LipScribe - Project Report

A Project Report
Presented to
The Faculty of the College of
Engineering

San Jose State University
In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Software Engineering

By

Gayathri Pulagam, Rishitha Bandi, Shreya Goyal, Surabhi Govil

12/2021

APPROVED

Gopinath Vinodh, Project Advisor

Dan Harkey, Director, MS Software Engineering

Rod Fatoohi, Department Chair, Computer Engineering

ABSTRACT

LipScribe - a digital assistant that helps people with hearing loss, by converting a speaker's lip movements into text

By Gayathri Pulagam, Rishitha Bandi, Shreya Goyal, Surabhi Govil

According to the National Institute on Deafness and other Communication Disorders (NIDCD) approximately 7.5 million people in the United States alone have difficulty using their voice [1]. NIDCD also reported that about 5% of children suffer from speech disorders with no known cause [1]. Commercially available devices [2] are either too expensive or restrictive.

Sravi, an existing solution that uses AI to convert phrases to text for a person, currently recognizes 20 predefined phrases and a patient can define their own phrases as well. Our application aims to generate the commonly used phrases and aims at creating a day to day conversion device for a person with hearing loss. In this project, we propose a solution that can help people with hearing loss to convert the lip movement of the person to text phrases. To achieve this we will train and evaluate our project on the Lip Reading in the wild (LRW) and (Lip Reading Sentences (LRS) datasets [12]. LRW and LRS are audio-visual speech recognition datasets collected from in the wild videos and are provided by the British Broadcasting Channel (BBC) [12]. The LRW dataset consists of about 1000 word utterances and the LRS dataset contains abundant naturally spoken sentences [12]. These datasets will provide us with ample data.

Our approach to develop this solution includes capturing the video of the speaker and then converts their lip movements into readable text. The outcome will be a chance for people with hearing loss to have more involved social conversations. We hope our application helps disabled people overcome isolation they face in their daily lives. From

the limited research that has been done in this area, some promising results have been observed and we would like to take it a step forward through this project.

Acknowledgments

We are deeply indebted to our advisor Professor Gopinath Vinodh for his able guidance from inception to delivering this project. We would like to thank the Department of Software Engineering at San Jose State University for giving us this opportunity to exploit our imaginations and use the knowledge accumulated over the span of this degree to build this project. We feel accomplished after this project as it helped us dive deep into Machine Learning and Neural Networks.

Table of Contents

List of Figures

List of Tables

Chapter 1. Project Overview

1.1 Introduction

The purpose of this project is to convert a person's lip movement into text phrases using a mobile application. While this topic has been researched, the current applications can recognize only a limited number of words. Having a small vocabulary limits the extent of conversation between people. The World Health Organization states that about 5% of the world's population are born with hearing loss [1]. Apart from this, it has been observed that 25% of people over the age of 60 can be affected by hearing loss [1]. In certain cases of serious accidents and illnesses, people can lose their voice temporarily making it difficult to communicate with others.

As there are a lot of cases where voice or hearing loss inhibits communication, it is important to research ways of improving communication. The proposed application might prove beneficial to people with hearing loss or temporary voice loss to communicate with others better. By deciphering the lip movements of the speaker, the application aims to generate text phrases. Using this application, hearing, and voice impaired people can have more involved and meaningful conversations with other people.

In certain instances, people might not know sign language, limiting their ability to have a conversation with hearing-impaired people. In such cases, this application can be used by hearing-impaired people to understand what the other person is saying. People who lose their voice temporarily due to accidents or illnesses find it difficult to communicate with others as they don't know sign language. In this scenario, people can use this application to communicate with others and to convey their symptoms. Apart from these use cases, this application can be used to understand conversations in certain videos where the audio is unclear.

The existing commercial solutions to lip reading are either not easily accessible or are capable of deciphering as little as 20 words. These applications do not facilitate regular conversation due to the mentioned constraints. We plan to train the model to detect at least 100 words which would help in having a more comfortable conversation.

The proposed application, LipScribe, has the potential to break through the barriers of communication between people by lip-reading. It can be used by people born with hearing loss or facing a temporary loss of voice. Furthermore, the application may be used to understand inaudible or unclear videos. LipScribe can revolutionize the way people with hearing or voice loss communicate with other people.

1.2 Proposed Areas of Study and Academic Contribution

Understanding visual speech or lip movement is a dynamic problem. To understand one's lip movement in a scenario where there is no audio provided requires understanding the speech in space and time. For an effective lip movement to text translation system it is necessary to understand how each word sounds. Audio in the real world may be corrupted by background noise. LipScribe, the android application being discussed in this report, aims at developing a platform which converts lip movement to text in real time. The application in the phone will capture a video stream and translate lip movements of the speaker to text. Frames will first be extracted from the video. Then, from each frame the mouth portion is cropped as the Region of Interest (ROI). An approach presented here [1] states OpenCV library can be used to create a localization around mouth area to track lip movements. Creating a ROI reduces the influence of background or incomplete lip image capture on the results. Facial features like the presence of a beard, skin color or moles on face can interfere with the network's predictions. Lab Color space technology can be used to mark the lip region, this is most efficient and is not affected by background noise [1]. Also several other pre-processing techniques like minimizing and maximizing pixels, filtering, and edge preservation for lip contour need to be applied [4]. All this preprocessing is essential as it reduces error rate.

After performing the necessary pre-processing, the lip region can be encoded into a feature vector and given to the three dimensional Convolution Neural Network (CNN) model which predicts the word.

Other approaches have also been researched before but deep learning models have always stood out. For instance, the Hidden Markov Model suffers from convergence and is thus not suited for the task [6]. There are a lot of existing state-of-the-art solutions using CNN to give voice to silent speech but these devices lack the ability to capture a wide vocabulary. Some of the solutions discussed in the section below support limited words/phrases they can recognize in terms of lip movement.

The approach of using two neural networks will enable a deeper understanding of hidden features and the correlation between words and sounds. To create a fully conversational device it is of paramount importance that complete phrases are outputted and using this project aim to achieve that in an efficient manner.

1.3 Current State of the Art

Automated lip reading is gaining popularity fast. The need to devise solutions to facilitate the freedom of expressing oneself without voice has been the most talked-about among researchers.

The scientists and doctors at one such hospital, the “NHS” in the UK have achieved considerable success in this field [7]. SRAVI is an app used by a patient to talk to doctors, nurses, and family members [7]. There is one restriction though, the ability to only recognize a predefined set of 20 phrases [7]. The UK NHS hospital research team working on SRAVI has added the capability to add more phrases to the app as per patients’ needs.

Luvius is another such technology primarily used for speech authentication and in a lot of application areas requiring speaker authentication. The developers behind Luvius have created what they call the LR technology [8]. LR captures the lip movements during speech. The people helming the project have designed a letter recognizing a database created by storing the characteristics of the phoneme [8]. A phoneme is the distinguishing sound of one word from another in a language. For recognizing any spoken word it is matched with the characteristics of the letter in the database. As each person has a different way of pronouncing words Luvius then creates a speech matching this lip movement with the audio of the user.

LipNet was created in a joint effort from Google’s AI DeepMind team and researchers at the University of Oxford[10]. Their trained neural network was 93.2 % more accurate than a human lip reader. A human lip-reader showed an accuracy of 52.3 %[10].

Sony, a leading tech giant, is working on a product called Intelligent Vision Image Sensor for their manufacturing units. They use AI to isolate the words irrespective of the distance between speaker and the sensor. The powerful sensors are not affected by any noise, and do not use voice to make any predictions[11].

Chapter 2. Project Architecture

2.1 Introduction

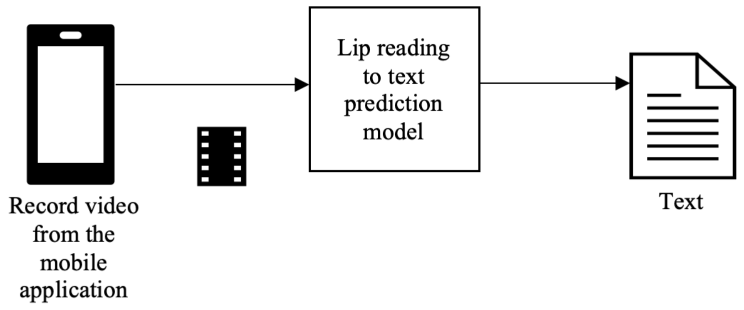

Lip reading to text prediction model
Record video from the mobile application
Text

Figure 2.1.1 Overview of the mobile application

The proposed model is primarily for hearing impaired people to understand what the opposite person is trying to communicate. A mobile application in which the real time video of the speaker is recorded and then text is generated~~.~~ The preprocessing of the video is performed in the android application and then extracted frames are passed as input to the deep learning model. Reading the text should help the people understand the context the speaker is trying to convey and continue to communicate. The video recorded should have the face of the speaker. Video is given to the model implemented for lip reading to predict the text as shown above in the figure 2.1.

LipScribe android application is developed using android studio. In android studio java implementation is adopted with importing required libraries and models. On launching the application, it asks for media and storage permission and allows that it starts recording video. FFMPEG library is used to extract frames from video and on further processing of the video frames deep learning algorithms are used. Android studio supports the external libraries imported as modules. OpenCV library is imported to incorporate the inbuilt face detection algorithm. From the detected face the features of the face are extracted using the facial landmarking algorithm. Using facial landmarks the Region of Interest (ROI) i.e. mouth is extracted. The sequence of images with the ROI are converted into the vector, i.e. embeddings of the video are used for classifying the lip movements. Lip movement classifier is used for classifying different features of the lip.

The features classified are used for generating the text using the text generator model as shown below in figure 2.2.

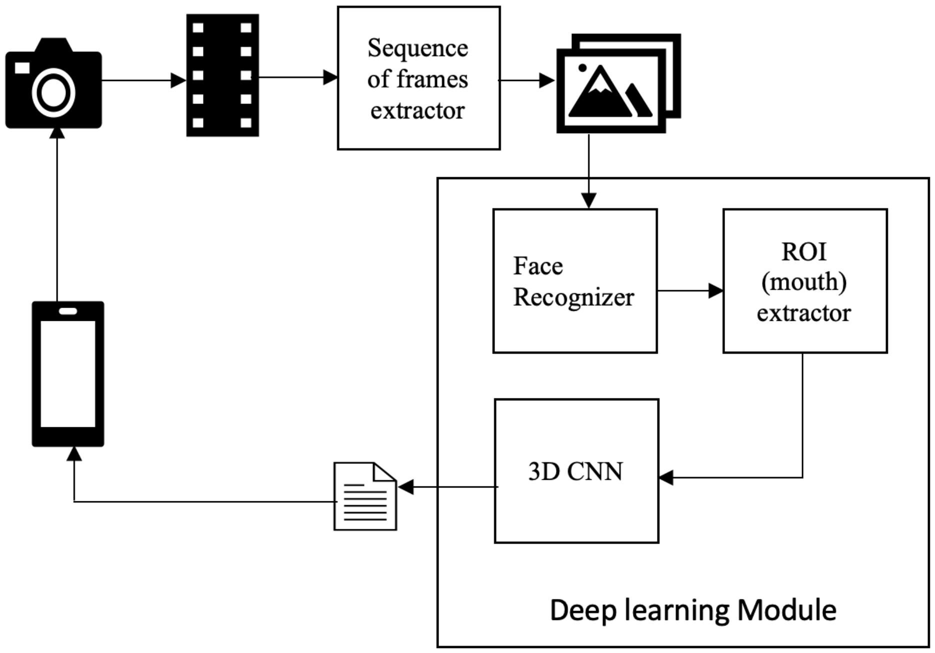


Figure 2.1.2 Architecture of the proposed application

The deep learning model for extracting lip features and generating the transcript of the video, are implemented in tensorflow. The tensorflow models are converted into tensorflow lite models as they are supported in android studio. The model can be imported to the android studio application folder and used to predict the output from them. Models don't have to be stored in the cloud and served to use in the application as the application can have them located in their local folders. However the mobile should have the capability of processing GPU required algorithms.

2.2 Architecture Subsystem

The architecture has subsystems for embeddings generation from the video and prediction of text from the embeddings. From the video recorded the frames are extracted with one second timeframe. From the frames extracted the face is detected using the haar cascade classifier. The mouth region of the face is identified and embeddings are generated. Embedding generation from the video is implemented in java.

The tensorflow model is trained to generate text from the embeddings. Model training is performed using the GPU in the local machine and the h5 extension model is saved and imported into the application for the prediction. The predicted output from the model in application is displayed on the screen.

2.3 Model Selection

A simple 3D convolutional neural network (CNN) was selected to model the application. 3D convolutions are generally used when dealing with video type input data. They work by applying a 3 dimensional filter to the dataset which moves in 3-directions (x, y, z) to calculate the low level feature representations. The 3 dimensions in case of a video are it's Height, Width, and Time. Their output shape is a 3-dimensional volume space such as cube or cuboid.

After each convolution layer a batch normalization, max-pooling 3D, and a dropout were added. Batch normalization acts like a regularization where it normalizes the output of the previous layer to reduce overfitting. Max Pooling 3D layer is specifically used for 3D data (spatial or Spatio-temporal). It downsampled the data. Dropout is used to avoid overfitting by switching off some neurons during training and not making them specialized. Dropout is added. All of these layers constitute the model. In the end, a fully connected dense layer is added to classify each video in one of the categories using Softmax which assigns decimal probabilities for multiclass classification.

Chapter 3. Technology Descriptions

3.1 Android Application

- It is developed using Java 8.
- Chaquopy of version 10.0 is used to invoke the python implementation from the Java implementation of android application.
- The OpenCV library is imported to the android application.
- FFmpeg library is used to access the recorded video from storage and extract frames from the video.

3.2 Deep Learning Model

- Python was used as the base language for the model. Developed a 3D Convolutional Neural Networks(3D CNNs) using Keras of version 2.2.4-tf and Tensorflow of version 2.1.0 as it has to support the packages of chaquoPy in android application. HaarCascade classifiers from OpenCV were used to detect face and mouth regions in a video frame. Video frames were extracted using the VideoCapture function from OpenCV library.

Chapter 4. Project Design

4.1. Architecture Design

LipScribe android application has two major entities: application and models. As shown in the below figure 4.1 each entity has the activities to perform. The user has to launch the mobile application to start the camera. Application invokes the camera to start processing the video. The processing of video with machine learning models is performed by machine learning models. However the models are at the application level. The generated text from the frames of the video are displayed on the screen parallely.

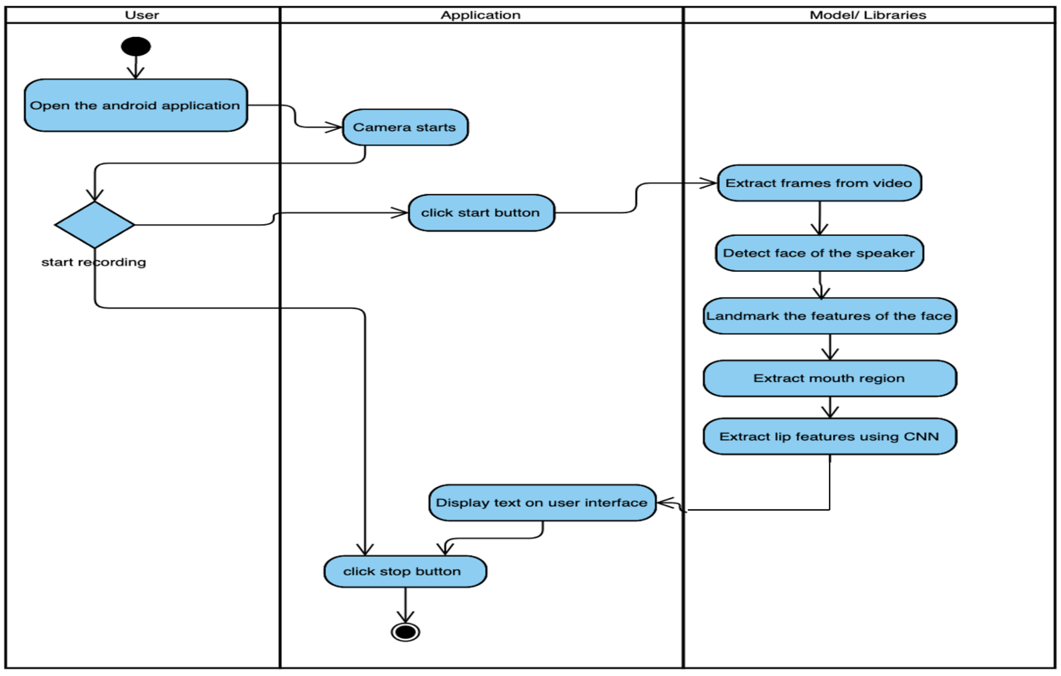

User
Application
Model/ Libraries
Open the android application
Camera starts
start recording
click start button
Extract frames from video
Detect face of the speaker
Landmark the features of the face
Extract mouth region
Extract lip features using CNN
Display text on user interface
click stop button

Fig 4.1.1 Activity diagram of LipScribe android application

4.2. Project Workflow

LipScribe android application helps in generating the readable text from the video of the speaker. The mobile application can record the video having the speaker in the frame. The real time video is recorded and the text is generated, which helps the impaired people to understand what the speaker is trying to express. LipScribe mobile application is developed in android studio which can import the external libraries and also the deep learning models as tensorflow lite models. Although the camera starts on launching the mobile application, the video is recorded and processed after clicking the start button and can be stopped with the stop button. The OpenCV library for detecting faces is imported as a module to the android studio. The 3D CNN(Convolutional neural network) is used to extract features using extracted regions of interest in the form of frames. The SoftMax layer at the end of the CNN is used to get the probability of each class to get the final result. Below figure 6.1. shows the classes for developing the mobile application.

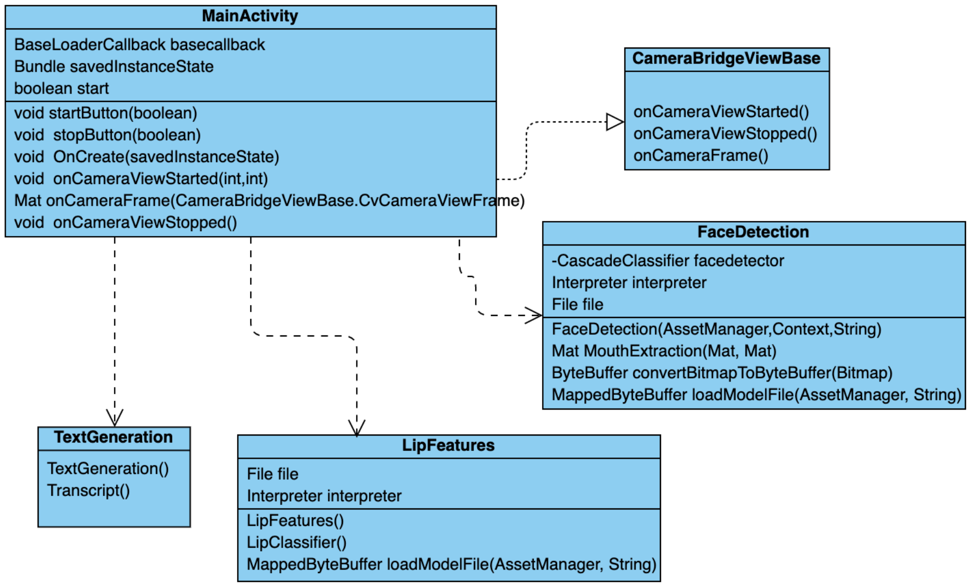

MainActivity
BaseLoaderCallback basecallback
Bundle savedInstanceState
boolean start
void startButton(boolean)
void stopButton(boolean)
void OnCreate(savedInstanceState)
void onCameraViewStarted(int,int)
Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame)
void onCameraViewStopped()
CameraBridgeViewBase
onCameraViewStarted()
onCameraViewStopped()
onCameraFrame()
FaceDetection
-CascadeClassifier facedetector
Interpreter interpreter
File file
FaceDetection(AssetManager,Context,String)
Mat MouthExtraction(Mat, Mat)
ByteBuffer convertBitmapToByteBuffer(Bitmap)
MappedByteBuffer loadModelFile(AssetManager, String)
TextGeneration
TextGeneration()
Transcript()
LipFeatures
File file
Interpreter interpreter
LipFeatures()
LipClassifier()
MappedByteBuffer loadModelFile(AssetManager, String)

Fig 4.2.1 Class diagram of the LipScribe application

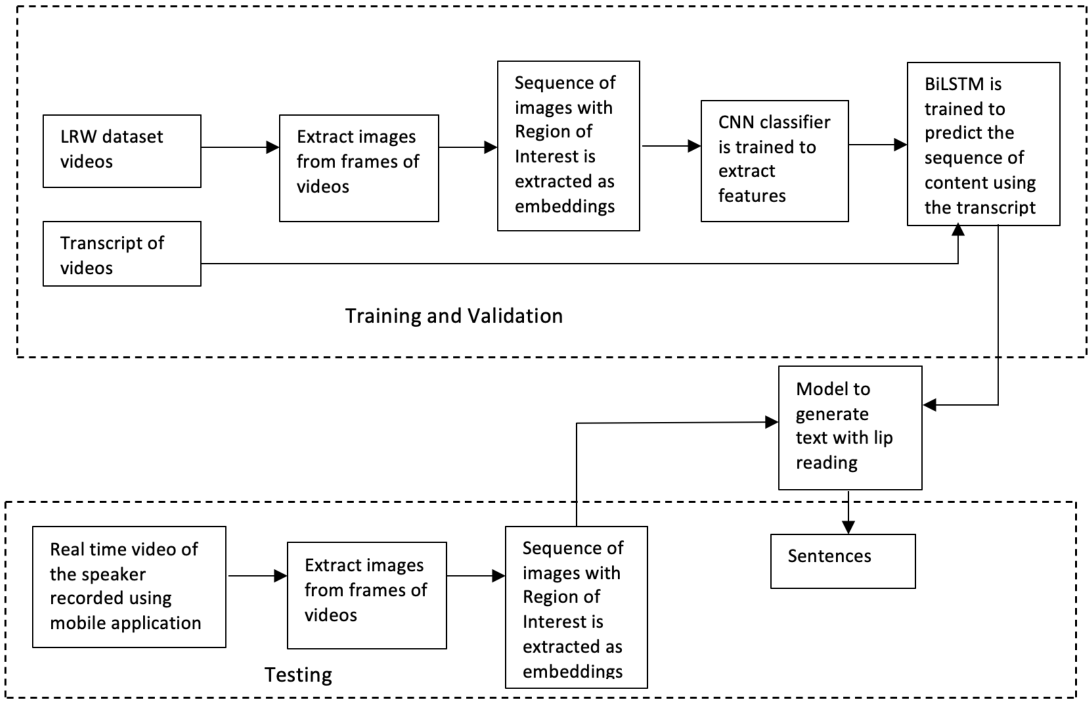

LRW dataset videos
Extract images from frames of videos
Sequence of images with Region of Interest is extracted as embeddings
CNN classifier is trained to extract features
BiLSTM is trained to predict the sequence of content using the transcript
Transcript of videos
Training and Validation
Model to generate text with lip reading
Real time video of the speaker recorded using mobile application
Extract images from frames of videos
Sequence of images with Region of Interest is extracted as embeddings
Sentences
Testing

Figure 4.2.2 Data flow diagram of LipScribe android application

4.3. Data Preprocessing and Model Training

The deep learning model used in the LipScribe application is trained on the LRW dataset which has videos of people speaking various words. Below are the steps to be performed in the training and testing phase.

Dataset

1. We leverage LRW (Lip Reading in the Wild) dataset [12] to train the model.
2. This dataset contains over 1000 utterances of 500 different words, spoken by hundreds of different speakers [12].
3. All videos are 29 frames (1.16) seconds in length, and the word occurs in the middle of the video[12].
4. To train the dataset, we used about 150 different words from the dataset. The list of words we used to train the model can be found here.

Training

1. The frames of the video are extracted using one second time frame using OpenCV's video reading library function.

2. OpenCV library's haar cascade face detection model is used to detect the faces in the frame and then subsequently haar cascade mouth detection is used to get the lip region from that face extracted.
3. For extracting the region of interest i.e mouth the facial landmarks are used.
4. The mouth image is then converted to an embedding which is nothing but a vector to feed to the CNN.
5. The 3D CNN classifier is trained on word embedding created in step 4 and the list of words to make the model learn to decipher lip movement to word.

Testing

1. Real time video of the speaker is recorded using the mobile application.
2. As part of the preprocessing of the frames of the video the region of interest is extracted and given to the trained model.
3. The model gives the transcript of the recorded video

Chapter 5. Project Implementation

5.1 Android Application Design

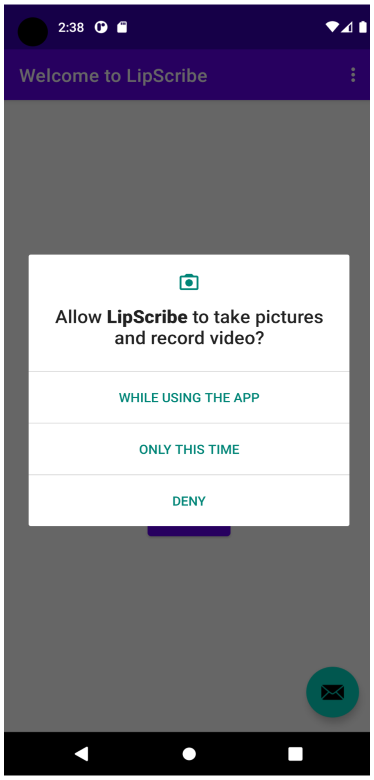


Fig 5.1.1(a)

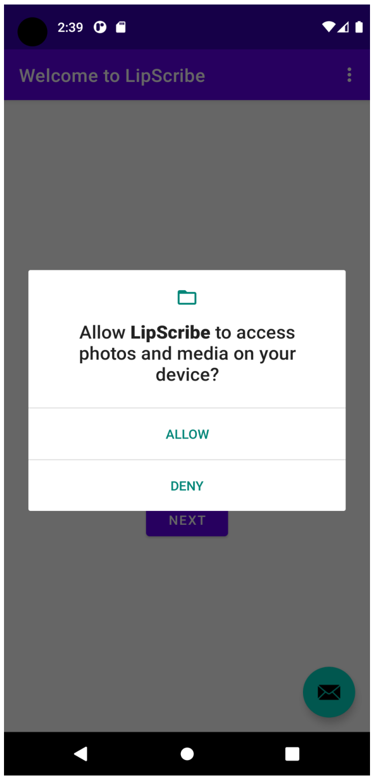

2:39
Welcome to LipScribe
Allow **LipScribe** to access photos and media on your device?
ALLOW
DENY
NEXT

Fig 5.1.1(b)

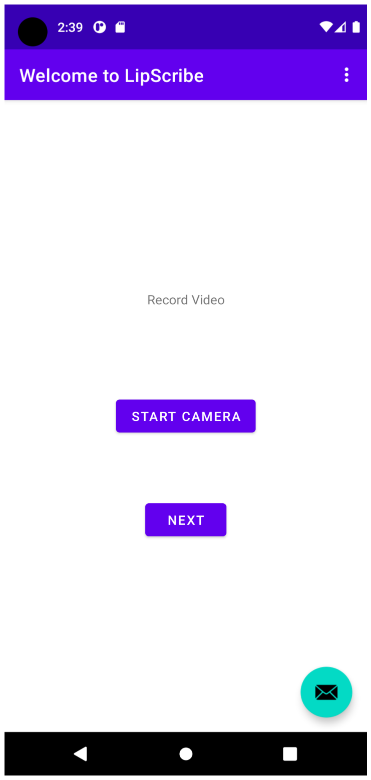


Fig 5.1.1(c)

Fig 5.1.1 screenshots of android application launch page

Fig 5.1.1(a) is the screenshot where the application requests for the permission to take pictures and videos. Fig 5.1.1(b) is the screenshot where the application requests permission to access the media and photos in the storage. Fig 5.1.1(c) is the screenshot of the launch page of android application.

- Created an android application which has a button on the first screen to start recording the video. First it checks if the device has a camera and if the camera has permission to record the video. If the permission is not enabled it requests for the permission to record videos and capture photos. It also checks for the external storage permissions. If not enabled it requests for read and write permissions.

- Application starts the camera and captures the video using the inbuilt camera of the android device. The recorded video is stored at the external storage.

- On completion of storing the video the python code is invoked using the chaquoPy library. The URI of the video is passed to the python code to access it and preprocess the video.

- Recorded video is preprocessed to convert in the input format for the deep learning model. FFMPEG library is used to convert recorded video into frames. Used Haar Cascade classifier to detect face and extract region of interest (lip region). If no frontal face is detected in the recorded video then it will return null and show "No face detected" in the UI of the application.

- Used Async task to add loading screen into the application which will be shown when tensorflow model is making prediction on extracted frames. Prediction task is the background task and loading screen is the main task for a short duration.

- Tensorflow model of predicting words from the video embeddings is put into the structure of the android application. The model predicts the word with the video embeddings generated from the recorded video.

- Tensorflow model returns text which will be displayed on the screen of android application.

5.2 Model Evaluation

- To evaluate the model, we split the LRW dataset into train, test and validation splits.
- The training accuracy was about 0.75, while the validation accuracy was 0.70.
- We tested the trained model with the test dataset and observed an accuracy of 0.60.
- We observed that the validation loss decreased with each epoch while the training and validation accuracies improve.

To evaluate the model, LRW dataset has been split into train, test and validation splits. The training accuracy was about 0.75, while the validation accuracy was 0.70. We have tested the trained model with the test dataset and observed an accuracy of 0.60. We observed that the validation loss decreased with each epoch while the training and validation accuracies improved.

Chapter 6. Testing and Verification

6.1. Model Testing

In the graphs below we depict the loss and accuracy in each epoch for training and validation sets. We can see the loss went down and the accuracy improved over epochs.

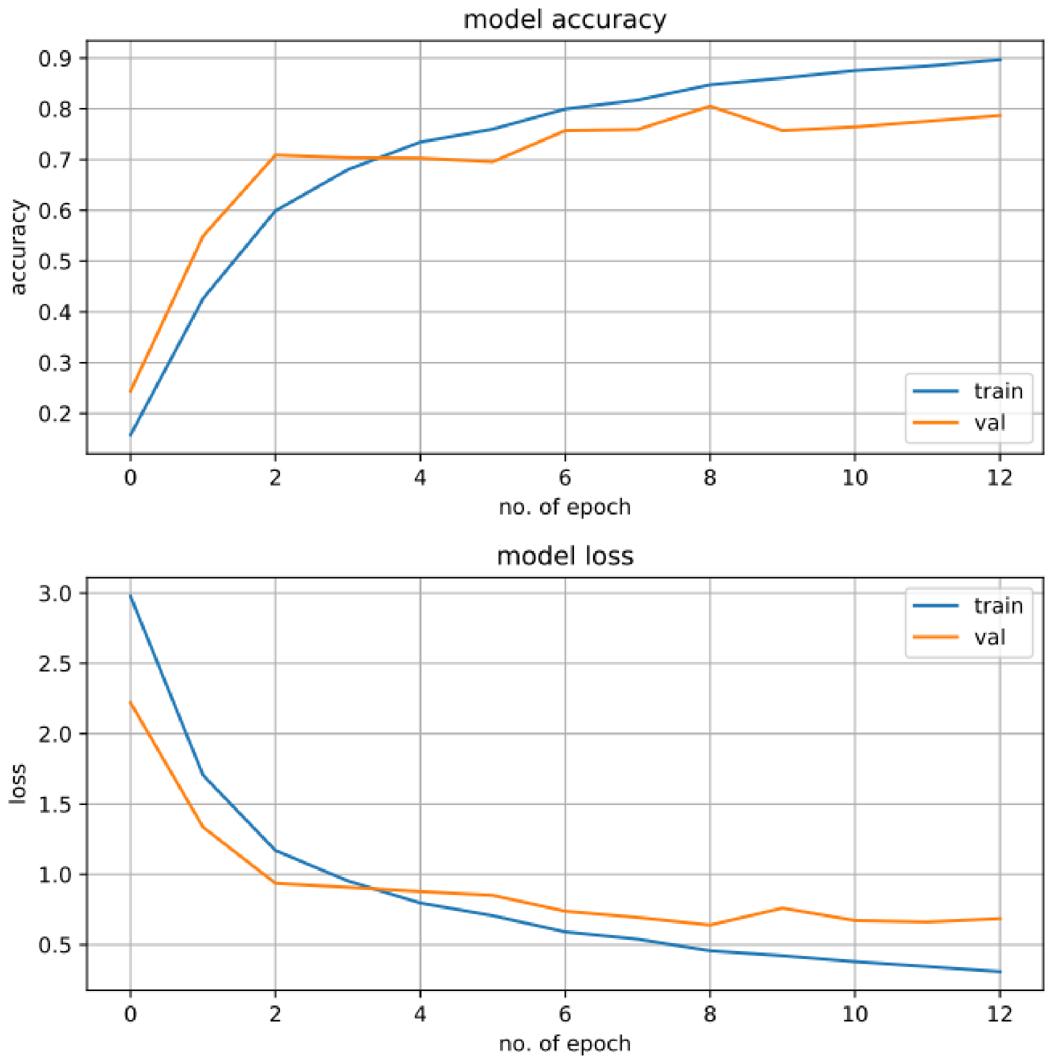
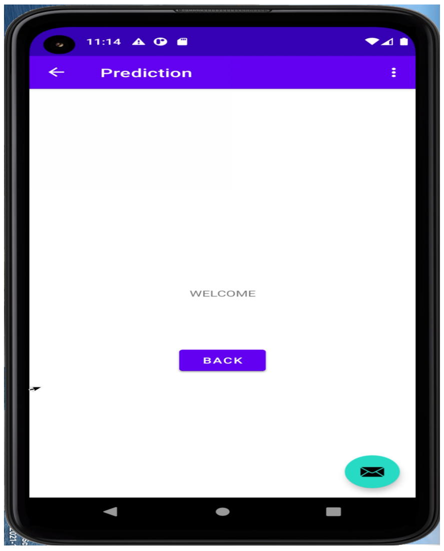


Figure 6.1.1 Model accuracy and loss curves

6.2. Application Testing

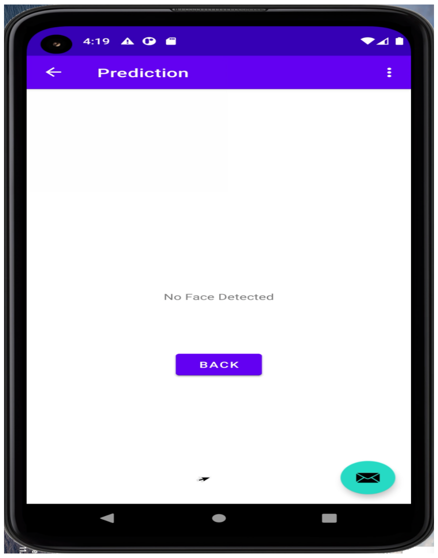
Test	Expected Result	Actual Result
3D CNN based face model should find a face and mouth region in the video and output lpl[;word being spoken.	1. If a face is found in the video, a .jpeg file of the mouth region saved in the image frames output folder 2. Incase of an undetected face in the video, raise an error	1. A .jpeg file of the mouth region was saved in the output image folder which was used to create embeddings 2. When a video was passed without the face, the model will output a message with no face detected.

Table 6.2.1 Results of Application Testing



WELCOME

Fig 6.2.1(a)



No Face Detected

Fig 6.2.1(b)

Fig 6.2.1(a) is the screenshot of the android application screen where the output of the model prediction is displayed on screen. Fig 6.2.1(b) is the screenshot of the android application screen when no face is detected in the video recorded.

Chapter 7. Performance and Benchmarks

7.1 Evaluation

To evaluate our model, we split the LRW dataset into train, test and validation datasets.

Dataset	**Train Accuracy**	**Test Accuracy**
Lip Reading in the Wild	0.75	0.60
GRID corpus (POC) contains 10 words	0.90	0.75

Table 7.1.1. Evaluation Metrics

Confusion Matrix
A confusion matrix i.e a table which is often used to describe the performance of a classification model on test data for which the true values are known. This evaluation method was used here on a smaller set of data to evaluate model performance on test data. As in Fig 7.1 we can see that in each case a good number of positives have been predicted and the false positives are less.

Fig 7.2 presents logs from Tensorboard visualization of the model training. We can see loss decreasing and accuracy increasing over epochs.

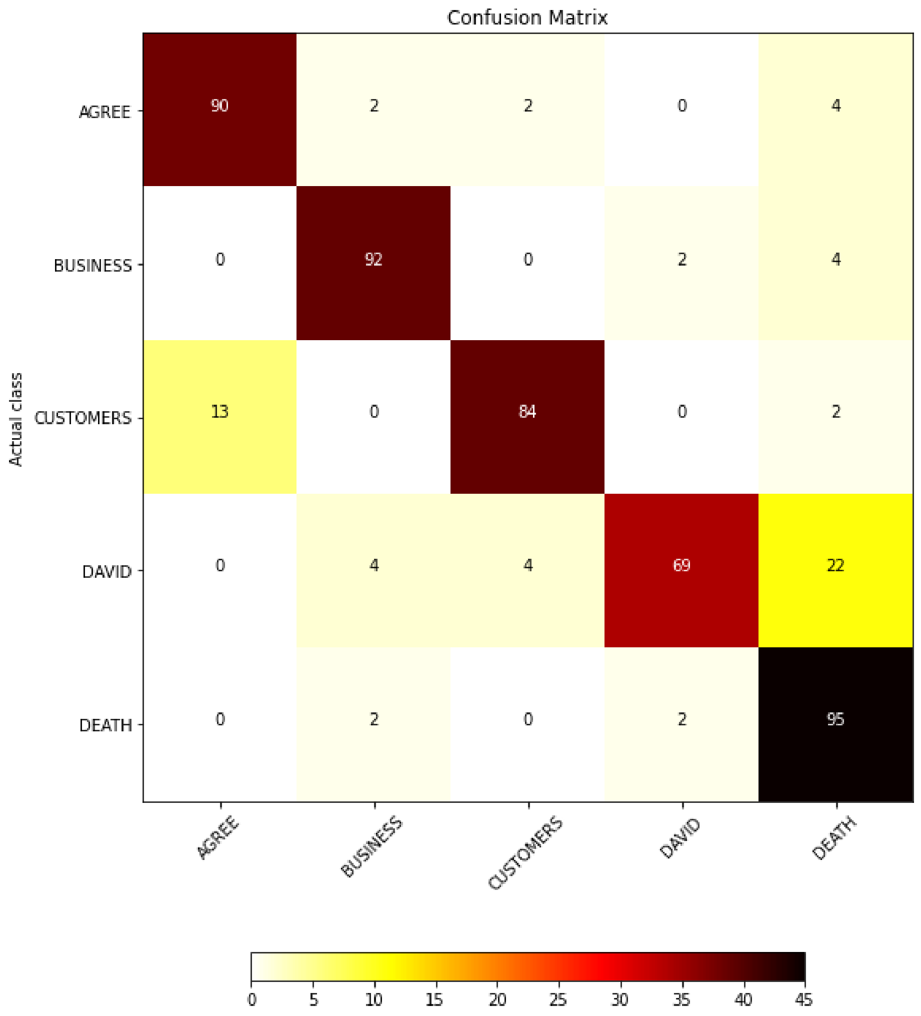


Figure 7.1.1 Confusion Matrix on Test Data after training

Tensorboard Logs

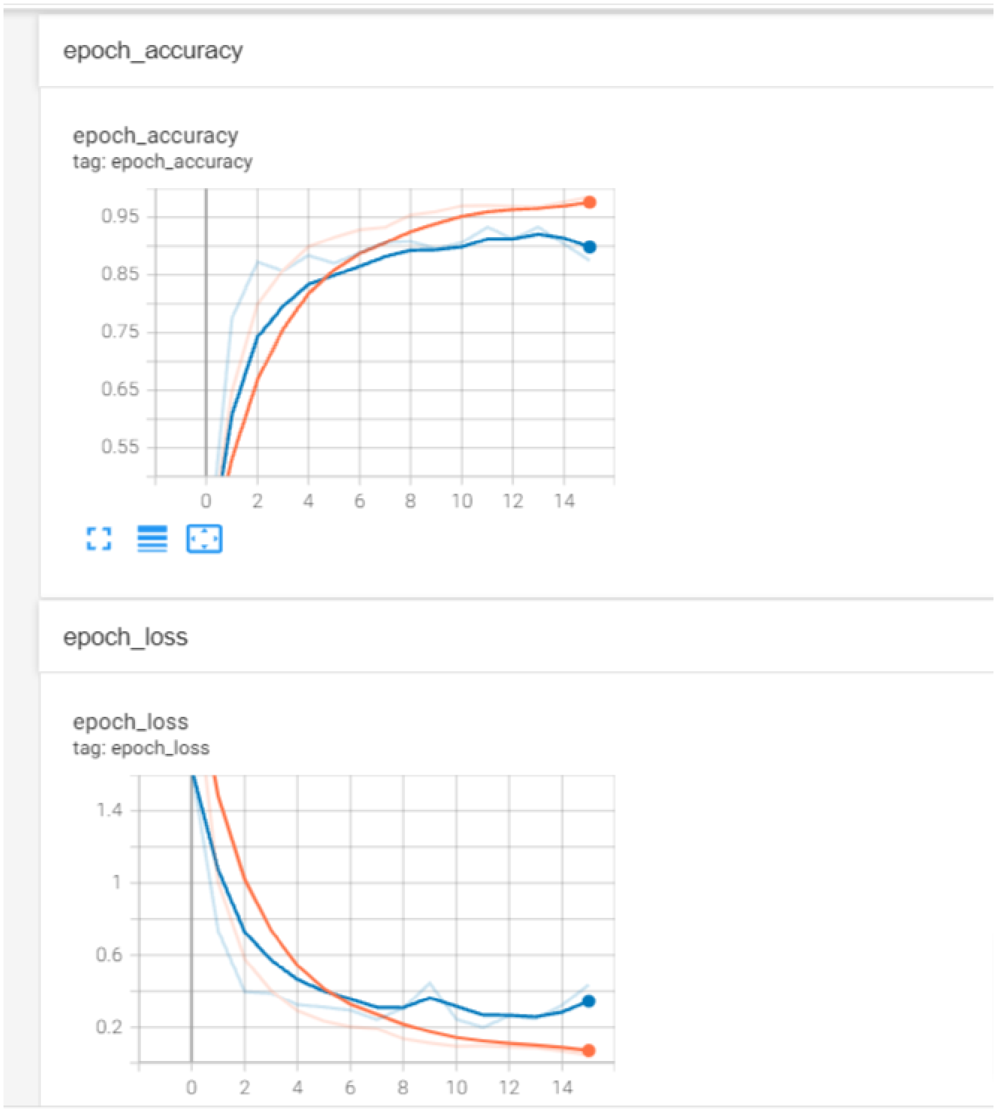


Figure 7.1.2 Tensorboard logs during training for Accuracy and Loss

Chapter 8. Deployment, Operations, Maintenance

8.1. Deployment

- A deep learning model to predict words will be exported as a tensorflow model. This tensorflow model will be coupled with the android app project.
- Add model.h5 and model.json both the files in the python folder of the application.
- Click on the Build APK option of Build to generate a new APK.
- Install APK in android device.

8.2. Operations

- Android device with the requirement of API 21: Android 5.0 (Lollipop).
- Android device with OS supporting GPU operations.
- Android devices should have a compatible camera for recording the video.
- The frame of video recording should have the face of a human from forehead to chin (Preferably one face for better accuracy).
- However the audio is recorded along with video, it is not considered for the prediction.

8.3. Maintenance

- Retraining the deep learning model with newly added input data.
- Export the model in tensorflow format and then update it in the android app project.
- Publish the updated version to the Google play store.

Chapter 9. Summary, Conclusions, and Recommendations

9.1. Summary

Lip reading to text generation is a task which has been researched very little. We trained a model which takes a video as input and generates words by deciphering the lip movements of the user. To train the model, we leveraged the LRW dataset from BBC [1] and embedded the model to a mobile application using Android studio to get real-time inferences. Our model is trained on over 150 commonly spoken words and can generate upto one word per second.

9.2. Conclusions

We developed a mobile application with an embedded model which translates lip movements of a person to words. The 3D CNN based model was trained with around 150 words which can be used in day-to-day conversations. The trained model reached about 80% accuracy on the test dataset. Using LipScribe, people who are hard of hearing can have comfortable conversations with others around them.

9.3. Recommendations for Further Research

For future work, the model can be trained on a larger corpus of words for a larger coverage of words. A sequence generation model can be added to the output of CNN. This can help form phrases from words outputted. As the model is stored on an edge device, a model optimization tool-kit could be used to optimize the model to reduce inference latency. It would be great to support an iOS-based application along with an android-based application.

Glossary

LRW : Lip reading in the wild
CNN : Convolutional Neural Network
ROI : Region of interest
AI : Artificial Intelligence
Chaquopy : Library used to integrate python code in android application.
FFmpeg : Open source project consisting of libraries to handles videos, audios and multimedia files

References

[1] Yao WenJuan, Liang YaLing and Du MingHui, "A real-time lip localization and tacking for lip reading," 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), 2010, pp. V6-363-V6-366, doi: 10.1109/ICACTE.2010.5579830.

[2] P. Sindhura, S. J. Preethi and K. B. Niranjana, "Convolutional Neural Networks for Predicting Words: A Lip-Reading System," 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2018, pp. 929-933, doi: 10.1109/ICEECCOT43722.2018.9001505.

[3] J. S. Chung, A. Senior, O. Vinyals and A. Zisserman, "Lip Reading Sentences in the Wild," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3444-3453, doi: 10.1109/CVPR.2017.367.

[4] B. Lin, Y. Yao, C. Liu, C. Lien and B. Lin, "Development of Novel Lip-Reading Recognition Algorithm," in IEEE Access, vol. 5, pp. 794-801, 2017, doi: 10.1109/ACCESS.2017.2649838.

[5] M. A. Abrar, A. N. M. N. Islam, M. M. Hassan, M. T. Islam, C. Shahnaz and S. A. Fattah, "Deep Lip Reading-A Deep Learning Based Lip-Reading Software for the Hearing Impaired," 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129), 2019, pp. 40-44, doi: 10.1109/R10-HTC47129.2019.9042439.

[6] P. Sujatha and M. R. Krishnan, "Lip feature extraction for visual speech recognition using Hidden Markov Model," 2012 International Conference on Computing, Communication and Applications, 2012, pp. 1-5, doi: 10.1109/ICCCA.2012.6179154.

[7] "SRAVI - Speech Recognition App for the Voice Impaired." https://www.sravi.ai/. Accessed 7 Jul. 2021.

[8] "Luvius Lip Reading – Patented speech-to-text innovation." http://luvius.nl/. Accessed 7 Jul. 2021

[9] "LipNet is the most accurate lip-reading software ever ... - Mashable." 9 Nov. 2016, https://mashable.com/article/lipnet-lipreading-software. Accessed 7 Jul. 2021.

[10] “LipNet - End-to-End Sentence-level Lipreading.” https://www.researchsnipers.com/wp-content/uploads/2021/05/Lipnet.pdf

[11] "Sony's New Lip-Reading Technology Could Boost ... - PCMag." 13 Jan. 2021, https://www.pcmag.com/news/sonys-new-lip-reading-technology-could-boost-accessibility-or-invade-privacy. Accessed 7 Jul. 2021

[12] LRW, *Lip Reading in the Wild.* Retrieved from https://www.robots.ox.ac.uk/~vgg/data/lip_reading/lrw1.html

Appendices

Appendix A.