recognition rate. However, the visual information needed for ALR will not be affected, so ALR can improve the recognition effect of speech recognition in noisy environments. In the field of security, first of all, with the popularity of face recognition technology, there are many attacks against face recognition system, such as photos, video playback, and 3D modeling, etc. Adding lip features can further improve the security and stability of the security system. In the field of vision synthesis, traditional speech synthesis can only synthesize a single voice, and lipreading technology can generate high-resolution speech scene video of specific people. Besides, in sign language recognition, lip movements are also combined to better understand the content of sign language or improve the accuracy of sign language recognition.

1.4 Goals and Objectives

To act as hearing aid for deaf people and voice for mute people.

Chapter 2

Literature Review

The introduction of Artificial Intelligence has greatly enhanced the interaction capabilities of people with hearing and speech related disabilities and impairments. With there being millions of people suffering from these disabilities, the use of suitable lip reading applications and models can allow them to engage in conversations, thus making them be connected to the real world.However, developing such a model is challenging for both designers and researchers. These models should be well designed, perfected, and integrated into smart devices to be widely available to all people in need of speech understanding assistance.
Lip reading can be conducted on the letter,word,sentence,digit or phrase level. It can also be based on video, voice, video with voice or video without voice as input. There have been studies focused on speaker-independent lip reading by adapting a system using the Speaker Adaptive Training (SAT), which was initially used in the speech recognition field. [1]. Research has also been done towards developing an audio-visual speech enhancement framework that operates at two levels: a novel deep-learning based lip-reading regression model and an enhanced, visually-derived Wiener filter for estimating the clean audio power spectrum. [2] The paper [3] uses CNN and Bi-GRU (Bi directional Gated recurrent unit). According to this algorithm, the system is decomposed into two blocks. The first block consist of lip segmentation. The mouth region is extracted using Haar Cascade classifier. Then hybrid active contours model with an improved of the edge by a designed filter is proposed. The second block consists to classify word lip-reading. First, deep convolutional neural network (CNN) is applied to extract frame features from videos who take the results of first block as inputs. Second, the Bi-GRU with two hidden layers is followed by a global average pooling layer. Finally, the word classification results are obtained by Softmax layer. Using segmented lip inputs can yield stronger features, and vastly improve recognition performance. The paper [4] proposes a novel lip-reading driven deep learning approach for speech enhancement that leverages the strengths of deep learning and analytical acoustic modeling. The proposed audio-visual speech enhancement framework operates at two levels: a novel deep learning based lip-reading regression model and an enhanced, visually-derived Wiener filter for estimating the clean audio power spectrum. This discusses the challenges of lipreading and presents LipNet, a model that can map a sequence of video frames to text, trained entirely end-to-end. On the GRID corpus dataset, LipNet achieves 95.2% accuracy in sentence-level, overlapped speaker split tasks.
The [5] uses the MIRAVL-VC1 dataset which outperforms previous datasets in various aspects. It uses modified form of residual network architecture and uses various techniques in data processing, augmentation and visualization to overcome the scarcity of data and improve the performance. Possible insight into possible improvements and future work in expanding the scale and generalization of the model.The paper [6] attempts to use phonemes as a classification schema for lip-reading sentences to explore an alternative schema and to enhance system per-

formance. In the paper [7], they try to improve the accuracy of speech recognition in noisy environments by improving the lip reading performance and the cross-modal fusion effect.The experimental results show that their method could achieve a significant improvement over speech recognition models in different noise environments.The paper [8] makes GhostNet better by creating Efficient-GhostNet. It improves performance with fewer parameters using a new method for communication within the network, making it more efficient.The improved Efficient-GhostNet is used to perform lip spatial feature extraction, and then the extracted features are inputted to the GRU network to obtain the temporal features of the lip sequences, and finally for prediction.The article [9] looks closely at different deep learning methods for lipreading, discussing how they are structured. It also lists various lipreading databases, providing details about them and the techniques used. The paper ends by talking about the challenges in current lipreading methods and suggesting possible future research directions.Lastly we studied about usage of extraction of audio as well as visual features from a video for predicting the spoken sentence/word in [2].The usage of both the features definitely increases the accuracy of the result but that also increases the complexity of the project as models to extract both audio and visual features are to be trained and the database to be used requires both audio and video essence which is more complex to collect and process than the database containing only visual features.

S.N	**Title**	**Summary**
1	Improved speaker independent lip reading using speaker adaptive training and deep neural networks [1]	Focused on speaker-independent lip reading by adapting a system using the Speaker Adaptive Training (SAT), which was initially used in the speech recognition field.
2	Lip-reading driven deep learning approach for speech enhancement [2]	Paper shows audio-visual speech enhancement framework that operates at two levels: a novel deep-learning based lip-reading regression model and an enhanced, visually-derived Wiener filter for estimating the clean audio power spectrum.Usage of both audio and video is done in this research
3	Lip reading of words with lip segmentation and deep learning [3]	Paper emphasizes algorithm combines CNN and Bi-GRU in two blocks. The first focuses on precise lip segmentation using a Haar Cascade classifier and an improved active contours model. The second block employs CNN for frame feature extraction, processed by Bi-GRU, leading to enhanced word lip-reading classification. .

4	Lipnet: End-to-end sentence-level lipreading [4]	The paper proposes a speech enhancement method using deep learning and analytical acoustic modeling. It incorporates a novel lip-reading regression model and an improved Wiener filter for clean audio power spectrum estimation. Addressing lipreading challenges, the paper introduces LipNet, an end-to-end trained model achieving 95.2% accuracy in sentence-level tasks on the GRID corpus dataset.
5	Lip reading word classification [5]	The paper leverages the MIRAVL-VC1 dataset, surpassing prior datasets. Employing a modified residual network architecture and innovative data processing, augmentation, and visualization techniques address data scarcity, enhancing performance. Future work may focus on scaling and generalizing the model for further improvements.
6	Developing phoneme-based lip-reading sentences system for silent speech recognition [6]	The attempts to use phonemes as a classification schema for lip-reading sentences to explore an alternative schema and to enhance system performance is done in the paper.
7	Improving speech recognition performance in noisy environments by enhancing lip reading accuracy [7]	The paper aims to enhance speech recognition accuracy in noisy settings by improving lip reading and cross-modal fusion. Experimental results demonstrate a noteworthy performance boost, surpassing traditional speech recognition models across diverse noise environments.
8	Research on a lip reading algorithm based on efficient-ghostnet [8]	The paper enhances GhostNet to create Efficient-GhostNet, achieving improved performance with fewer parameters. A novel communication method within the network enhances efficiency. Efficient-GhostNet is applied for lip spatial feature extraction, followed by inputting features to a GRU network for temporal lip sequence feature extraction and prediction.

9	Survey of research on lipreading technology [9]	The article extensively examines diverse deep learning approaches for lipreading, elucidating their structures. It provides a comprehensive list of lipreading databases, detailing their characteristics and associated techniques. Concluding, the paper addresses challenges in current lipreading methods and proposes potential avenues for future research.

Table 2.1: Review Matrix with Research Papers and summary of corresponding papers.

Chapter 3

Feasibility Study

3.1 Technical Feasibility

The technical feasibility of a lip-reading application falls on usage of advanced image and video processing techniques in order to capture and process and analyze clear lip movement. The system should be seamlessly running with speech-to-text capability. And usage of natural language processing is a must in order to increase the transcription ability. Real-time processing capability and consideration of hardware requirements is a must for bringing the system into practical use. Through the use of large and varied dataset using an effective model and also keeping the functioning of system in different lighting conditions in mind a consistent user experience is expected.

3.2 Economic Feasibility

The economic feasibility of a project verifies project's financial viability by examining a project's costs, benefits, and risks to determine whether it is financially viable and worthwhile to pursue. For this system, the economic feasibility would involve the cost of training and fine-tuning multiple image and video processing models, implementation of the project its software development keeping required hardware in mind along with its maintenance cost.

3.3 Schedule Feasibility

The scheduling feasibility of a project is an assessment of whether the project and be completed within a specified time frame maintaining quality standard. There are several factors which could impact the schedule, including the availability of the resource materials the project is estimated to take a little over then 3 months. There is expectations to finish the documentation and testing of the system in the specified time frame.

Chapter 4

Requirement Analysis

4.1 Hardware and Software Requirements

4.1.1 Software Requirements

Python and Deep Learning

We used Python as our programming language for this project. Python is a high-level general-purpose computer programming language often used to build websites and software, automate tasks, and conduct data analysis. It is simple, free, easy to use and highly compatible language consisting of a lot of libraries as well as built-in data structures. Having better library ecosystem, better visualization options, platform independence,and it is well known simplicity, consistency and flexibility, Python has proven itself to be one of the best picks for Artificial Intelligence and Machine Learning. Machine learning is a branch of Artificial Intelligence, where we start with an image and extract it's salient features. Then we created a model that describes or predicts the object on the basis of those features. On the other hand, for Deep Learning, we skip the manual step of extracting the features from the object and directly feed the images into a Deep Learning Algorithm, which then predicts the object. Deep Learning can be used to eliminate the limitations of Machine Learning since it makes it easier to handle complex problems as well as helps us predict through huge amount of data with ease too. Thus, Deep learning is a subset of machine learning which provides the ability to machine to perform human-like tasks without human involvement. It provides the ability to an AI agent to mimic the human brain. Deep learning can use both supervised and unsupervised learning to train an AI agent. Here we utilized technique of Deep Learning and concepts of computerized neural networks using Python for the completion for this project. It serves as the primary programming language for lip reading project, providing a flexible and easy-to-read syntax.

OpenCV

OpenCV is an open-source computer vision library that provides a wide range of tools and functions for image and video processing. In our lip-reading project, we use OpenCV to capture and process video frames, apply image preprocessing techniques (such as resizing, filtering, and normalization), and extract relevant features from lip movements such as color or shape information. It also converts the preprocessed frames into a format suitable for input to a PyTorch model.

Tensorflow

TensorFlow, an open-source machine learning framework from Google, empowers developers with a versatile platform for building and deploying artificial intelligence models. Developed by the Google Brain team, it excels in flexibility and scalability. Using a symbolic math library, TensorFlow efficiently defines and trains neural networks, making it a pivotal tool in diverse applications, spanning research to production. Its extensive community support and ecosystem contribute to its popularity, enabling the seamless integration of machine learning into various domains.

Keras

Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation. Meant to be relatively easy to learn, Keras is embedded in TensorFlow and can be used to perform deep learning fast as it provides inbuilt modules for all neural network computations. At the same time, computation involving tensors, computation graphs, sessions, etc can be custom made using the Tensorflow Core API.

Matplotlib

Matplotlib is a plotting library for Python that allows to create a variety of static, animated, and interactive visualizations. In lip reading project, Matplotlib can be used for visualizing different aspects of data and results. For example, we use it to plot training/validation curves, display video frames with overlaid predictions, or create graphs to illustrate the performance of the lip reading model. The Matplotlib is used to visualize the training/validation loss curves during model training.By combining these tools, we can create a comprehensive lip reading system that leverage computer vision, deep learning and visualization to understand and interpret lip movements from video data.

NumPy

NumPy, a vital Python library, empowers numerical computing with high-performance arrays and mathematical functions. Essential for data manipulation and analysis, it underpins diverse scientific and engineering applications. NumPy's array-oriented operations facilitate efficient tasks such as linear algebra and statistics. Its optimized performance and seamless integration make it a cornerstone in the data science landscape, supporting numerous libraries and frameworks.

Imageio

Imageio, a lightweight Python library, simplifies image input and output operations. With a focus on simplicity and efficiency, Imageio supports reading and writing various image formats. Its user-friendly interface and broad compatibility make it a valuable tool for handling image data in diverse applications, from scientific research to multimedia development.

4.2 Functional and Non-Functional Requirements

4.2.1 Functional Requirements

1. **Pre-processing:**
    - Identify and track the face in the video sequence
    - Extract the region of interest (ROI) containing the lips.
2. **Feature Extraction:**
    - Extract the region of interest i.e lips using static slicing function.
3. **Words Recognition**
    - Based on the extracted features, classify the visual information into corresponding words.
    - Employ deep learning models trained on large lip-to-words datasets.

4.2.2 Non-Functional Requirements

1. **Accuracy**
    - The system should achieve a high level of accuracy in translating lip movements to phonemes and subsequently to words and sentences.
    - Specify a target accuracy percentage based on existing benchmarks or project goals.
2. **Real-time performance**
    - The system should process and translate visual information with minimal latency, ideally in real-time.
    - Define an acceptable delay threshold for lip-to-text conversion.
3. **User Interface**
    - The system should have a user-friendly interface for capturing video, displaying results, and interacting with the system.
    - The system should have a user-friendly interface for capturing video, displaying results, and interacting with the system.
4. **Resource Efficiency**
    - The system should be able to run efficiently on available hardware resources, without excessive memory or processing power requirements.
    - Optimize the model and algorithms to minimize resource utilization without compromising accuracy.

Chapter 5

Methodology

5.1 Project requirements(Hardware and software)

5.1.1 Software Development Model

The Agile model is an adaptable and iterative software development process that puts the needs of the client and flexibility first. It breaks the project up into manageable chunks known as sprints or iterations, enabling regular review and modification. Close collaboration between cross-functional teams results in functional software at the conclusion of each iteration. This cycle of iteration guarantees prompt reaction to evolving needs, promoting ongoing enhancement and contentment for the client. The Agile Manifesto's concepts of agile development include a strong emphasis on people and their relationships, functional software, customer collaboration, and adapting to change. In dynamic development contexts, the Agile approach has gained widespread adoption as a framework that encourages efficiency and reactivity.

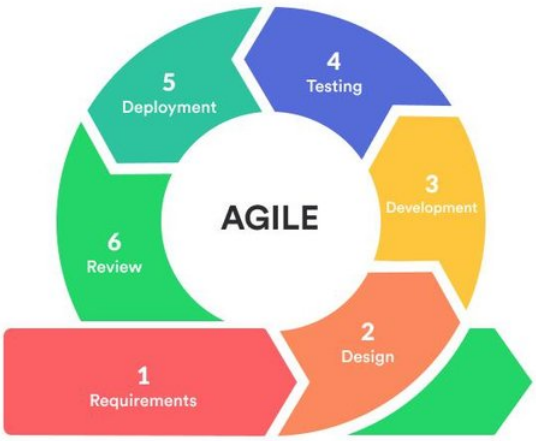


Figure 5.1: Agile Model Image.

5.2 Block Diagram

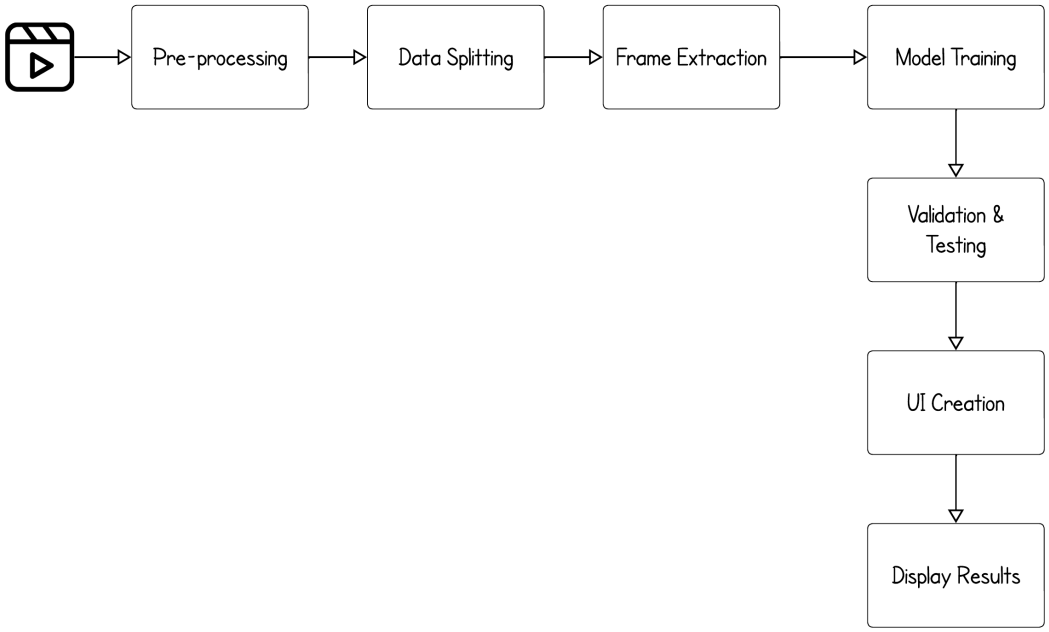


Figure 5.2: Block diagram of the system

5.3 Description of working flow of proposed system

1. **Selection of Dataset:**

   Lipreading datasets (AVICar, AVLetters, AVLetters2, BBC TV, CUAVE, OuluVS1, OuluVS2) are plentiful [10] [11], but most only contain single words or are too small. One exception is the **GRID corpus** (Cooke et al., 2006), which has audio and video recordings of 34 speakers who produced 1000 sentences each, for a total of 28 hours across 34000 sentences.

   We used the **GRID corpus** over other datasets because it is sentence-level and has the most data. The sentences are drawn from the following grammar: ***command*** + ***color*** + ***preposition*** + ***letter*** + ***digit*** + ***adverb***. The categories consist of, respectively, {bin, lay,place, set}, {blue, green, red, white}, {at, by, in, with}, {a, . . . , z}\{w}, {zero, . . . , nine}, and {again, now, please, soon}, yielding 64000 possible sentences.

   For example, two sentences in the data are **"set blue by A four please"** and **"place red at C zero again"**.

(a) Videos

```
0 13000 sil
13000 20000 bin
20000 25750 blue
25750 28250 at
28250 32750 b
32750 44000 nine
44000 54250 now
54250 74500 sil
```

(b) Alignments of Videos

Figure 5.3: Grid Corpus dataset

2. **Dataset Pre-processing:**

   In the selected Dataset **GRID corpus** there are total of 34 talkers(18 male,16 female),but to train our model we used only data of 1 male speaker and 1 female speaker which contains of about 1000 videos and 1000 alignments of each individual video. The data was split into different sets to use for training and validation.**80%** of the data was provided to training set and **20%** of the videos were provided into validation set. We used data from another male speaker for testing.

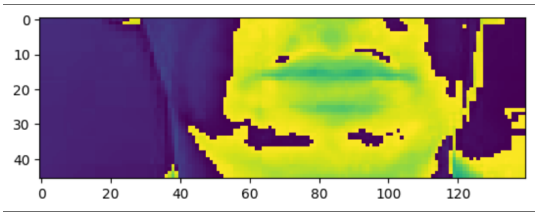
3. **Feature Extraction:** In our project, the region of interest lies within the lip and mouth area of the speaker. To segregate the region of interest from the whole image, the following slicing mechanism with static facial coordinates was used:

   ```
   frames.append(frame[190:236,85:260,:])
   ```

   Here, the .append() method in Python is being used to store sliced video frames. Inside the method, there is a set of facial coordinates used to focus on the required part of the face.

   - **190:236** : This selects rows 190 to 235 (inclusive). It specifies the vertical range of the region of interest.
   - **80:280** : This selects columns 80 to 279 (inclusive). It specifies the horizontal range of the region of interest.
   - **:** : This indicates that no specific color channels are selected and all of the RGB channels are to be used during slicing.

(a) Before Slicing



(b) After slicing

Figure 5.4: Feature Extraction

4. **Model Training:**

- **Convolution 3D layer:**
  The Conv3D layers in the our model perform 3D convolution, extracting spatial features from input volumes. These layers use learnable filters to convolve across three dimensions, capturing complex patterns in video or volumetric data. The extracted features are crucial for tasks like video classification, action recognition, and medical imaging, enhancing model performance.
- **Max-Pooling3D:**
  Max pooling in the model reduces spatial dimensions by selecting the maximum values from neighboring groups in feature maps. Applied after 3D convolutional layers, it efficiently captures essential spatial information, aiding in feature extraction. This downsampling technique enhances the model's capacity for tasks such as video analysis and volumetric data processing.
- **Time-distributed layer:**
  In the model, the TimeDistributed layer processes the output of preceding layers independently at each time step. Applied after 3D convolutional and max-pooling layers, it facilitates the capture of temporal dependencies in sequential data, enhancing the model's ability for tasks like video analysis and spatiotemporal feature extraction.
- **Bi-Directional LSTM:**
  The Bidirectional LSTM layers in the model process sequential data bidirectionally, capturing dependencies in both forward and backward directions. These layers enhance temporal understanding, crucial for tasks like video analysis. With dropout regularization, they mitigate overfitting, improving generalization. The bidirectional nature enables the model to comprehend complex temporal relationships, making it suitable for applications requiring sequential data interpretation.

  Bidirectional Long Short-Term Memory (BI-LSTM) is an extension of the traditional Long Short-Term Memory (LSTM) architecture, com-

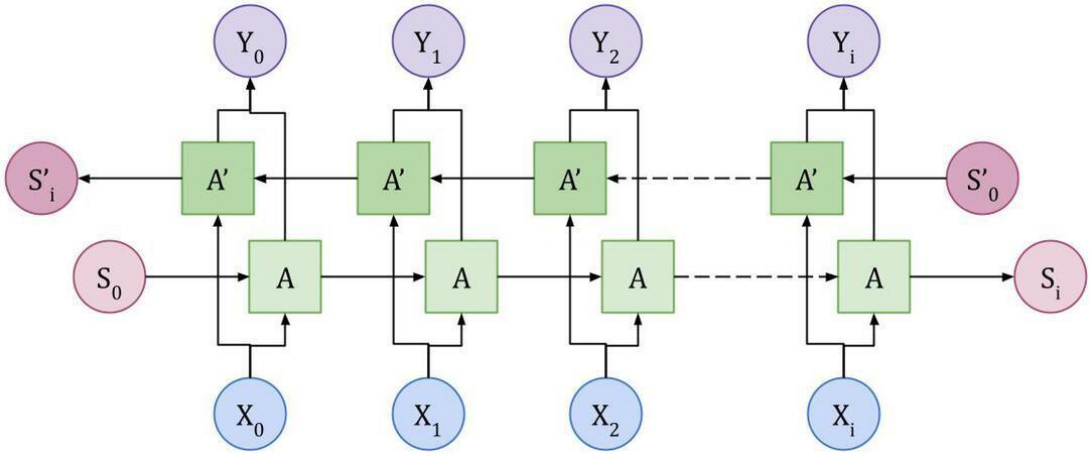


Figure 5.5: Bi-directional LSTM

monly used in deep learning for sequential data processing, such as in natural language processing tasks or time series analysis. The mathematics behind BI-LSTM involves understanding the LSTM cell and how bidirectionality is incorporated.

- **LSTM Cell:**

  (a) **Forget Gate:** A sigmoid function is usually used for this gate to make the decision of what information needs to be removed from the LSTM memory. This decision is essentially made based on the value of h_{t-1} and x_t. The output of this gate is f_t, a value between 0 and 1, where 0 indicates completely getting rid of the learned value, and 1 implies preserving the whole value. This output is computed as:

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f)$$

  where b_f is a constant and is called the bias value.

  (b) **Input Gate:** This gate makes the decision of whether or not the new information will be added into the LSTM memory. This gate consists of two layers: 1) a sigmoid layer, and 2) a "tanh" layer. The sigmoid layer decides which values need to be updated, and the tanh layer creates a vector of new candidate values that will be added into the LSTM memory. The outputs of these two layers are computed through:

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i)$$
$$\tilde{c}_t = \tanh(W_{ch}h_{t-1} + W_{cx}x_t + b_c)$$

  where i_t represents whether the value needs to be updated or not, and $\tilde{c}_t$ indicates a vector of new candidate values that will be added into the LSTM memory. The combination of these two layers provides an update for the LSTM memory in which the current value is forgotten using the forget gate layer through multiplication of the old value (i.e., c_{t-1}) followed by adding the new candidate value

$i_t \cdot \tilde{c}_t$. The following equation represents its mathematical equation:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t$$

where f_t is the result of the forget gate, which is a value between 0 and 1 where 0 indicates completely getting rid of the value, whereas 1 implies completely preserving the value.

(c) **Output Gate:** This gate first uses a sigmoid layer to make the decision of what part of the LSTM memory contributes to the output. Then, it performs a non-linear tanh function to map the values between -1 and 1. Finally, the result is multiplied by the output of a sigmoid layer. The following equations represent the formulas to compute the output:

$$o_t = \sigma(W_{oh} h_{t-1} + W_{ox} x_t + b_o)$$
$$h_t = o_t \cdot \tanh(c_t)$$

where o_t is the output value, and h_t is its representation as a value between -1 and 1.

(d) **Bidirectional Aspect:**
In a BI-LSTM, the sequence is processed in both forward and backward directions. The final hidden state (h_t) for a time step is the concatenation of the forward (h_t^{forward}) and backward (h_t^{backward}) hidden states.

$$h_t = [h_t^{\text{forward}}, h_t^{\text{backward}}]$$

This bidirectional processing allows the model to capture information from both past and future contexts, enhancing its ability to understand sequential dependencies.
In summary, the mathematics behind BI-LSTM involves the computations within the LSTM cell, which includes forget gates, input gates, cell state updates, and output gates. The bidirectional aspect involves processing the sequence in both forward and backward directions to capture information from past and future contexts.

- **Drop-Out:**
The Dropout layers in our model introduces regularization by randomly setting a fraction of input units to zero during training. This prevents overfitting, improving model generalization. Applied after Bidirectional LSTM layers, Dropout enhances the network's robustness and aids in better capturing temporal dependencies in sequential data.
- **Dense Layer:**
The Dense layer is pivotal in our model, is fully connected, linking every neuron to the preceding layer's neurons. With 41 output units, it captures intricate patterns from the learned features, playing a key role in final predictions. Particularly impactful in multiclass classification, this layer contributes to the model's ability to discern and classify diverse patterns in the input data.

5. **Activation Function :**

(a) **Softmax**

The softmax activation function is designed to work with multi-class classification tasks, where an input needs to be assigned to one of several classes.The softmax function uses a vector of real numbers as input and returns another vector of the same dimension, with values ranging between 0 and 1. Because these values add up to 1, they represent valid probabilities. The mathematical formula for the softmax function is given by:

$$\text{Softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Where:

Softmax(z)$_i$ represents the i-th element of the resulting softmax vector.

e denotes Euler's number, the base of the natural logarithm.

Z_i is the $i-th$ element of the input vector The denominator $\sum_{j=1}^{K} e^{z_j}$ represents the sum of the exponentiated values of all elements of $\mathbf{z}$, ensuring that the resulting vector represents a valid probability distribution over the classes.

The softmax activation function is generally used for multi class classification in computer vision and natural language processing. In our project the goal is to classify the lip movement into corresponding phonemes or words. The softmax function is used in output layer of the neural network to provide a probability distribution over possible classes.Each class represents a different word or phoneme, and the class with the highest probability is chosen as the output. By using the softmax function in the output layer of the neural network, the lip reading system can effectively model the uncertainty in predicting spoken words or phonemes based on visual cues from lip movements. It provides a probabilistic interpretation of the predictions, which can be useful for downstream tasks such as language understanding or human-computer interaction.

(b) **Rectified Linear Unit(Relu) function:**

The rectified linear activation function or ReLU is a non-linear function or piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.The Relu function can be mathematically defined by :

$$\text{ReLU}(x) = \max(0, x)$$

Where $\mathbf{x}$ is the input to the function.

The ReLU function outputs the input value if it's greater than zero, otherwise, it outputs zero. Graphically, it looks like a linear function for $x \geq 0$, with a slope of 1, and zero for x<0. The ReLU function is computationally efficient as it can be implemented by simply thresholding

a matrix of activations at zero. It also helps mitigate the vanishing gradient problem, which is a difficulty encountered when training neural networks. In our lip reading project, Relu activation function are used in hidden layers.These hidden layers process the extracted features, transforming them through linear transformations followed by nonlinear activations. By using ReLU activation, the network can introduce nonlinearity, enabling it to learn complex patterns and relationships between the input features.

6. **Optimizer:**
   We used the Adam optimizer, a popular optimization algorithm used in TensorFlow and other deep learning frameworks. It stands for Adaptive Moment Estimation and combines the advantages of two other popular optimization methods: RMSProp and Momentum. ADAM is a stochastic gradient descent algorithm based on estimation of 1st and 2nd-order moments. The algorithm estimates 1st-order moment (the gradient mean) and 2nd-order moment (element-wise squared gradient) of the gradient using exponential moving average, and corrects its bias. The final weight update is proportional to learning rate times 1st-order moment divided by the square root of 2nd-order moment. We used the ADAM optimizer in our source code by simply specifying it as our optimizer of choice.

7. **Testing & Validation:**
   A small subset of the GRID corpus dataset has been used as test data while validation is done alongside training by splitting the initial dataset as mentioned earlier.
   Two metrics, **Word Error Rate(WER)** and **Character Error Rate(CER)** are used as performance evaluation parameters. WER(or CER) is defined as the minimum number of word (or character) insertions, substitutions, and deletions required to transform the prediction into the ground truth, divided by the number of words(or characters) in the ground truth. Note that WER is usually equal to classification error when the predicted sentence has the same number of words as the ground truth, particularly in our case since almost all errors are substitution errors. For eg: Consider a random instance where our model predicts a ground truth of **"bin blue by m zero please"** as **"bin blue by b zero please"**. In this case, the **WER** is calculated to be 0.1667 and the **CER** is 0.04.

8. Description of the model:

Layer Type	Output Shape	Summary
Conv3D(activation='relu')	(None, 75, 46, 175, 128)	3D convolutional layer for spatial feature extraction.
MaxPooling3D	(None, 75, 23, 87, 128)	Max pooling to reduce spatial dimensions.
Conv3D(activation='relu')	(None, 75, 23, 87, 256)	3D convolutional layer for spatial feature extraction.
MaxPooling3D	(None, 75, 11, 43, 256)	Max pooling to reduce spatial dimensions.
Conv3D(activation='relu)	(None, 75, 11, 43, 75)	3D convolutional layer for spatial feature extraction.
MaxPooling3D	(None, 75, 5, 21, 75)	Max pooling to reduce spatial dimensions.
TimeDistributed	(None, 75, 7875)	Applies the flatten operation to each time step.
Bidirectional(LSTM,activation='Tanh')	(None, 75, 256)	Bidirectional LSTM capturing temporal dependencies.
Dropout	(None, 75, 256)	Dropout for regularization.
Bidirectional(LSTM,activation='Tanh')	(None, 75, 256)	Bidirectional LSTM capturing temporal dependencies.
Dropout	(None, 75, 256)	Dropout for regularization.
Dense(activation='softmax')	(None, 75, 41)	Dense output layer with 41 units.

Table 5.1: Description of Each Layer in the Model

9. UI generation

To make the project user-friendly, a graphical interface has been developed using the Python library Streamlit. Within this interface, users can select videos from a dropdown menu. Upon selection, the chosen video is displayed in MP4 format. Additionally, a Mp4 of the video is shown at a slower pace, facilitating observation of lip movements. The machine learning model decodes lip movements into tokens, representing different words. These tokens are then displayed in the interface for users to observe how the model works.

Finally, the interface presents both the original text spoken in the video and the text predicted by the model.

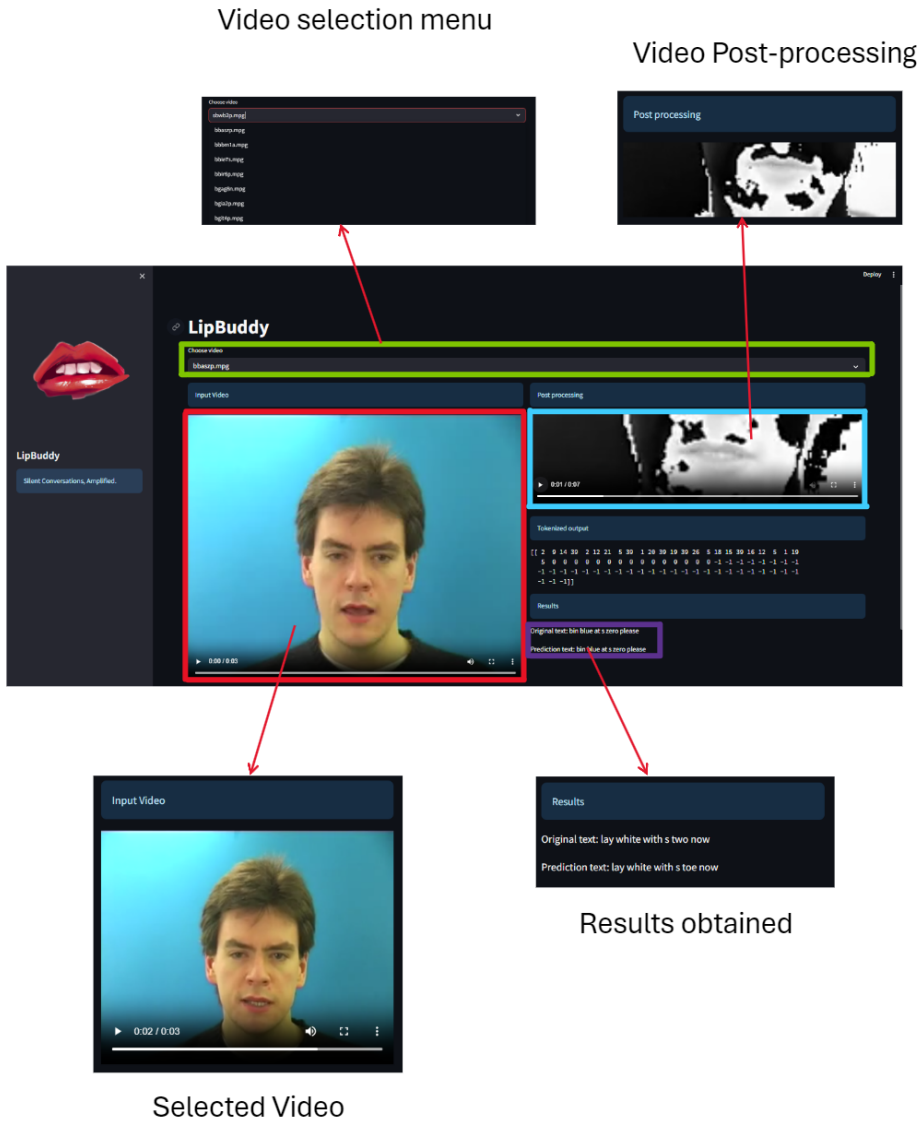


Figure 5.6: UI of LipReading project

Chapter 6

Results and Discussion

6.1 Analysis of Result

6.1.1 Training vs Validation Loss Curve

The model was run over 50 epochs. The Y-axis represents the loss values whereas the X-axis represents the no of epochs. Training and validation losses of **0.7881** & **0.6637** were observed at the end of the 1st epoch and the values at 50th epoch were found to be **0.0169** & **0.0064** respectively.

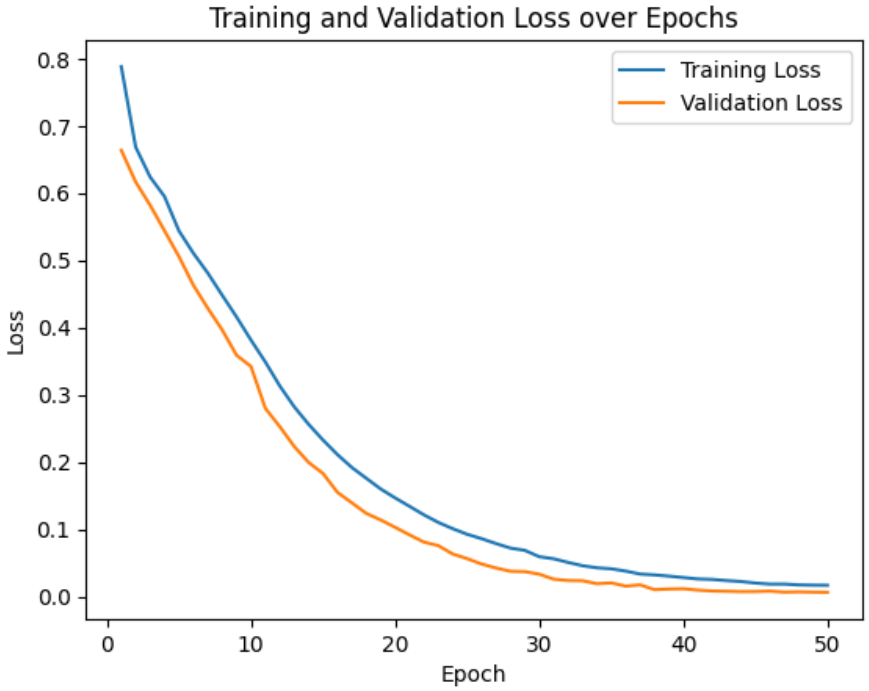


Figure 6.1: Loss Curve

6.1.2 Word Error Rate(WER) and Character Error Rate(CER) Curve

At the beginning, we can observe WER and CER to be very high meaning the predictions are incorrect but as training goes on across increasing epochs, we see the value of these parameters decreasing meaning the model is learning to predict correctly with training. At the 50^{th} epoch both WER and CER is close to 0 meaning the predictions are close to actual words.

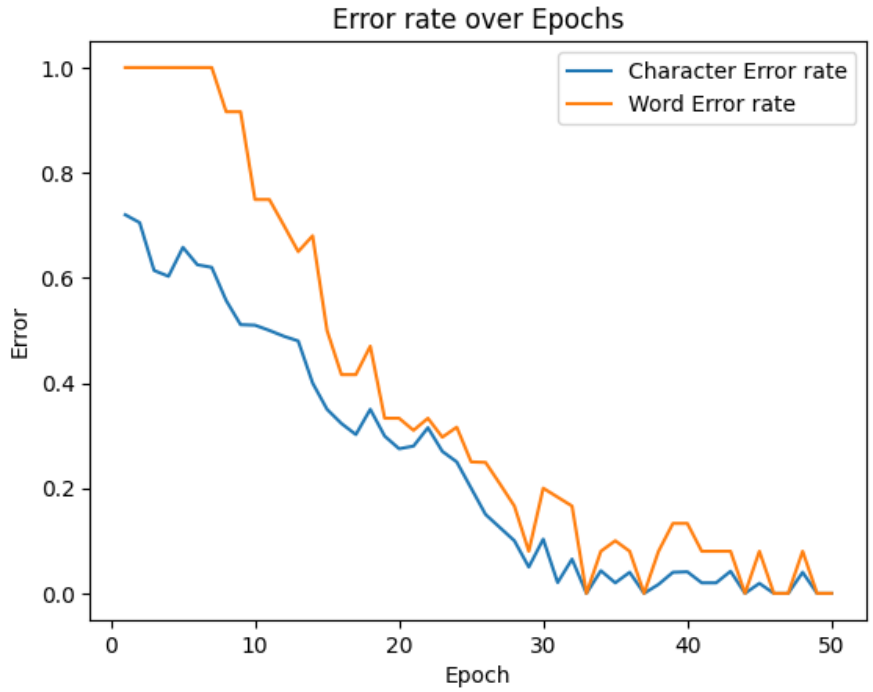


Figure 6.2: WER & CER Curve

6.1.3 Testing Phase

The average values of **Word Error Rate(WER)** and **Character Error Rate(CER)** was calculated to be **0.1706** and **0.0712** over 50 test videos, all different from the ones used in training and validation. Meaning the model was able to predict 83% of the words correctly and 93% of the characters correctly. Even more accuracy on test videos can be achieved by training the model for more number of epochs.

Chapter 7

Conclusion

In conclusion, this Lip-reading project explores a method of deciphering the speaker's words through lip movements without using any audio for processing. We found a way to recognize lip movements and generate its equivalent text with accuracy close to the actual spoken words. Looking forward, our projects aims to contribute towards making lip-reading accessible to those with hearing disabilities as well as mute people. Besides this, the model can be used to decode a person's speech when they are speaking in noisy environments.

Limitation of our Model

1. Only works with videos with resolution of 360*288 pixels.
2. Videos should have certain framing such that lip can be segmented properly.
3. Works on videos of exactly 25fps.

Challenges Faced

1. There are huge variable of data with speaker speaking different words so to choose different speakers with similar words for training was difficult.
2. Processing video is GPU intensive work so capable hardware was hard to come-by.
3. The letter classing of some words are similar so its was hard for model to recognize those words. Eg: 'b' and 'p' have similar lip movements.

Bibliography

[1] I. Almajai, S. Cox, R. Harvey, and Y. Lan, "Improved speaker independent lip reading using speaker adaptive training and deep neural networks," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2722–2726.

[2] A. Adeel, M. Gogate, A. Hussain, and W. M. Whitmer, "Lip-reading driven deep learning approach for speech enhancement," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 3, pp. 481–490, 2019.

[3] M. Miled, M. A. B. Messaoud, and A. Bouzid, "Lip reading of words with lip segmentation and deep learning," *Multimedia Tools and Applications*, vol. 82, no. 1, pp. 551–571, 2023.

[4] Y. M. Assael, B. Shillingford, S. Whiteson, and N. De Freitas, "Lipnet: End-to-end sentence-level lipreading," *arXiv preprint arXiv:1611.01599*, 2016.

[5] A. Gutierrez and Z. Robert, "Lip reading word classification," *Comput Vision-ACCV*, 2017.

[6] R. El-Bialy, D. Chen, S. Fenghour, W. Hussein, P. Xiao, O. H. Karam, and B. Li, "Developing phoneme-based lip-reading sentences system for silent speech recognition," *CAAI Transactions on Intelligence Technology*, vol. 8, no. 1, pp. 129–138, 2023.

[7] D. Li, Y. Gao, C. Zhu, Q. Wang, and R. Wang, "Improving speech recognition performance in noisy environments by enhancing lip reading accuracy," *Sensors*, vol. 23, no. 4, p. 2053, 2023.

[8] G. Zhang and Y. Lu, "Research on a lip reading algorithm based on efficient-ghostnet," *Electronics*, vol. 12, no. 5, p. 1151, 2023.

[9] M. Hao, M. Mamut, N. Yadikar, A. Aysa, and K. Ubul, "A survey of research on lipreading technology," *IEEE Access*, vol. 8, pp. 204 518–204 544, 2020.

[10] Z. Zhou, G. Zhao, X. Hong, and M. Pietikäinen, "A review of recent advances in visual speech decoding," *Image and Vision Computing*, vol. 32, no. 9, pp. 590–605, 2014.

[11] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II 13*. Springer, 2017, pp. 87–103.

Appendix:

Source Code For Lip-Reading Algorithm in Python

The source code corresponding to the developed algorithm is hosted on GitHub and included in the repository below:

`https://github.com/chiragooner/Minor-Project`

Feedback and suggestions are welcome and greatly appreciated, as they help improve the project for everyone. It is encouraged to browse the code, submit issues, and even contribute to the project if you are interested. Any input from the community will be valued and acknowledged.

For any issues, suggestions, or requests for access, feel free to reach out at:

Chirag Khatiwada: KCE077BCT001@khwopa.edu.np

Bishesh Pokharel: KCE077BCT014@khwopa.edu.np

Mahim Rawal: KCE077BCT019@khwopa.edu.np

Rowel Maharjan: KCE077BCT027@khwopa.edu.np