

INTERDISCIPLINARY PROJECT REPORT
at

**Sathyabama Institute of Science and
Technology (Deemed to be University)**

Submitted in partial fulfillment of the requirements for the
award of Bachelor of Engineering Degree in Computer
Science and Engineering

By

Nehal Ahmed A
(Reg.No. 40731069)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY JEPPIAAR NAGAR, RAJIV GANDHI
SALAI,
CHENNAI – 600119, TAMILNADU**

APRIL 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **NEHAL AHMED A (40731069)** who carried out the project entitled "**LIP READING RECOGNITION**" under my supervision from FEB 2023 to APRIL 2023.

Internal Guide

Ms. B.BalasaiGayathri M.E.,

Head of the Department

Dr.S.Vigneshwari,M.E.,Ph.D.

Submitted for Viva voce Examination held on 17-04-2023

Internal Examiner

External Examiner

DECLARATION

I, **NEHAL AHMED A** hereby declare that the project report entitled “**LIP READING RECOGNITION**” done by me under the guidance of **Ms.B.BalasaiGayathri M.E.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE: 17-04-2023



PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing, Dr.S.Vigneshwari,M.E.,Ph.D. and Dr. L. Lakshmanan M.E.,Ph.D., Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. B.BalasaiGayathri M.E.** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Machine Learning techniques give computers the capability to train and learn by using sample inputs and provide outputs which leads a model to test the test cases instead of being programmed. Lip reading recognition is a task of converting speech (lip movements) to readable format like text without audio. Lip reading recognition is a technology that helps for communication using devices. The uses of this concept are immense for the people who are who have hearing deficiency and vocally challenged. In situations where there is lot of noise, its very difficult to listen to each other. The noise around us can sometimes cause a huge impact on the conversation which may lead to miscommunication, the example can be a phone call in a theatre. In future, this problem can be solved by Lip Reading Recognition Technology. This technology helps one to transmit information without using vocal cords. This technology uses Lip Tracking which is a biometric system, it works based on a genuine system that can be developed using different levels of video processing, and hence it's possible to get lip contour and the exact location of key points in the subsequent frames which is usually referred as lip tracking. This technology works on the lip movement and generates frames where the other end person can easily get the result of it.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	(i)
	LIST OF FIGURES	(ii)
1	INTRODUCTION	
	1.1 LIP READING RECOGNITION	11
	1.2 LITERATURE SURVEY	12
	1.3 PROPOSED SYSTEM	16
2	AIM AND SCOPE OF THE PRESENT INVESTIGATION	
	2.1 AIM	18
	2.2 SCOPE	18
	EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED	
	3.1 DATASETS AND FEATURES	19
	3.2 WORKING OF LIP-READING RECOGNITION WITH DIFFERENT MODELS	21
		6

3.2.1 DYNAMIC TIME WARPING	21
3.2.2 SHAPE TEMPLATE	22
3.2.3 SNAKE'S APPROACH	23
3.2.4 HIDDEN MARKOV MODEL	24
3.2.5 CONVOLUTIONAL NEURAL NETWORKS MODEL	26
3.3 METHODS	
3.3.1 CNN + LSTM BASELINE	28
3.3.2 DEEP LAYERED CNN + LSTM	30
3.3.3 IMAGE-NET PRETRAINED VGG-16 FEATURES + LSTM	31
3.3.4 FINE-TUNED VGG-16 + LSTM	32
RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	
4.1 RESULTS AND DISCUSSION	33

4.2 CNN + LSTM BASELINE	34
4.3 DEEP LAYERED CNN + LSTM	34
4.4 IMAGE-NET PRETRAINED VGG-16 FEATURES + LSTM	35
4.5 FINE-TUNED VGG-16 + LSTM	36
4.6 UNSEEN DATA	37
 SUMMARY AND CONCLUSIONS	
 REFERENCES	 41
 SCREENSHOTS	 43
 SOURCE CODE	 43

LIST OF FIGURES

Fig no	Fig title	Page no
1.1	System Architecture	17
3.1.1	Original Input image	20
3.1.2	Horizontally flipped image	20
3.1.3	Example full input sequence of length 5	21
3.1.4	Example full input sequence of length 10	21
3.2.1	Dynamic time warping	22
3.2.2	Shape Templates used as inputs to Lip reading	23
3.2.3	Lip localization using Snake's approach on input image	24
3.2.4	Speech recognition using hidden Markov model	25
3.2.5	Lip-Reading recognition using CNN	27
3.3.1	CNN + LSTM Baseline model layer architecture diagram	28
3.3.2	Structural diagram courtesy of CS231N	29
3.3.3	Deep Layered CNN + LSTM model layer architecture diagram	30
3.3.4	VGG-16 Structure	31
4.1.1	Seen Subject Accuracy	33
4.1.2	Seen Subject Accuracy Comparison Graph	33

4.3.1	Loss and Accuracy Plots for Deep-Layered CNN + LSTM	34
4.4.1	Loss and Accuracy plots for ImageNet Pre-trained VGG-16 Features + LSTM	36
4.5.1	Confusion Matrix for VGG-16 + LSTM	37
4.6.1	Confusion Matrix for VGG + LSTM	38
4.6.2	ImageNet Pre-trained VGG-16 features with accuracy of 0.14	39
4.6.3	ImageNet Pre-trained VGG-16 features with accuracy of 0.14	39

CHAPTER-1

INTRODUCTION

1.1 LIP READING RECOGNITION

Lip Reading Recognition technology is mainly used for speech tracking by means converting the speech into understanding format by externally read the developments of the lips, face, and tongue. In lively situations, where the acknowledgment of any discussion is troublesome, which can be visual discourse acknowledgment that gives a successful way to get discourse to get better results. Lip reading can be difficult sometimes because of the different types' accents, speed of talking, facial highlights, skin shading, and so forth. However, there are a large group of utilizations, because of which this issue expects noteworthiness. It is extremely useful to remove the communication gap especially when it is communicated in a noisy place, and so forth. Lipreading is a difficult task for anybody without the knowledge, especially without any context of the concept.

Most lip-reading actions, besides the lips may sometimes make use of tongue and teeth to pronounce a word or letter which are latent and difficult to disambiguate without context. Lip reading is a complex issue because of different kind of lip shapes & color. Lip reading is similar to guesswork and will not be accurate because of pronunciation differences, bad position of faces, having hands over mouth, moustaches, too fast speech, low lighting, and beards etc.

The lip-reading recognition has been documented since long time, which includes the people with hearing deficiency use lip-reading method where it can be an adjustment to get and understand the fluent speech of the speaker. When it comes to converting these kinds of issues into automation there can be many difficult challenges comparing the ordinary recognition which can be audio. These audio speeches which has the pre-defined units are known as phonemes, mapping between the phonemes and the words are given by the

pronunciation dictionaries. And it's easy to map video than audio as a result because of the compressed video rates are higher than uncompressed audio. So, it's a easy and efficient way to get the better results using video than the audio formats.

This paper investigates the task of speech recognition from video without audio. We present several neural network models with varying successes in this classification task. The input data to our algorithm is sequences of still images taken from frames of video footage. We use different neural network models to output one of 10 words that are spoken (or mouthed) by a face in the input images. We explore and combine a number of different models including CNNs, RNNs, and existing publically available pre-trained networks to assist in mouth recognition. This is an interesting learning task given that video traffic is growing at a high rate throughout the web, and this model could help extract data and process it to gain insights into the action or topics occurring in a video. Applications of a visual audio classifier range from play prediction in sporting events to profanity detection on social media sites to a live action lip reading mobile application.

In the past, research efforts have been far more focused on gesture recognition rather than visual speech recognition, making this for a new and exciting field to explore. There are a few existing systems and applications for lip reading, although most do not use neural networks but instead other machine learning techniques. More advanced visual speech recognition models such as Google's DeepMind LipNet network were published only a few months ago.

1.2 LITERATURE SURVEY

A neural network-based lip reading system is suggested in this study. The system lacks a language and relies only on visual clues. With only a few number of visemes to recognize as classes, the system is designed to lip read sentences with a wide variety of vocabulary and recognize words that may not

have been included in system training. The system was put through its paces on the difficult BBC Lip Reading Sentences 2 (LRS2) benchmark dataset.

Speech identification from visual-only recordings of a speaker's face is possible using a processing pipeline based only on neural networks, producing much higher accuracy than traditional approaches. Feed forward and recurrent neural network layers (specifically, Long Short-Term Memory (LSTM)) are layered to form a single structure that is trained by back-propagating error gradients across all layers. In this study, the performance of a stacked network was experimentally assessed and compared to that of a typical Support Vector Machine classifier using common computer vision characteristics.

They cover the key strategies for audio-visual speech recognition that have been developed during the last two decades in this chapter. They start with the visual feature extraction challenge and then go on to audio-visual fusion. In both cases, they describe some of the strategies used during the Johns Hopkins summer 2000 workshop (Neti et al., 2000). They also explore the issue of audio-visual speaker adaptation, which is critical for establishing systems across databases or building speaker-specific models. They next go over the primary audio-visual corpora that have been utilized in the literature for ASR trials, such as the IBM audio-visual LVCSR database. Following that, they report experimental findings on automated speech reading and audio-visual ASR. They address the challenge of automated recognition of damaged speech as an application of speaker adaptation. Finally, they wrap off the chapter with a review of the present status of audio-visual ASR and what they see as unresolved issues in this field.

This research demonstrates the risk of not employing a variety of speakers in the training and test sets. They provide categorization findings from AVletters 2, a high-definition version of the well-known AVletters database. They show that by carefully selecting features, it is feasible to achieve visual-only lip-

reading performance that is very similar to audio-only recognition for single and multi-speaker configurations.

They studied the performance of a machine-based lipreading system with shape-only parameters as well as comprehensive shape and appearance parameters. They also compared the performance of a machine-based lipreading system to that of a human lip-reader. The automatic method surpasses human lip-readers, they discovered. Surprisingly, when adding full appearance data to the machine-based approach, there is minimal gain in identification accuracy for relatively basic jobs, but human lip-readers show large improvements in performance. Finally, they tested the effect of 'speaker training' on human lip-reading skill and discovered that even very modest training is enough to increase performance.

This project's purpose was to recognize words and sentences uttered by a talking visage, with or without audio. In contrast to prior efforts that concentrated on identifying a restricted number of words or phrases, they approach lip reading as an open-world problem — unrestrained natural language sentences and in-the-wild films.

The creation and recording of a new Czech audio-visual speech database intended for studies on various impaired situations are presented in this study. They focused on adjusting the lighting. As a result, the audio-visual speech database UWB-07-ICAVR was created, which is a great resource for testing algorithms for visual speech parameterization in different lighting.

They suggested a novel visual speech recognition system that makes use of both 2D pictures and depth maps obtained with RGB-D cameras. Their method is based on 3D rigid tracking of the speaker's face to extract image and depth thumbnails reliably. These are then utilized to generate motion and appearance descriptors, which are subsequently fed into an SVM classifier. They created a new dataset (MIRACL-VC) and utilized two other available datasets to evaluate

their approach (CUAVE and OuluVS). The assessment findings acquired on these three datasets reveal that their system is very competitive with other recent research in the literature. They also demonstrated that depth data improves the LR system's performance considerably, and that the combination of appearance and motion descriptors improves the SI performance.

In this study, they offer a method for distinguishing a target speaker's speech signal from background noise and other speakers by exploiting visual information from the target speaker's lips. They proved that the deep network can synthesize comprehensible speech from highly noisy audio segments captured in unconstrained 'in the wild' contexts by predicting both the phase and amplitude of the target signal.

They explored how viewers identify languages from visual-only presentations of speech in this paper. Their initial objective was to reproduce and enhance the findings of SotoFaraco et al. (2007) utilizing a different experimental methodology and collection of languages. Overall, the results of Experiment 1 validated previous findings that language identification may be accomplished only through visual cues. Although they were unable to reproduce the impact of linguistic experience (monolingual vs. bilingual) on sensitivity, they did discover an effect on response bias; bilingual individuals displayed a larger bias toward their native language than monolingual speakers. This response bias was greatest among bilingual English speakers, but a similar pattern was observed among Spanish bilinguals, with a greater bias toward Spanish.

They provided a completely automated technique to distinguishing native from non-native speech in English based only on visual cues in this study. Overall, the purpose of this work is to present a fundamental study of visual discrimination between native and nonnative speech, therefore establishing a research topic that can be immensely valuable in biometric applications. They show that important information for distinguishing between native and non-native speech is present in the visual stream, which is likely to boost

performance when paired with audio-only approaches, particularly in loud conditions.

1.3 PROPOSED SYSTEM

One of the most extensively investigated areas in affective computing and human–computer interaction is automatic speech recognition based on lip reading. In this research, we will offer a solution for automated word identification and recognition based on diverse lip motions. We will demonstrate a method for detecting and recognizing human lip expressions. The suggested system's goal is to provide a technique that automatically recognizes and classifies diverse human lip expressions. Automatic speech recognition (ASR) relies heavily on visual speech information, especially when audio is distorted or unavailable. Despite the success of audio-based ASR, visual speech decoding remains a major challenge. Our project's purpose is to recognize words and sentences uttered by a talking visage, with or without audio.

The two-stream model is further extended with the state-of-the art approach for recognition of isolated words. This method consists of 3D convolutional layer that is followed by softmax layer, Bidirectional Gated Recurrent Unit and 18-layer Residual Network. The former method is more relevant to tasks like isolated word detection, recognition and classification, also the latter to Sentence level differentiation and large vocabulary continuous speech recognition. however, recent publishes in speech recognition and natural language processing declare that direct modeling of words is efficient even for and large vocabulary continuous speech recognition.

This research focuses on the advancement of lip recognition, which will greatly outperform previously presented systems. A video of a human's lip expression

is obtained. A video to frames converter is used to convert the video to frames. As the input image, the average frame is used.

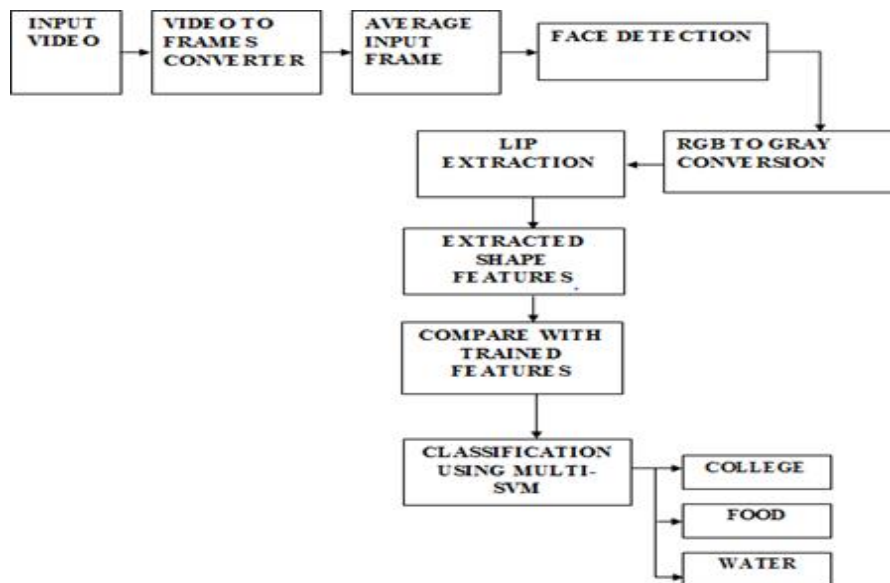


Fig -1.1: System Architecture

The captured photos include a human's face and a backdrop image. As a result, the captured pictures are initially subjected to pre-processing techniques such as Region Of Interest (ROI) segmentation to eliminate the background. The method of finding the area of a face in a picture is known as face detection. Following the face detection, the face is retrieved using the bounding box approach. The result of the face detection and extraction is an RGB color face image. Using the RGB to grey scale conversion procedure, the RGB face picture is transformed to grey scale. The Lip is the real ROI component. Initially, the face was regarded as the ROI component. The Lip is then recognized and retrieved from the Face using orientation (region) estimation. The lip picture is used to extract the form characteristics. Using the retrieved characteristics, we will categories the various human voice outputs using the MultiSVM method. As a result, the human's unique lip movement were identified.

CHAPTER-2

AIM AND SCOPE OF PRESENT INVESTIGATION

2.1 AIM

Lip reading, also known as visual speech recognition, aims at predicting the sentence being spoken, given a muted video of a talking face. Thanks to the recent development of deep learning and the availability of big data for training, lip reading has made unprecedented progress with much performance enhancement.

2.2 SCOPE

Lipreading is a process of extracting speech by watching lip movements of a speaker in the absence of sound. Humans lipread all the time without even noticing. It is a big part in communication albeit not as dominant as audio. It is a very helpful skill to learn especially for those who are hard of hearing.

Deep Lipreading is the process of extracting speech from a video of a silent talking face using deep neural networks. It is also known by few other names: Visual Speech Recognition (VSR), Machine Lipreading, Automatic Lipreading etc.

The primary methodology involves two stages:

- i) Extracting visual and temporal features from a sequence of image frames from a silent talking video
- ii) Processing the sequence of features into units of speech e.g. characters, words, phrases etc. We can find several implementations of this methodology either done in two separate stages or trained end-to-end in one go.

CHAPTER- 3

EXPERIMENTAL OR MATERIAL AND METHODS, ALGORITHMS USED

3.1 DATASETS AND FEATURES

We used the MIRACL-VC1 data set [0] containing both depth and color images of fifteen speakers uttering ten words and ten phrases, ten times each. The sequence of images represents low quality video frames. The data set contains 3000 sequences of varying lengths of images of 640 x 480 pixels, in both color and depth representations, collected at 15 frames per second. The lengths of these sequences range from 4 to 27 image frames. The words and phrases are as follows:

Words: begin, choose, connection, navigation, next, previous, start, stop, hello, web

Phrases: Stop navigation, Excuse me, I am sorry, Thank you, Good bye, I love this game, Nice to meet you, You are welcome, How are you, Have a good time

For the sake of time and utilizing smaller data sizes, we focused on building a classifier that can identify which word is being uttered from a sequence of images of the speaker as input. We ignored the set of phrase data and also the depth images for the spoken word data. We built classifiers for both seen and unseen people. (Seen meaning that the model is trained on all people saying all words but saves certain trials for test and validation. Unseen removes people from training and adds them to exclusively to either testing or validation. The split is thirteen people for train, one for validation, and one for test.) The resulting datasets are (1200/150/150) (train/test/validation) examples for seen and (1300/100/100) (train/test/validation) examples for unseen. The class label distribution for the dataset is even as each person performs the same number of trials per word.

Preprocessing was an important part of working with this dataset. First, we utilized a python facial recognition library, dlib, in conjunction with OpenCV and a pre-trained model [2] to isolate the points of facial structure in each image and crop it to only include the face of the speaker, excluding any background that could interfere with the training of the model. We had to limit the size of every facial crop to a 90x90 pixel square in order to create uniform input data sequences for the model.

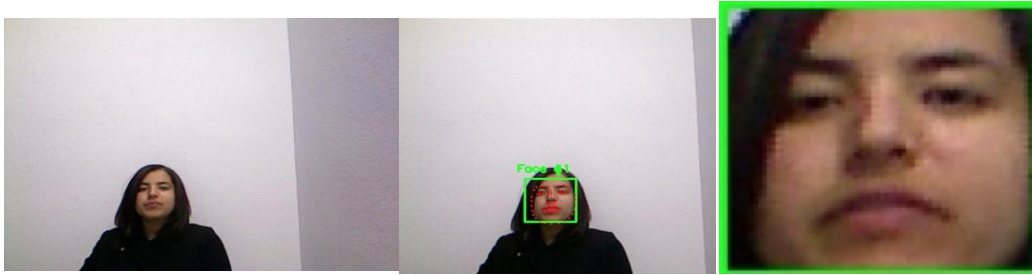


Fig -3.1.1: (left to right) Original Input image (part of a sequence) in the MIRACL-VC dataset; OpenCV and dlib facial recognition software labelling key points on around a detected face; final cropped image.

One issue with this data set is its small size. To increase the number of training sequences, we performed data augmentation. We tripled the data set in size by adding a horizontally flipped version of each image and a randomly pixel-jittered version of each image.

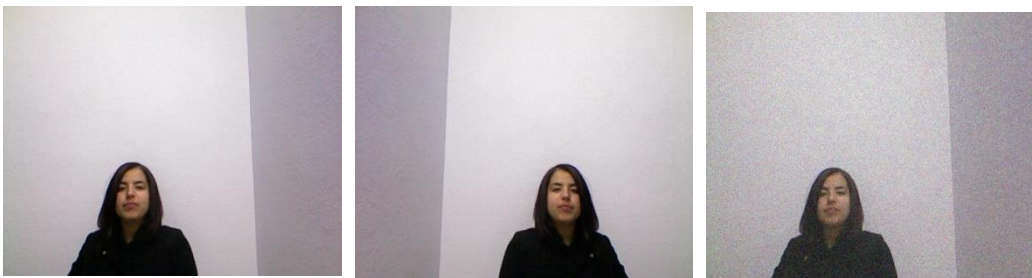


Fig -3.1.2: (left to right) Original Input image (part of a sequence) in the MIRACL-VC dataset; a horizontally flipped image; a jittered image.

In summary, each model receives a single image sequence as input with anywhere from 4 to 27 images in the sequence – and produces a single word classification label as output.



Fig -3.1.3: Example full input sequence of length 5. The subject is speaking “begin.”



Fig -3.1.4: Example full input sequence of length 10. The subject is speaking “hello.”

3.2 WORKING OF LIP-READING RECOGNITION WITH DIFFERENT MODELS

3.2.1 DYNAMIC TIME WARPING

The distance between two data and path wrapping values from sound are calculated by optimal wrapping path and the optimal wrapping path is calculated by Dynamic time warping algorithm. The distance between comparison is known as warping path. The two paths are said to be same, if the warping path produced are smaller. In different times two words from the same word by the same user can occur. For example, to can be pronounced with to or too. The time alignment problem is efficiently solved by Dynamic time warping method. Therefore, this algorithm is most often used in calculating

pattern similarities. The processed data will always remain in time zone, so the sequence of data will vary at time. The following diagram illustrates between two sequence of numbers.

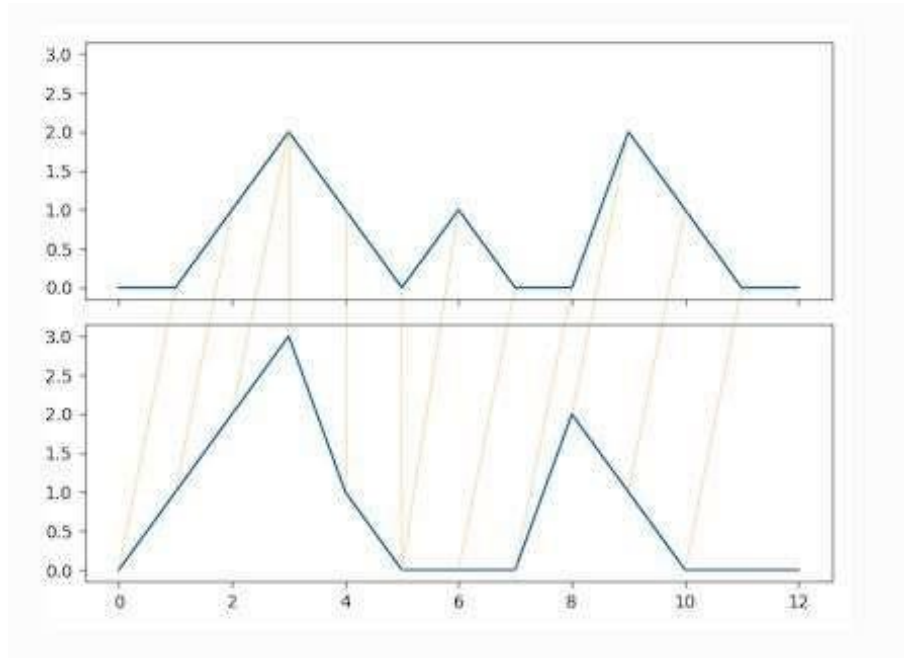


Fig -3.2.1: Dynamic time warping

3.2.2 SHAPE TEMPLATE

shape template with snakes is used to extract lip contours from image. The parameterized shape template describes the lips. An object is modelled within this shape template. The model can be made to fit the object in the image by adjusting its parameters. The template shape is based on previous knowledge of shape of the lips. To describe the shape of upper and the lower lips shape template consist of parabolas. These shape templates will interact dynamically along with image through an energy function which will draw the shape template on to salient features by changing the parameters of the parabolas. To produce fields containing features of interest such as valleys, peaks or edges are done by preprocessing of image. These fields are interacted through energy function by shape template. As the energy equation is minimized the parabolas will tend towards already known lip shapes that is the advantage of internal potentials in shape templates. These reduces the probability of shape

template getting on wrong medium which may be possible with snakes. The limited flexibility of using predefined lip shapes is the major drawback of shape template. The following figure explains how templates are used to improve lip reading recognition.

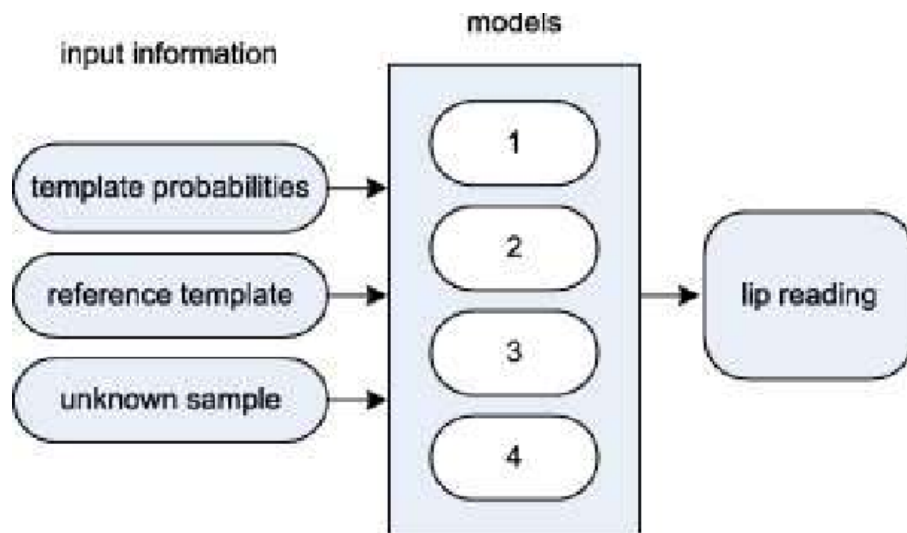


Fig -3.2.2: Shape Templates used as inputs to Lip reading.

3.2.3 SNAKE'S APPROACH

The most effective lip-reading system is developed by Bregler and Omohundro [11] which was the combination of trained lip shapes and snake algorithm. Which also contains the technique of nonlinear manifolds. In the feature space the lips are represented as points by configuration and also the surface or manifold of the space contains all the possible lip configuration. The configuration space produced by trained dataset of points and the structure and dimensions points lie is induced. Manually controlled snakes are trained using collected data. Unknown lip shapes are tracked using the snakes which are controlled by using learned manifold. From lip manifold of the image is estimated back project using initial crude. The grey level gradient is result of one iteration around the snake points. After completion of each iteration the lip manifold gets updated by snake point projection, so throughout the iteration only legal lip shapes are considered. The snake is trained to converge only to

legal lip shapes. The training data collected using manually controlled snake will determine the legal lip shapes. The system mentioned above uses modified snakes which controls through the use of learned lip shapes. It uses two methods that adapts to different lips without any previous training. The snakes can be controlled by two-dimensional pattern templates of lip edge contour instead of images gradient. The snakes should be able to provide the dynamics to move with the lips and adapt new shapes, while the pattern template should be able to provide stability.

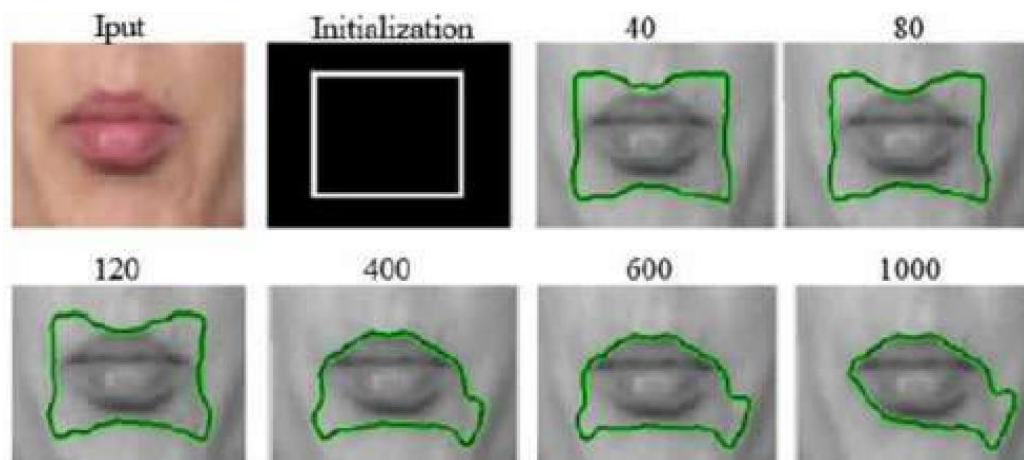


Fig -3.2.3: Lip localization using Snake's approach on input image. The number above each image shows the number of iterations needed for snake parameters tuning.

3.2.4 HIDDEN MARKOV MODEL

Hidden Markov Model is trained with visual features which has three different states and diagonal covariance with Gaussian Mixture Model concerned with each task which all are sequence of embedded trained and which has tested with view angle dependence. Visual speech synthesis approaches and the generated visual parameters by HMM which uses constraints of dynamic ("delta") features. Visual parameter trajectories can predict the mouth motions rendered by the video. The drawback of Hidden Markov Model based in visual speech synthesis can be the resulted blurring because of the feature dimension

which reduces statistical modeling. With this automatic lipreading which uses continuous density Hidden Markov Models as a pattern matching which is done by statistical components. The mixture of diagonal covariance matrix and the multi-dimensional gaussian mixtures are the result of the probabilities of the observed pattern matching of Hidden Markov Model. This method is specifically considered for lip-reading recognition where the independent context, the consideration of whole word, lip shape, facial expression, accent, etc. With the exception of some results which can be not that accurate considering different scenario all the Hidden Markov Model parameters are estimated by Viterbi training dataset, where a different training algorithm is used which is known as generalized probabilistic descent. To train this the available audio and the identical visual counterparts are synced together to obtain a string length which will be used for recognition.

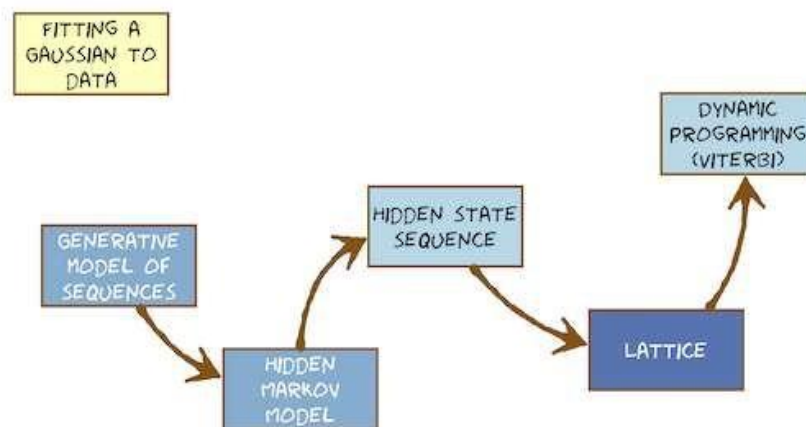


Fig -3.2.4: speech recognition using hidden markov model

3.2.5 CONVOLUTIONAL NEURAL NETWORKS MODEL

Lip reading issue can be resolved by CNN and LSTM methods, by using the excellent convolution neural networks the features of images are derived. CNN model contains 3 different convolution layers, these 3 layers are designed with

regularization. AlexNet, GoogleNet supported CNN by structured are employed to differentiate and compare the performance with designed with CNN model. Classification model and dataset size can affect the performance of the module. Data augmentation techniques uses address data augmentation. Experimental studies that are based on CNN models which can be with or without augmented dataset it depends on various mini batch size values are to be performed.

- AvLetters dataset is used to transfer learning which is supported by CNN by using AlexNet
- CNN architecture is also compared and proposed by supporting transfer learning of CNN
- This CNN method is not only having better performance it is also easy to develop. It provides the whole information of described CNN and the model which are pre-trained used throughout the research.

Convolutional Neural Networks (CNN) comes under artificial neural network which is specialized to maintain multi-dimensional and large set of data. Least layers of convolution process is used by convolutional networks which are neural networks instead of using general matrix multiplication. The main and the basic components of CNNs are pooling layer, activation functions and convolution layer, fully connected layers, regularization, loss layer, optimization. The learnable filter set is included by convolutional layer. The ability to learn with a fully connected layer is the structure layer of convolutional network. The important parameters for this layer are number of filters, stride and spatial extend. The pooling layer reduces the network model's cost. Pooling layer makes the system resistant to a very small position changes, the pooling method constantly uses the operations such as Minimum, Maximum, Sum, Average, etc. There is also choice of activating the function as ELU, sigmoid, ReLU, etc., apparently it also affects the performance of CNN. ReLU, linear function which is piecewise, the return negative inputs make to zero activation function and, also without changing the output positive input. To achieve higher classification accuracy ELU which is an activation function is

used, which also allows neural networks. The frequently used function is sigmoid function which continuous and derivable function. After convolution and pooling layers fully, connected layer comes in CNN. This study has all the layers and neurons are fully connected to previous layers. The last layer of CNN is the Loss Layer, which results the comparison should be evaluated between the predicted labels and the final evaluated labels during the training of the model. To prevent loss function issue SoftMax is used and to resolve overfitting problem Regularization techniques is used, which is important the most important problem for deep neural networks. The following diagram illustrates how the CNN and LSTM methods are divided into two layers and by layering them how we get the results.

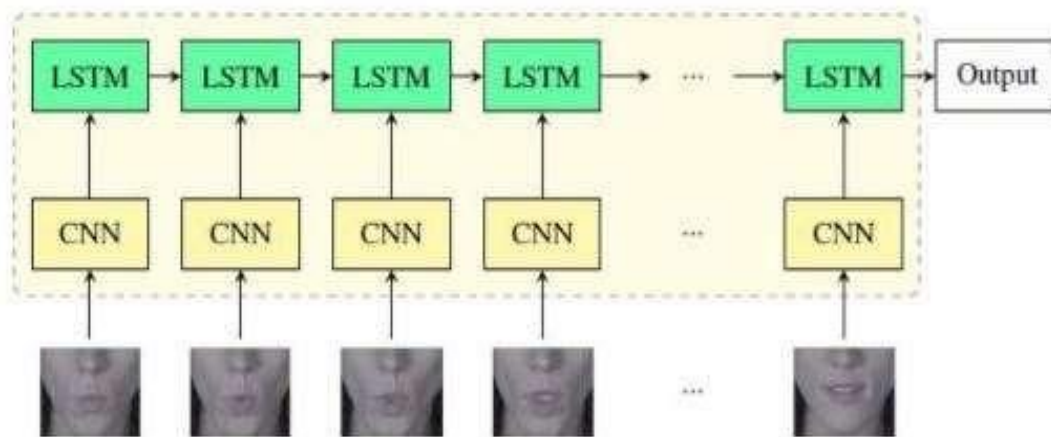


Fig -3.2.5: Lip-Reading Recognition using Convolutional Neural Networks (CNN) model.

3.3 METHODS

In this section we describe the different models that we created to solve the lip reading problem. We created four models: a Baseline CNN + LSTM network; a more robust and deep layered CNN + LSTM network inspired by Deep Mind's LipNET[1]; an LSTM network placed on top of bottleneck features developed by a VGG16 network pre- trained on ImageNet; and the same LSTM network on top of VGG16 with fine-tuning of the last

convolutional block.

3.3.1 CNN + LSTM BASELINE

Our first model ran every image of our sequenced input through a Convolutional Neural Network and then fed the flattened outputs as a sequence into a Long Short Term Memory Recurrent Neural Network, which produced a single output, making it a many-to-one RNN. We then added a Fully Connected layer that mapped to 10 units, and used a softmax activation layer to produce the probabilities of every word, of which we took the highest:

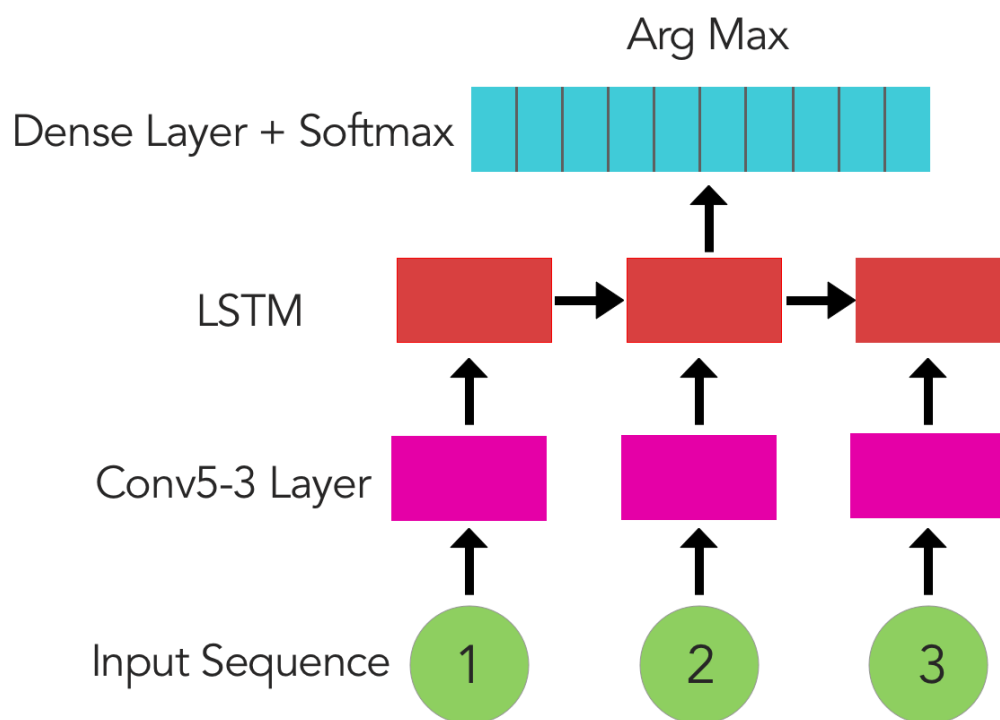


Fig -3.3.1: CNN + LSTM Baseline model layer architecture diagram

Our convolutional layer had a kernel size of 5x5 and depth of 3 filters -- inspired from LipNET's architecture and was added to help the model make sense of the high- level features of the images. It achieves this by running the kernel across the image, mapping the dot products of the pixel overlaps to a new layer, and stacking together the layers produced by every filter:

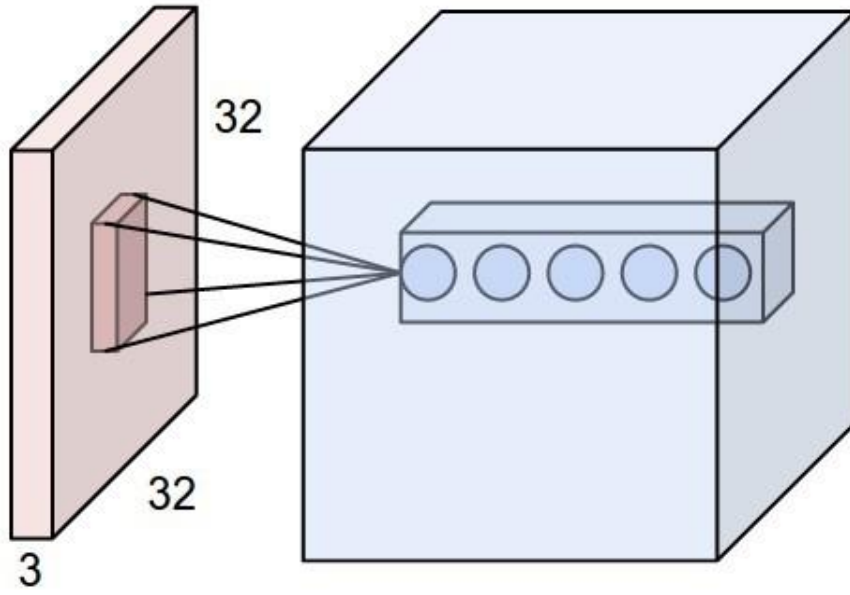


Fig -3.3.2: Structural diagram courtesy of CS231N

The LSTM was added to package the entire sequence of CNN outputs into a single layer without losing the temporal understanding of the video frames. In particular, an LSTM fixes the vanishing gradient problem present in vanilla RNNs, which inhibits the backpropagation of gradients to occur [16]. It does so by adding 4 gates (input (i), forget (f), output (o), new memory (c)) whose activations can be learned, in order to control whether or not to hold on to information:

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) \quad \text{(Input gate)}$$

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) \quad \text{(Forget gate)}$$

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) \quad \text{(Output/Exposure gate)}$$

$$\tilde{c}_t = \tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) \quad \text{(New memory cell)}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \quad \text{(Final memory cell)}$$

$$h_t = o_t \circ \tanh(c_t)$$

Given that we use softmax as our last activation, our loss function is cross entropy loss:

$$loss = - \sum_i \log \left(\frac{\exp(Wx_i)}{\sum_j \exp(Wx_j)} \right)$$

Finally, we used the Adam Optimizer to better navigate through the loss function.

3.3.2 DEEP LAYERED CNN + LSTM

We expanded on our baseline by first adding 2 more layers of CNNs, in order to develop an understanding of more intricate features in our input images. We made our LSTM bidirectional, to avoid outweighing the output with frames in the latter parts of the sequence, and added dropout and batch normalization after every CNN layer. We kept our dropout probability at 0.2 given that we performed it several times across the model. We also interspersed 2x2 Max Pooling layers with strides of 2 between the CNNs. This model is even more similar to LipNET's:

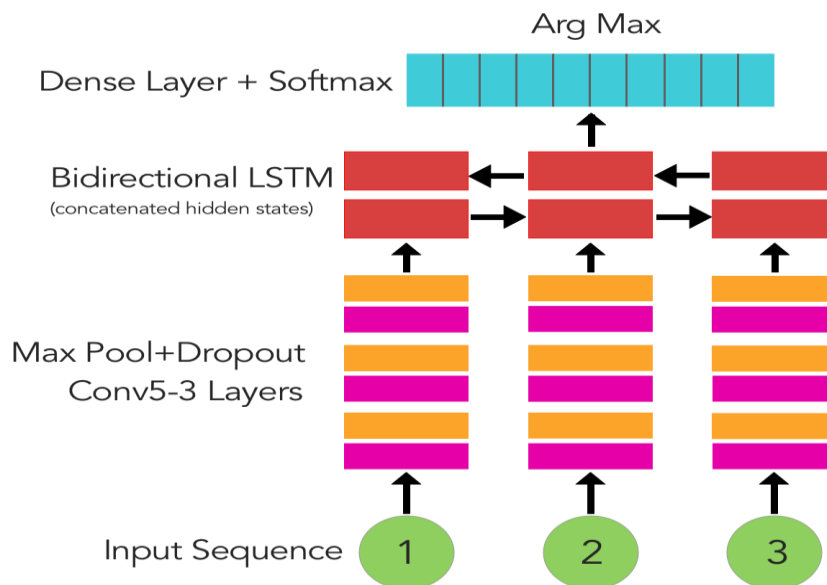


Fig -3.3.3: Deep Layered CNN + LSTM model layer architecture diagram

3.3.3 IMAGE-NET PRETRAINED VGG- 16 FEATURES + LSTM

Given that we decided to focus only on words and not phrases, we were limited to 1500 data points (15 people uttering 10 words for 10 iterations). This is a really small dataset by deep learning standards; as a result, we decided to employ transfer learning by making use of a VGG-16 network pre-trained on ImageNet[17]. VGG networks [18] were developed under the premise that a smaller kernel size allows for deeper networks without increasing the number of parameters, thus increasing the number of non-linearities without the need of greater memory usage.

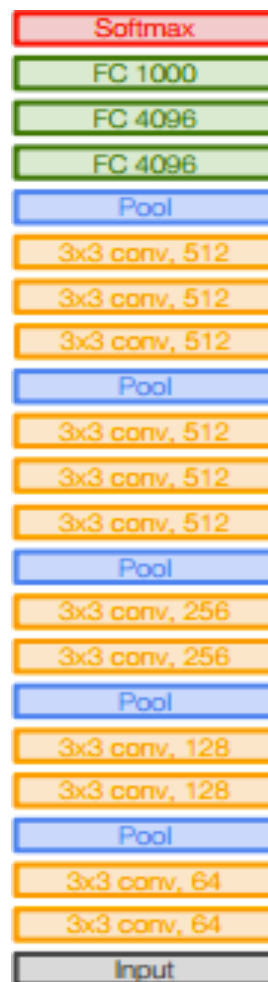


Fig -3.3.4: VGG-16 Structure

ImageNet is a dataset currently consisting of more than 14 million images, some of which contain annotations and object bounding boxes. 952,000 of those images are of humans, reason for which this network could be utilized for the lip reading problem.

We extracted the bottleneck features of our dataset up until the last convolutional layer of VGG. We then fed this result into the top portion of our second model -- the Bidirectional LSTM, and performed the training. This meant that we froze the VGG model weights, and only updated the weights of our LSTM and Dense layer. This sped up training considerably, given that our data was only processed by the VGG network once per training trial.

3.3.4 FINE-TUNED VGG- 16 + LSTM

Our last model consisted of unfreezing the last convolutional layer block of the pretrained VGG-16 model (3 CNN layers with 512 3x3 kernels) and training it alongside the Bidirectional LSTM and Dense Layers that we placed on top. This was done with the intention of capturing the more complex features of our input images, since later CNN layers tend to capture less obvious characteristics of an image. Given that the unfrozen convolutional layer was initialized with pretrained ImageNet weights, we also initialized the LSTM and Dense layers with the weights of the top portion of our third model (the LSTM and Dense layer training on top of the VGG-16 bottleneck features), to prevent a random weight initialization from recklessly modifying the VGG weights when backpropagating. We also prevented this by switching to a regular SGD optimizer rather than Adam, since SGD is subtler in updating weights.

CHAPTER-4

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

4.1 RESULT AND DISCUSSION

We used accuracy as our primary metric, although we also looked at the confusion matrices of these models to better understand where the errors were occurring. With 10 classes, a random baseline for this classifier is 0.1. All four of our models outperformed this baseline, with the Fine-tuned VGG-16 + LSTM achieving top test accuracy. We tested on both seen and unseen subjects. Results for seen subjects were relatively good, but our accuracy for unseen subjects gravitated barely above the random choice metric of 10% for all ten models.

Model	Training	Validation	Test
Baseline	85%	64%	39%
Deep CNN +LSTM	52%	39%	25%
Frozen VGG +LSTM	100%	76%	55%
Fine-tuned VGG +LSTM	100%	79%	59%

Fig -4.1.1: Seen Subject Accuracy

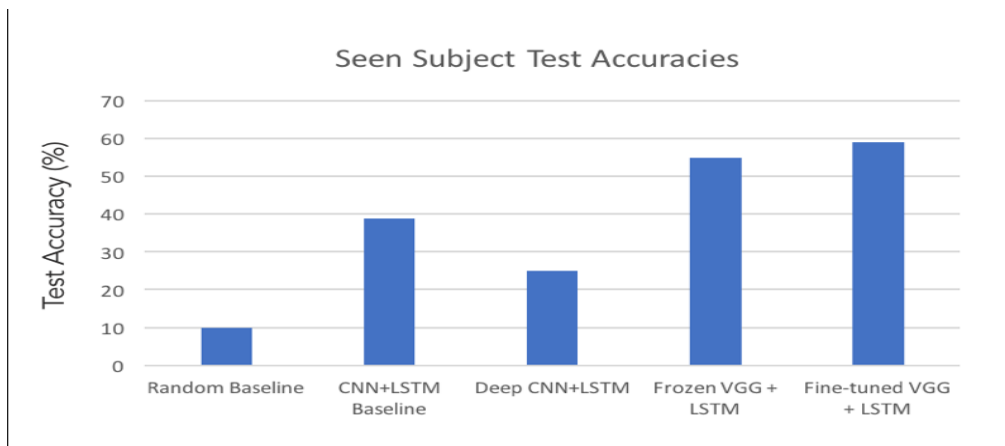


Fig -4.1.2: Seen Subject Accuracy Comparison Graph

4.2 CNN + LSTM BASELINE

Our baseline got surprisingly good results for seen subjects, with a 39% accuracy for the test set. As a result, we suspect that it is relatively easy to distinguish between words uttered if the word and the subject have been seen multiple times before. This would mean that there's several high level features that make this distinction easy -- reason for which only one CNN layer was required to obtain decent results.

4.3 DEEP LAYERED CNN + LSTM

Our deep layered model performed worse than our baseline, with a test accuracy of 25%, significantly lower than validation(39%) and training(52%)

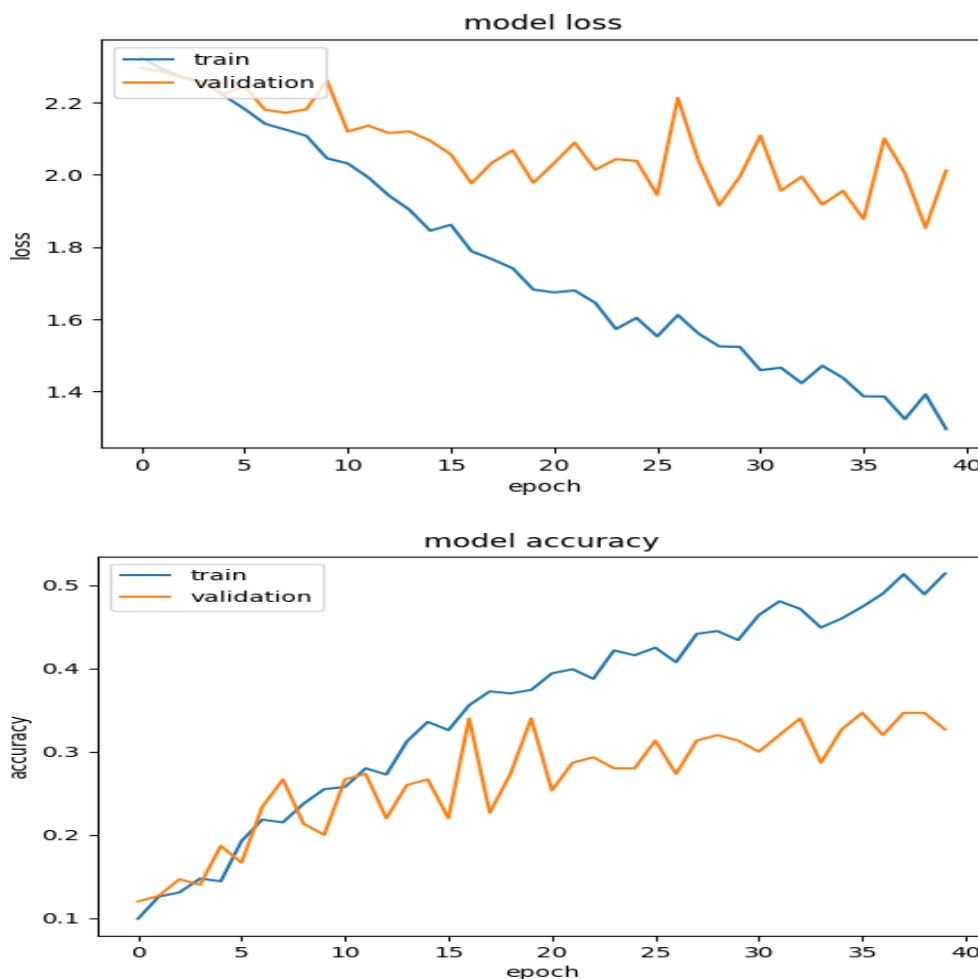


Fig -4.3.1: Loss and Accuracy Plots for Deep-Layered CNN + LSTM

Building up on the analysis made for the baseline, it seems like classifying on words spoken by seen subjects is not complex, and thus having the complex representations outputted by our three layered convolutional network was unnecessary. Another distinction between this model and the baseline was the addition of dropout and batch normalization. Again, it seems that regularization wasn't necessary for seen subjects, so it harmed this model's performance. The jagged loss plot below suggests that the model was having trouble navigating the validation loss space, which might suggest that it did not have a good intuition for what features to look for when making predictions.

4.4 IMAGE-NET PRETRAINED VGG- 16 FEATURES + LSTM

This model achieved 100% training accuracy, which is probably due to the fact that we added the frozen VGG-16 model pre-trained on ImageNet. The learning rate used for this model was 0.00006, which is dramatically smaller than that used in the above two models (0.0001 and 0.001, respectively). Again, this was probably due to having pre- trained weights, although it's somewhat intriguing because the VGG weights were frozen, and the LSTM and Dense layer weights were randomly initialized. This might suggest that the representations outputted by the VGG were much more significant than the work done by the LSTM and Dense layer that we added on top, so the model worked best when we mitigated the effect of the top model by reducing the learning rate.

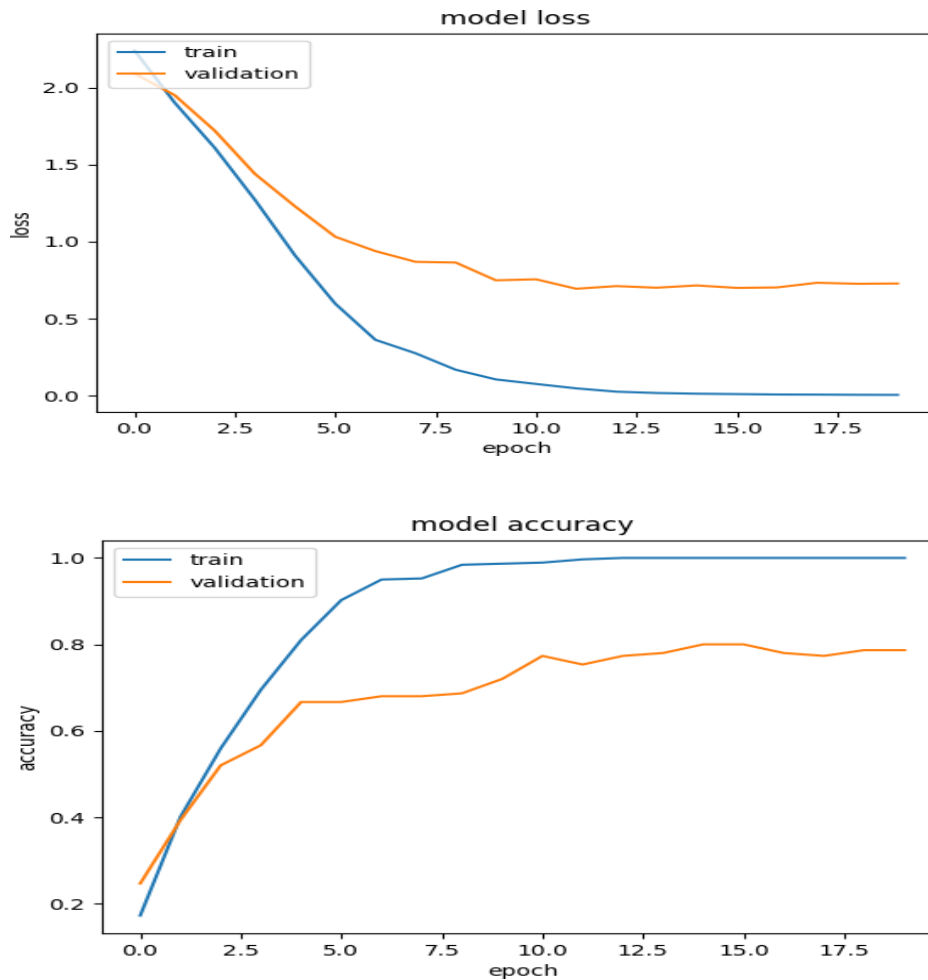


Fig -4.4.1: Loss and Accuracy Plots for ImageNet Pretrained VGG-16 Features + LSTM

The accuracy plot suggests that more regularization could have lifted our validation and test scores. However, we found that increasing the dropout rate made the model worse; we suspect that the short length of our dataset was what caused this behavior. We thus decided to stick to a 0.2 dropout probability.

4.5 FINE-TUNED VGG- 16 + LSTM

The fine-tuned VGG-16 + LSTM gave us the best test results, with 59% test accuracy. A breakdown of predictions can be seen in its confusion matrix in Figure 4.5.1 .

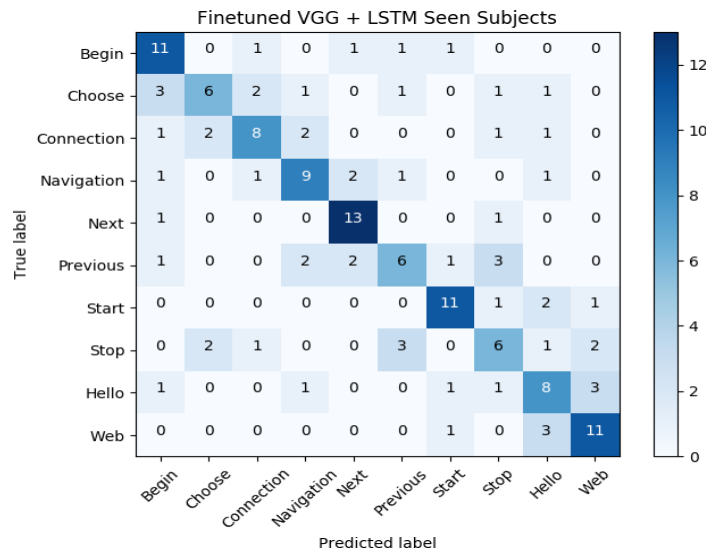
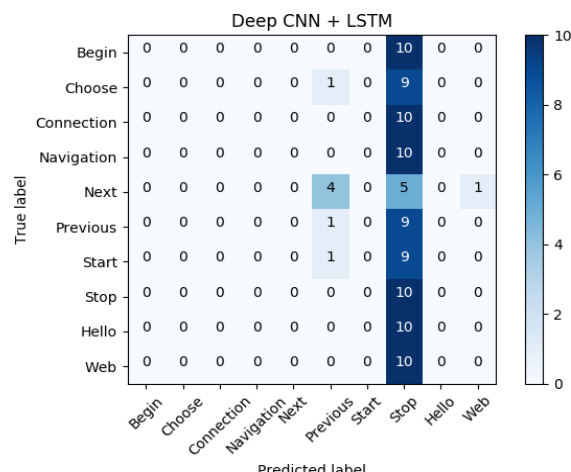


Fig -4.5.1: Confusion Matrix for VGG-16 + LSTM

This model was only trained for 5 epochs with a learning rate of 0.0001 -- it's validation accuracy shot up to 79% immediately and stayed there, with it's validation loss also staying constantly at slightly above 0.7. It's training loss was almost 0 from the start for training. This behavior is due to the pre-trained weights that were added to the LSTM and Dense layer, which were generated in the previous model. As a result, the only difference between both models was the fine-tuning that occurred in the last 3-layered convolutional block of the VGG-16 model, which helped us improve testing accuracy by 5%.

4.6 UNSEEN DATA

Upon further analysis, we found some possible explanations for our bad results in unseen validation and test data (which again, wavered slightly above 10%). This is the confusion matrix for our second model, the Deep CNN + LSTM on unseen subjects.



Our model is predicting “Stop” for 92% of the words. We realized cross-validation could have helped mitigate this issue; a possible explanation for this result is that the person in the test set spoke faster than any other subject, and as a result, most of the words uttered by the subject are thought to be “stop”, since “stop” has perhaps the shortest pronunciation within the dataset. The confusion matrix of unseen subject for our fine-tuned VGG model also suggests a correlation between short words:

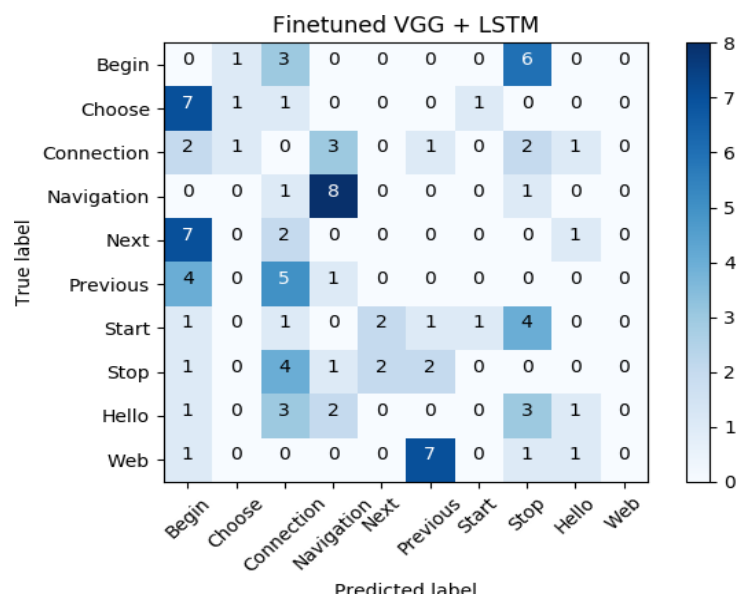


Fig -4.6.1: Confusion Matrix for VGG + LSTM

“Begin” is predicted for “Choose” and “Next” a total of fourteen times. But it’s also interesting to note that “Web”, “Choose”, and “Start” were predicted a total of only five times; this might suggest that the model chooses only a few words to focus its predictions on -- again probably because of our limited data set. In the future, a bigger data set, cross validation, and data augmentation could help us improve our score for unseen subjects.

One way we attempted to improve results for unseen subjects was with data augmentation. We saw with the ImageNet Pretrained VGG-16 Features + LSTM, that although validation accuracy is still abysmal, the

addition of randomly augmented data (flipped and jittered) improved validation accuracy by a factor of two. Unfortunately, time did not permit to run our other models with the augmented data, so we decided to exclude its use from our main analysis. It is, however, a vital addition that could help improve scores for this task when working with limited datasets in the future.

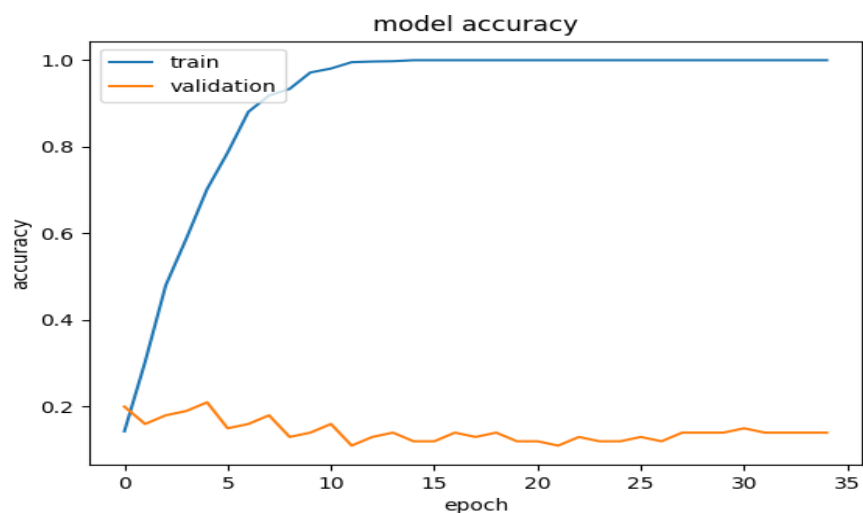


Fig -4.6.2: The ImageNet Pretrained VGG-16 Features + LSTM without data augmentation achieving an accuracy of 0.14

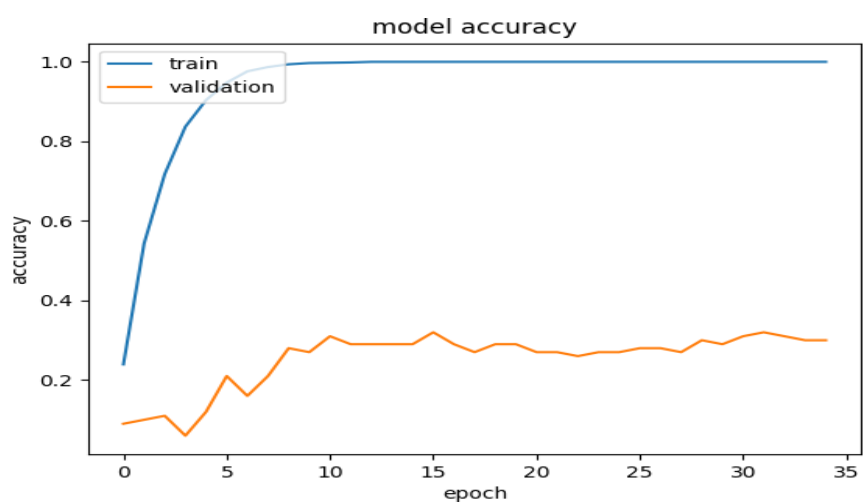


Fig -4.6.3: The ImageNet Pretrained VGG-16 Features + LSTM with data augmentation achieving an accuracy of 0.3

CHAPTER-5

SUMMARY AND CONCLUSIONS

Overall, we found that the inclusion of pre-trained facial recognition CNNs highly improved our models. The augmentation of our data proved helpful but only in the instance of unseen people. Our best model was the FineTuned VGG + LSTM. Regarding failures of specific models, the Deep Layered CNN + LSTM architecture was inspired by the LipNet architecture [1], which was designed to handle phrase inputs and is trained on a much larger corpus. Perhaps this explains why our baseline outperforms the Deep Layered CNN + LSTM architecture. Generally, in all models, we found it very difficult to avoid overfitting with unseen people. Thus, certain models and hyper parameters are a better fit depending on whether we are working with seen or unseen people for testing and validation. More work needs to be done to reduce overfitting even for seen people for the models that include pretrained networks. These reached training accuracies of 1 while validation accuracy remained close to .75.

Given more time and resources, the models outlined in this paper could be greatly improved. We think the addition of regularization would reduce the overfitting in our models even further. We also have yet to experiment with the number of filters in the fully connected layers. We only had 3 filters per layer, just like LipNET had, but other papers used anywhere from 64 to 512 filters per single CNN layer. Additionally, accuracy improvements could be found with further hyperparameter tuning and investigation of even more optimizer types. We also would have gotten value from saliency maps. Without them it is hard to know if the model is accurately focusing on mouth data or other aspects of the input sequences. Finally, performing analysis of confusion matrices earlier in our exploration process could have helped us mitigate the problems that we ran into with unseen subjects, given that we could have adjusted our models based on the patterns we perceived.

This project is easily extendible and raises the question of how to perform visual speech recognition on a much larger corpus (perhaps the entire

English dictionary). How could the addition of audio data improve our ability to interpret the video as text? Is it easier to understand speech from video of a single word being spoken or entire phrases and sentences? This question could easily be investigated since the MIRACL-V1 dataset includes phrase inputs and would be an interesting area of exploration. Additionally, most speech recognition tasks in real life require phrase inputs over single words.

REFERENCES

- [0] Ahmed Rekik, Achraf Ben-Hamadou, and Walid Mahdi. A new visual speech recognition approach for RGB-D cameras. In Image Analysis and Recognition - 11th International Conference, ICIAR, Vilamoura, Portugal, October 22- 24, 2016, Proceedings, Part II, pages 21–28.
- [1] Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. Lipnet: Sentence-level lipreading. CoRR, abs/1611.01599, 2018.
- [2] Yuru Pei, Tae-Kyun Kim, and Hongbin Zha. Unsupervised random forest manifold alignment for lipreading. In The IEEE International Conference on Computer Vision (ICCV), December 2017.
- [3] Rekik A., Ben-Hamadou A., Mahdi W. (2015) Human Machine Interaction via Visual Speech Spotting. In: Battiato S., Blanc-Talon J., Gallo G., Philips W., Popescu D., Scheunders P. (eds) Advanced Concepts for Intelligent Vision Systems. Lecture Notes in Computer Science, vol 9386. Springer, Cham
- [4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2019.
- [5] C. Sui, R. Togneri, and M. Bennamoun, “Extracting deep bottleneck

features for visual speech recognition,” in ICASSP, 2015, pp. 1518–1522.

[6] S. Petridis and M. Pantic, “Deep complementary bottleneck features for visual speech recognition,” in IEEE ICASSP, 2016, pp. 2304–2308.

[7] Y. Li, Y. Takashima, T. Takiguchi, and Y. Ariki, “Lip reading using a dynamic feature of lip images and convolutional neural networks,” in IEEE/ACIS Intl. Conf. on Computer and Information Science, 2016, pp. 1–6.

[8] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, “Audio-visual speech recognition using deep learning,” *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, 2015.

[9] Ayaz A. Shaikh, Dinesh K. Kumar, Wai C. Yau, M. Z. Che Azemin, and Jayavardhana Gubbi. Lip reading using optical flow and support vector machines. 2010 3rd International Congress on Image and Signal Processing, 1:327–330

[10] Koller, O., Ney, H., Bowden, R.: Deep learning of mouth shapes for sign language. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 85–91 (2015)

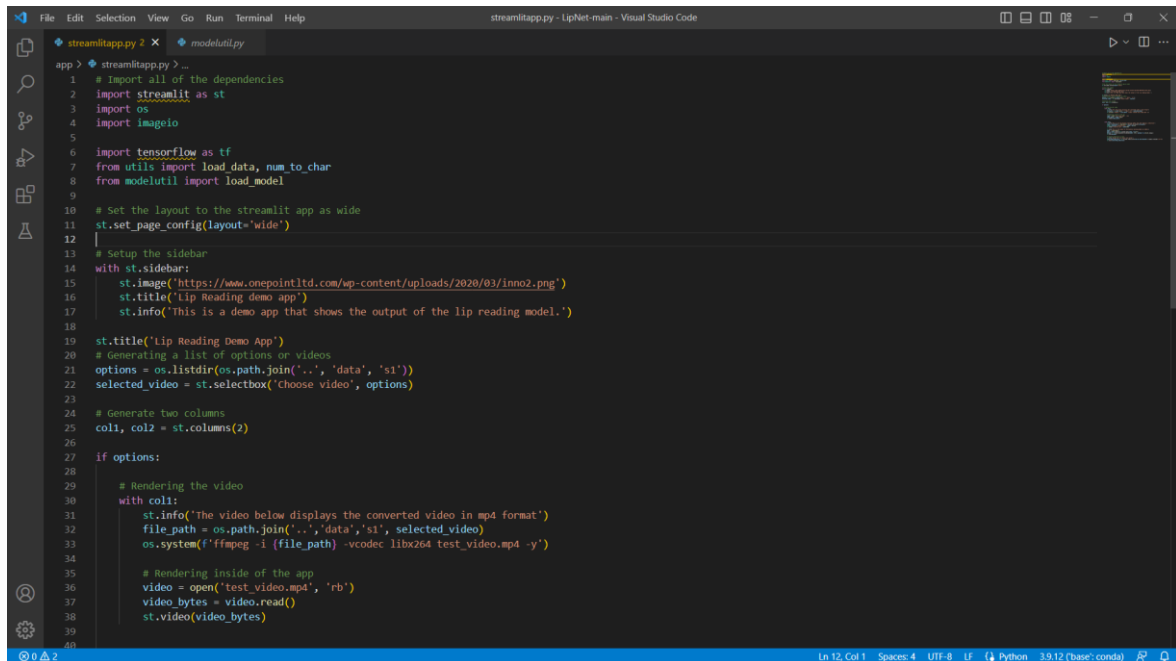
[11] Graves, A., Fernandez, S., Gomez, F., and Schmidhuber, J. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In ICML, Pittsburgh, USA, 2006.

[12] M. Wand, J. Koutnik, and J. Schmidhuber. Lipreading with long short-term memory. In IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6115–6119, 2016.

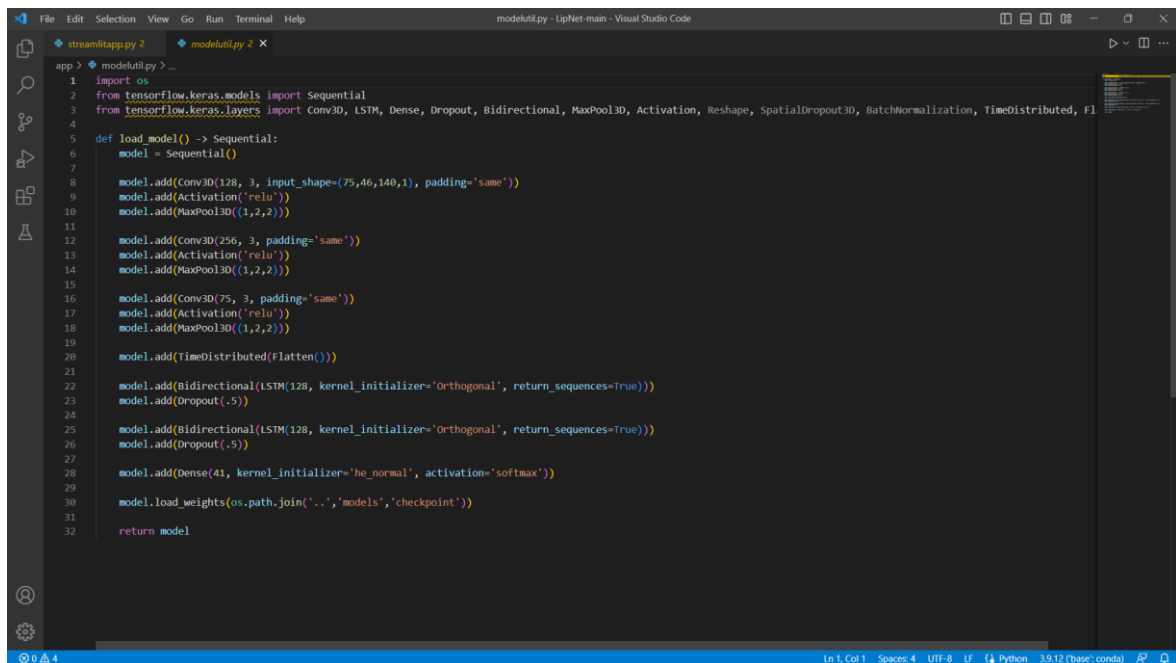
[13] J. S. Chung and A. Zisserman. Lip reading in the wild. In Asian Conference on Computer Vision, 2016.

[14] Ravi Garg, Vijay Kumar B. G, and Ian D. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. CoRR, abs/1603.04992, 2016.

SCREENSHOTS



```
1 # Import all of the dependencies
2 import streamlit as st
3 import os
4 import imageio
5
6 import tensorflow as tf
7 from utils import load_data, num_to_char
8 from modelutil import load_model
9
10 # Set the layout to the streamlit app as wide
11 st.set_page_config(layout='wide')
12
13 # Setup the sidebar
14 with st.sidebar:
15     st.image('https://www.onepointltd.com/wp-content/uploads/2020/03/inno2.png')
16     st.title('Lip Reading demo app')
17     st.info('This is a demo app that shows the output of the lip reading model.')
18
19 st.title('Lip Reading Demo App')
20 # Generating a list of options or videos
21 options = os.listdir(os.path.join('.', 'data', 's1'))
22 selected_video = st.selectbox('Choose video', options)
23
24 # Generate two columns
25 col1, col2 = st.columns(2)
26
27 if options:
28     # Rendering the video
29     with col1:
30         st.info('The video below displays the converted video in mp4 format')
31         file_path = os.path.join('.', 'data', 's1', selected_video)
32         os.system(f'ffmpeg -i {file_path} -vcodec libx264 test_video.mp4 -y')
33
34     # Rendering inside of the app
35     video = open('test_video.mp4', 'rb')
36     video_bytes = video.read()
37     st.video(video_bytes)
```



```
1 import os
2 from tensorflow.keras.models import Sequential
3 from tensorflow.keras.layers import Conv3D, LSTM, Dense, Dropout, Bidirectional, MaxPool3D, Activation, Reshape, SpatialDropout3D, BatchNormalization, TimeDistributed, Flatten
4
5 def load_model() -> Sequential:
6     model = Sequential()
7
8     model.add(Conv3D(128, 3, input_shape=(75,46,140,1), padding='same'))
9     model.add(Activation('relu'))
10    model.add(MaxPool3D((1,2,2)))
11
12    model.add(Conv3D(256, 3, padding='same'))
13    model.add(Activation('relu'))
14    model.add(MaxPool3D((1,2,2)))
15
16    model.add(Conv3D(75, 3, padding='same'))
17    model.add(Activation('relu'))
18    model.add(MaxPool3D((1,2,2)))
19
20    model.add(TimeDistributed(Flatten()))
21
22    model.add(Bidirectional(LSTM(128, kernel_initializer='Orthogonal', return_sequences=True)))
23    model.add(Dropout(.5))
24
25    model.add(Bidirectional(LSTM(128, kernel_initializer='Orthogonal', return_sequences=True)))
26    model.add(Dropout(.5))
27
28    model.add(Dense(41, kernel_initializer='he_normal', activation='softmax'))
29
30    model.load_weights(os.path.join('.', 'models', 'checkpoint'))
31
32    return model
```

The image shows a Visual Studio Code editor window with a dark theme. The title bar at the top reads 'utils.py - LpNet-main - Visual Studio Code'. The editor has two tabs open: 'streamlitapp.py 2' and 'utils.py 2 x'. The 'utils.py' tab is active, displaying a Python script. The script defines a vocabulary, a function to load video frames, a function to load alignments, and a function to load data. The status bar at the bottom shows 'Ln 18, Col 32', 'Spaces: 4', 'UTF-8', 'LF', 'Python 3.9.12 (base\'s conda)', and a search icon. The left sidebar contains icons for Explorer, Search, Run and Debug, Extensions, and Settings. The right sidebar shows a file explorer for the project.