Lip Reading using Deep Learning

Robin Anburaj B
Department of Artificial Intelligence and Machine Learning
Sri Shakthi Institute of Engineering and Technology Coimbatore, India
robinanburajb22aml@srishakthi.ac.in

Dinesh J
Department of Artificial Intelligence and Machine Learning
Sri Shakthi Institute of Engineering and Technology Coimbatore, India
dineshj22aml@srishakthi.ac.in

Deepak T
Department of Artificial Intelligence and Machine Learning
Sri Shakthi Institute of
Engineering and Technology Coimbatore, India
deepakt22aml@srishakthi.ac.in

Mrs. R. Hemavathi
Department of Artificial Intelligence and Machine Learning
Sri Shakthi Institute of Engineering and Technology Coimbatore, India
hemavathiaiml@siet.ac.in

Abstract**—** Lip reading, the process of interpreting speech by visually observing the movements of the lips, has emerged as a critical area of research with applications spanning communication aids for the hearing impaired, silent speech interfaces, and enhanced human-computer interaction. This paper reviews recent advancements in lip reading technologies, focusing on the integration of machine learning and computer vision techniques. We explore state-of-the-art methods including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer-based models that have significantly improved the accuracy and robustness of lip reading systems. The study highlights the importance of large annotated datasets, such as LipNet and LRW, which have facilitated the training of deep learning models.

Additionally, we examine multimodal approaches that combine visual information with audio signals to enhance performance, especially in noisy environments. Despite substantial progress, challenges remain in addressing speaker variability, low resolution, and real-time processing. Future research directions are discussed, emphasizing the need for more diverse datasets, improved model generalization, and real-world application testing. This comprehensive review underscores the potential of advanced lip reading technologies to revolutionize communication accessibility and human-computer interaction.

This paper presents the method for Vision based Lip Reading system that uses convolutional neural network (CNN) with attention-based Long Short-Term Memory (LSTM). The dataset includes video clips pronouncing words sentence. The pretrained CNN is used for extracting features from pre-processed video frames which then are processed for learning temporal characteristics by LSTM. The SoftMax layer of architecture provides the result of lip reading. In the present work experiments are performed with two pre-trained models. The system provides 80% accuracy using Tensorflow and ensemble learning.

Keywords— CNN; RNN; LSTM; Tensorflow; lip reading; deep learning

INTRODUCTION

In recent years, machine learning has made substantial influence on social progress, promoting the rapid growth and development in artificial intelligence technology and solving a variety of real-world problems. Many human-computer interaction and virtual reality (VR) technologies are built using automatic lip reading technology. It has the potential to be extremely useful in visual perception and human language communication. Automatic lip reading can help learning and understanding lip language and lip movements for speech recognition by reducing the time and effort required. In noisy environments where audio speech recognition may be problematic, visual lip reading plays a important role in human- computer interaction. It can also be used as a hearing aid for those who are having hearing disabilities. An automatic lip reading system can be utilized to recognize speech, making the life of hearing-impaired people easy. If audio in a video is not of good quality or audio is noisy, then speech to text recognition is difficult, so in such cases vision based lip reading system helps to get accurate result.

The feature extraction and classification are usually the two processes in traditional lip reading systems. Previously, most feature extraction algorithms used pixel values taken from the region of interest (ROI), that is, mouth as visual input in the first stage. The abstract image features are retrieved using principal component analysis (PCA), discrete cosine transform (DCT), discrete wavelet transform (DWT) etc. [1]. The second stage involves support vector machine (SVM) and hidden Markov model classifiers (HMM) for prediction based on visual features obtained from first stage [1]. This method assures that no information is lost but size of the feature vector can be very large and may also contain considerable redundancy. The transformation method concentrates the majority of the image's energy into a limited number of coefficients thus eliminating redundancy. The transform coefficients are ranked according to the importance of the data they represent [2]. The classification is mostly based on static and dynamic data. The dynamic data have the spatial and temporal variations. HMM, Artificial Neural Network (ANN), Gaussian Mixture Model (GMM) and other classifiers can be defined [3].

Due to significant growth and development in the field of computer vision along with deep learning, the research is focused on the end-to-end deep learning architectures where there is no manual intervention for extracting the features. The LSTMs were introduced about couple of decades ago. Since then, a lot of success is observed in a variety of human language technologies, such as bidirectional LSTM based acoustic models and language models in speech recognition. Oscar Koller et al. [4] demonstrated CNN-LSTM network training for recognition tasks categorizing more than 1000 classes such as action gesture, activity and sign language interpretation. Although CNNs have made tremendous progress in gesture and sign language processing, motion appears to play a significant role in these tasks and relying on the HMM sequence for capturing temporal change may not be sufficient. The goal of combining deep CNN with LSTM layers is to make it easier to train the entire network. For efficient training, the LSTM can work with a vast amount of data. They can model nonlinearity without relying on Markov assumption, which may be an advantage over more standard models like HMM [4].

The proposed method for automatic lip reading recognition has three phases. The initial process is to extract keyframes from a sample video, which are then utilized to find critical points of the lip region or mouth and locate the ROI. This ROI is computationally processed in successive frames. The VGG19 network and ResNet50 are used to extract the characteristics from the original mouth image. First step is to preprocess the input video which includes extracting keyframes and positioning of mouth. The second is an attention-based LSTM network, which uses video keyframe features that helps learning sequential information with attention weights. SoftMax layer provides the final recognition result.

This paper is ordered in five sections. Section II provides literature review focusing upon the research efforts using deep neural networks. Section III provides detailed methodology proposed for automatic lip reading. Section IV presents the results along with critical observations and comments. Section V concludes the paper providing future direction.

I. LITERATURE REVIEW

Several research papers were reviewed to understand the work done in this area of automated lip reading. The researchers have used CNN based approaches with varying architectures of pre-trained models [1][5-9] and datasets. The highest accuracy reported with single word videos is 88%. Table I provides comparison of these approaches considering Methodology, Dataset used, Performance and limitations.

II. PROPOSED WORK

The proposed system uses CNN and attention-based LSTM. The block diagram of the lip reading system is shown in Fig. 1.

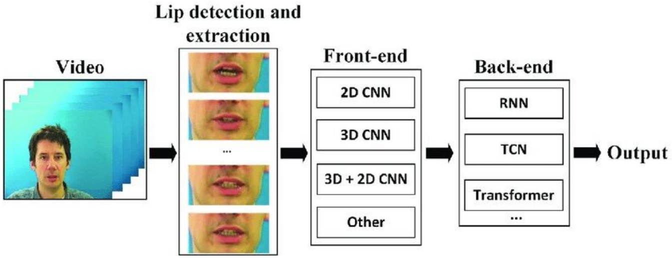


Fig. 1. Block diagram of Lip Reading System [1]

The video (without audio) is presented to the system as input. This video gets preprocessed to form the video sequence with region of interest. Then the CNN is used to extract spatial characteristics from the keyframes presented in terms of a feature vector. Further, these feature vectors will be used by attention-based LSTM to extract temporal characteristics from the sequence of frames. After LSTM the fully connected layers are added to predict the speech. The methodology followed to perform major tasks is explained in the following sub-sections.

A. *Preprocessing of input video*

As mentioned in Fig. 1 the input video needs to be preprocessed. To obtain the desired frames to be processed for feature extraction the steps as shown in Fig. 2 are followed. The input is the facial video of a person uttering some text. To understand some useful information from the input video there is a need to preprocess individual frames of video. Hence the first task is to extract the frames from the video.

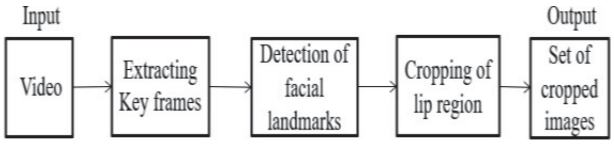


Fig. 2. Preprocessing of input video

TABLE I. LITERATURE REVIEW: COMPARISON OF SELECTED PAPERS

Paper / Year of Publication	**Methodology**	**Limitations**	**Dataset**	**Performance**
[1] /2019	"Deep convolutional neural network (VGG19) and attention – based long short-term memory"	Requires a good quality of video	Own dataset consisting of three males and three females	Accuracy of 88.2%
[5] / 2019	"Convolutional neural network along with pre-trained models (AlexNet, GoogleNet)"	Gives more accurate result specifically for alphabet level recognition.	AvLetters [10] dataset	Accuracy of 64.40%.
[6] / 2016	"VGGNet along with SVM"	Model trained from scratch did not perform well as size of dataset is small	MIRACL-VC1 [11] dataset	Accuracy of 76%.
[7] / 2017	"Deep convolutional neural network (VGG16) and attention – based long- short term memory"	Does not support larger dataset	MIRACL-VC1 dataset	Validation accuracy of 79% and test accuracy of 59%.
[8] / 2020	"Viseme concatenation and 3D Convolutional Neural Networks"	Limited training data	MIRACL-VC1 dataset	Accuracy of 76.89%.
[9] / 2017	"Combination of spatiotemporal convolutional, residual and bidirectional Long Short-Term Memory networks."	If phonetic and "visemic" content of the word pairs are similar then correct identification of the first and last viseme of a word may be difficult.	LRW [12] dataset	Accuracy of 83.00%.

Usually, videos are acquired with a frame rate of 71 frames in three second. To build the model, it becomes essential to extract the features and to get the hidden relationship of the sequence of frames. The way the word gets pronounced and the length of each utterance is different concerning the subject uttering the word. Also, it may happen that while pronouncing a specific word there is a series of redundant information regarding the lip movements. Therefore, the redundant information must be removed from all the extracted original frames. It will also help to balance with training speed and recognition results. Thus, the proposed method does not use all the frames from the video sequence, instead only the keyframes are being extracted and the lip regions are segmented for further processing.

Extraction of keyframes

The video or the time of the utterance or the total number of frames is split into 10 equal intervals, and a random frame is picked from each interval as a keyframe. Thus, video gets converted into 10 keyframes providing uniform input length. For example, if a video comprises of 40 frames, then it is partitioned into 10 parts with equal intervals. That is, 10 partitions with four frames per interval. Further after randomly picking 1 frame per partition, the entire video now gets converted into a sequence of 10 frames.

Detection of facial landmarks

The keyframes extracted from the video are further processed to detect the Facial landmarks. Detecting facial landmarks can be seen as a subset of the problem of shape prediction. Given the input typically ROI indicating object of attention, the shape predictor localizes interest points or the key points around the shape. In the context of facial landmarks, these methods attempt to identify key facial structures on the identified face.

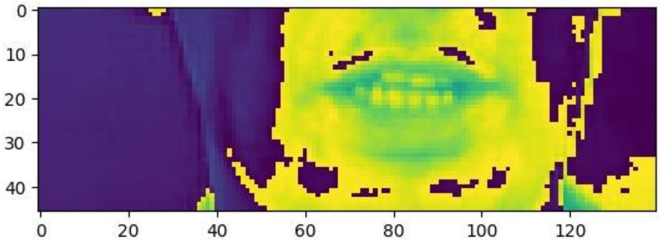


Fig. 3. Detection of facial landmarks and cropping of lip region

It involves a two-step process: i) Locating the face in the image. ii) Identifying the most important facial structures in the ROI of face. The dlib library is used for detecting landmarks. The facial landmark detector of the dlib library is based on the research paper by Kazemi and Sullivan [13]. This algorithm provides 68 facial landmarks. The indexes of these landmarks can be viewed as in the image shown in Fig. 3 (a).

Localization of mouth /lip region

After detection of facial landmarks, referring to the points representing the lips in the image, the lip region is cropped. The points that were considered for cropping the lip area are 49, 55, 52, and 58 as shown in Fig. 3 (b) which are the extreme points depicting the lips. Each keyframe is cropped as per the coordinates of the lips and resized to the standard size of 224 x 224.The output of the preprocessing step is the set of 10 cropped lip images per video.

B. Implementation of CNN and LSTM

The set of cropped images are given as input to the spatial characteristics block consisting of CNN as shown in Fig. 4 The VGG19 and ResNet50 CNN architectures are used. VGG19 and ResNet50 give the set of spatial features (Feature vector) for each input frame. VGG19 architecture is as shown in Fig. 4. and Fig. 5 describes the architecture of ResNet50 pre-trained network.

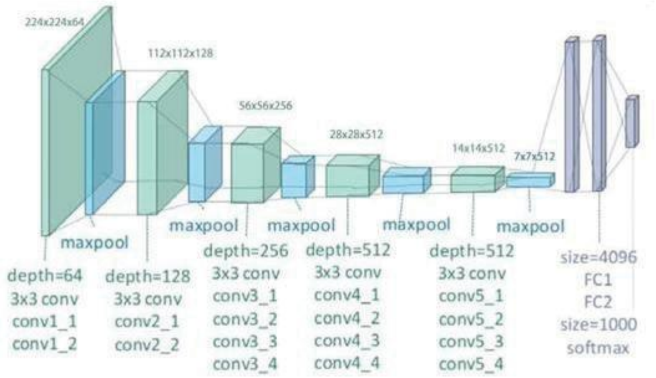


Fig. 4. Architecture of CNN

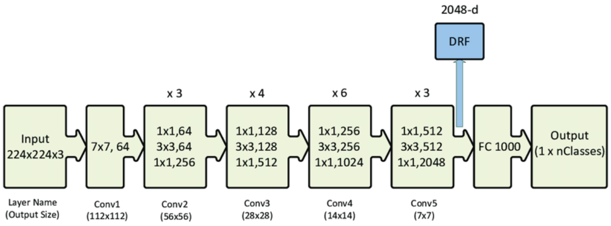


Fig. 5. Architecture of ResNet50 Pre-trained Network [14]

The pre-processed lip images of size 224 x 224 are given as input to Tensorflow and LipNet networks. For the purpose of feature extraction, Tensorflow architecture up to the first fully connected layer is used. For LipNet also the layers up to FC1000 are used. Thus, both the architectures provide feature maps for each frame of individual video. The size of feature vector are 4096 and 2048 dimensions for Tensorflow and LipNet respectively. The set of feature vectors then are provided as an input to the Time characteristics block of Fig. 1 consisting of attention-based LSTM. LSTM is a special type of RNN which will learn long-term dependency information.

LSTM with an attention mechanism is used so that model pays greater attention to the effective areas of the entire video. The feature vector for every frame is thus weighted and used as an input to the LSTM. The attention- based LSTM layers are introduced to add temporal information of lip sequences.

LSTM consists of different memory blocks called cells. There are three gates, input, output, and forget gate. The first step in LSTM will be of deciding to drop useless information from the cell state by using cell state and the new information will be stored in the cell state. Each time the cell state will be updated as per input given for learning sequence information (sequence relation) as per time interval.

The sequence information is provided to the fully connected layer which then flattens the sequence information into a single vector, which is given to the SoftMax layer. The SoftMax layer converts the output in terms of probabilistic result and based on the probability, results are predicted.

C. *Performance analysis*

Performance analysis is done by using a confusion matrix. The dataset gets divided into training and testing datasets to do the performance analysis of the algorithm/system. The overall accuracy gives the rate of correct prediction. Additionally, to get more insights about the results, class-wise and subject-wise accuracies are also obtained.

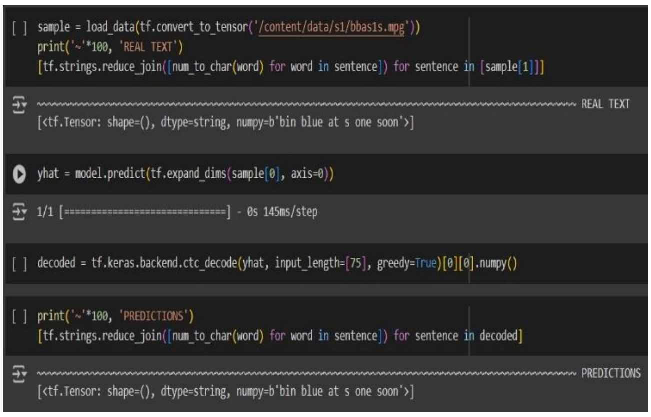
Further to improve the performance of system, ensemble learning is experimented.

Ensemble Learning

Ensemble methods in statistics and machine learning combine multiple learning algorithms to provide higher predictive performance than any of the individual learning algorithms. Thus, for implementation of ensemble method, seven different models are used. Experiments are performed with CNN + LSTM and also with CNN + attention-based LSTM using VGG19 and ResNet50. Each model votes for a particular class. The class with maximum voting will provide the final result of the method.

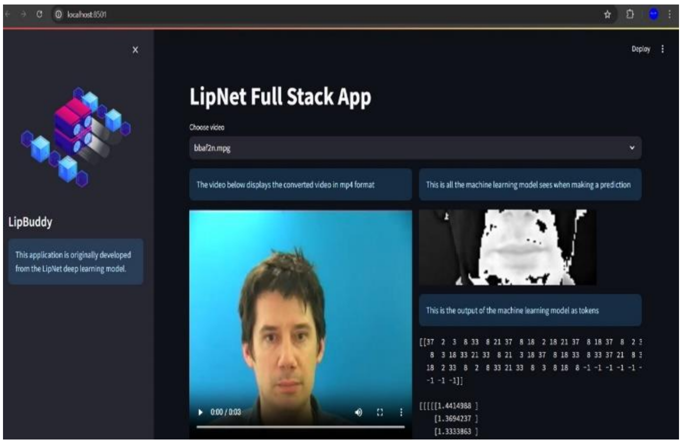
III. RESULTS AND DISCUSSION

Dataset - The dataset [15] has total 1300 gifs of single persons pronouncing digits from zero to nine. Six people are facing the camera and speak sentences like Blue, world,etc.. All the videos are recorded into the full-frontal pose. Each speaker speaks words what we speak in real world naturally.



```
[ ] sample = load_data(tf.convert_to_tensor('/content/data/s1/bbas1s.mpg'))
    print('~'*100, 'REAL TEXT')
    [tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in [sample[1]]]

    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ REAL TEXT
    [<tf.Tensor: shape=(), dtype=string, numpy=b'bin blue at s one soon'>]

    yhat = model.predict(tf.expand_dims(sample[0], axis=0))

    1/1 [==============================] - 0s 145ms/step

[ ] decoded = tf.keras.backend.ctc_decode(yhat, input_length=[75], greedy=True)[0][0].numpy()

[ ] print('~'*100, 'PREDICTIONS')
    [tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in decoded]

    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ PREDICTIONS
    [<tf.Tensor: shape=(), dtype=string, numpy=b'bin blue at s one soon'>]
```

Fig 6.Acutal and Predicted sentence of trained model



LipBuddy

This application is originally developed from the LipNet deep learning model.

The video below displays the converted video in mp4 format

This is all the machine learning model sees when making a prediction

This is the output of the machine learning model as tokens

```
[[37  2  3  8 33  8 21 37  8 18  2 18 21 37  8 18 37  8  2 3
   8  3 18 33 21 33  8 21  3 18 37  8 18 33  8 33 37 21  8 3
  18  2 33  8  2  8 33 21 33  8  3  8 18  8 -1 -1 -1 -1 -1 -
  -1 -1 -1]]

[[[[1.4414988 ]
   [1.3694237 ]
   [1.3333863 ]
```

Fig: Deployment of Lip Reading Model

After implementing the proposed methodology, the prediction results are obtained with following architectures and models:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv3d (Conv3D)             (None, 75, 46, 140, 128)  3584

 activation (Activation)     (None, 75, 46, 140, 128)  0

 max_pooling3d (MaxPooling3D  (None, 75, 23, 70, 128)  0
 )

 conv3d_1 (Conv3D)           (None, 75, 23, 70, 256)   884992

 activation_1 (Activation)   (None, 75, 23, 70, 256)   0

 max_pooling3d_1 (MaxPooling  (None, 75, 11, 35, 256)  0
 3D)

 conv3d_2 (Conv3D)           (None, 75, 11, 35, 75)    518475

 activation_2 (Activation)   (None, 75, 11, 35, 75)    0

 max_pooling3d_2 (MaxPooling  (None, 75, 5, 17, 75)    0
 3D)

 time_distributed (TimeDistr  (None, 75, 6375)         0
 ibuted)

 bidirectional (Bidirectiona  (None, 75, 256)          6660096
 l)

 dropout (Dropout)           (None, 75, 256)           0

 bidirectional_1 (Bidirectio  (None, 75, 256)          394240
 nal)

 dropout_1 (Dropout)         (None, 75, 256)           0

 dense (Dense)               (None, 75, 41)            10537

=================================================================
Total params: 8,471,924
Trainable params: 8,471,924
Non-trainable params: 0
```

The class-wise, subject-wise and overall performance is analyzed to gain more insights about the proposed methodology.

The mouth images and remove the redundant information, then calculate the center position of the mouth based on the coordinate points of the image boundary, denoted as $(x0,y0)(x0,y0)$. The width and height of the lip image are represented by w and h, respectively, $L1L1$ and $L2L2$ represent the left and right, upper and lower dividing lines surrounding the mouth, respectively. According to the following formula to calculate the bounding box of the mouth:

$$L1 = x0 \pm w2, L1 = x0 \pm w2,$$

(1)

$$L2 = \mathrm{y}0 \pm h2, L2 = \mathrm{y}0 \pm h2,$$

(2)

The first step in LSTM is making a decision to discard useless information from the cell state, which is accomplished by a decision called “the forget gate”. This gate reads $ht-1ht-1$ and $xtxt$, then it outputs a value in the [0,1] interval for each number in cell state $\mathrm{C}t-1\mathrm{C}t-1$. The calculation process is as follows:

$$ft = \sigma(Wf \cdot [ht-1,xt] + bf), ft = \sigma(Wf \cdot [ht-1,xt] + bf),$$

(3)

where $\sigma\sigma$ is the hidden activation function, $ht-1ht-1$ is the hidden state at time $t-1t-1$, $xtxt$ is the input at time tt, and bb is the bias.

Then, it is determined that new useful information is stored in the cell state. It consists of two parts: First, a sigmoid layer is called the “input gate layer” and it determines which value will be updated. Then a new candidate value vector is created by tanh, the activation function that processes the data on the state and output is tanh in LSTM. $\vec{\mathrm{C}}\, t\mathrm{C}\rightarrow t$ is added to the state, and the old cell state $\mathrm{C}t-1\mathrm{C}t-1$ is updated to $\mathrm{C}t\mathrm{C}t$. Second, the cell updates useful information into cell status and multiply the old cell state $\mathrm{C}t-1\mathrm{C}t-1$ and the output of “forget gate” $ftft$ as the part input of cell, then summing it with the product of “input gate” output $itit$ and candidate information $\vec{\mathrm{C}}\, t\mathrm{C}\rightarrow t$. The result of the calculation is the updated $\mathrm{C}t\mathrm{C}t$. This is the new candidate and it changes based on how much we decide to update each state. The calculation processes are as follows:

$$it = \sigma(Wi \cdot [ht-1,xt] + bi), it = \sigma(Wi \cdot [ht-1,xt] + bi),$$

(4)

$$\vec{C}\, t = \tanh(WC \cdot [ht-1,xt] + bC), C\rightarrow t = \tanh(WC \cdot [ht-1,xt] + bC)$$

,

(5)

$$Ct = ft * Ct-1 + \mathrm{i}t * \vec{\mathrm{C}}\, t, Ct = ft * Ct-1 + \mathrm{i}t * \mathrm{C}\rightarrow t,$$

(6)

Finally, the output value is determined based on the filtered cell state. Firstly, the sigmoid layer determines the output portion of the cell state. Then, the cell state is passed through tanh and multiplied by the output of the sigmoid layer to obtain the result of the cell. The value range of the sigmoid function is [0,1], which is most suitable for controlling the opening and closing of various doors. This part of the calculation processes is shown as follows:

$$ot = \sigma(Wo \cdot [ht-1,xt] + bo), \mathrm{o}t = \sigma(Wo \cdot [ht-1,xt] + bo),$$

(7)

$$\mathrm{h}t = \mathrm{o}t * \tanh(Ct), \mathrm{h}t = \mathrm{o}t * \tanh(Ct),$$

(8)

The key to the LSTM network is the cell state.The calculation process runs through the horizontal line. It runs directly across the chain with only a small amount of linear interaction and it will be easy to keep the information flowing on it.

The CNN is used as the encoder and the LSTM network is used as the decoder. In the decoding process, we introduce the attention mechanism and learn the attention weights $(\alpha\alpha)$, thus the model pays more attention to the effective area of the whole video [23]. The feature vectors for each frame are weighted and then all video frame sequences (vv) are simultaneously used as input $\phi(V)\phi(V)$ to the LSTM network. The input to the attention-based LSTM model is as follows:

$$\phi(V)=\sum{}^{n\mathrm{i}=1}\alpha_{ti}v_{i}, \phi(V)=\sum \mathrm{i}=1 n\alpha tivi,$$

(9)

The learning of weight $\alpha\alpha$ is related to the state of a hidden layer unit on the LSTM network and the feature vector of the current time. The correlation score of $\alpha_{ti}\alpha ti$ is as follows:

$$e_{ti}=\tanh(W \cdot h_{t-1}+U \cdot v_{i}+b), eti=\tanh(W \cdot ht-1+U \cdot vi+b),$$

(10)

where $h_{t-1}ht-1$ is the output of the hidden unit state at time $t-1t-1$, $v_{i}vi$ is the eigenvector of the video frame ii, and WW, UU, bb respectively represent the weight matrix to be learned and the offset parameters, the activation function is tanh. Normalization can be obtained as follows:

$$\alpha_{ti}=\exp(e_{ti})\sum{}^{\mathrm{n}k=1}\exp(e_{tk}), \alpha ti=\exp(eti)\sum k=1\mathrm{n}\exp(etk),$$

(11)

where $\alpha_{ti}\alpha ti$ represents the conditional probability $(P_{(a|e)}P(a|e))$ of the video feature vector of the video frame ii at time tt and the entire video feature vector, furthermore, $\sum_{nk=1}\alpha_{ti}=1\sum k=1nati=1$. The closer the relationship between the frame and whole video feature vector, the bigger the attention weight will become. Then the attention-based LSTM network input at time tt is as follows:

$$h_{t}=f_{rnn}(h_{t-1},\phi(V)), ht=frnn(ht-1,\phi(V)),$$

(12)

where $f_{rnn}frnn$ is a unit of LSTM, $h_{t-1}ht-1$ is the state of the hidden layer unit at time $t-1t-1$, and $\phi(V)\phi(V)$ is the input at time tt after increasing the attention weights.

Although the introduction of the attention mechanism will increase the amount of computation, it can selectively focus on the effective information in the video and reduce the interference of invalid information, thus the performance level of the network model can be significantly improved.

CONCLUSION

The primary objective of this project was to develop a system capable of accurately interpreting spoken words by analysing lip movements in a given video. Leveraging machine learning and image classification techniques, our approach has demonstrated notable success in recognizing and categorizing diverse speech patterns. By employing advanced deep learning models, we have created a robust lip reading system that can effectively translate visual speech into text, paving the way for enhanced accessibility and human-computer interaction.

FUTURE ENHANCEMENTS

In the future, lip reading using deep learning could focus on improving accuracy and robustness through multimodal approaches, integrating visual and audio data. Advanced architectures like transformers and attention mechanisms could better capture the temporal dynamics of lip movements. Transfer learning and large-scale datasets can enhance model generalization across diverse speakers and conditions. Real-time processing capabilities can be improved with optimized neural networks and edge computing. Additionally, incorporating contextual understanding and language models can refine interpretation accuracy. Ethical considerations, such as privacy and consent, will be crucial as these technologies become more sophisticated and widely used.

REFERENCES

[1] Lu, Y.; Li, H. “Automatic Lip-Reading System Based on Deep Convolutional Neural Network and Attention-Based Long Short- Term Memory,” Appl. Sci. 2019, 9, 1599. https://doi.org/10.3390/app9081599

[2] Patricia Scanlon , Richard Reilly and Philip de Chazal, “Visual Feature Analysis for Automatic Speech reading,” International Conference on Audio-Visual Speech Processing, September 2003

[3] Priyanka P. Kapkar and S. D. Bharkad, “Lip Feature Extraction and Movement Recognition Methods”, International Journal of Scientific & Technology Research, vol.8, August 2019.

[4] Oscar Koller, Sepehr Zargaran and Hermann Ney, “Re-Sign: Re-Aligned End-to-End Sequence Modeling with Deep Recurrent CNN- HMM,” Human Language Technology & Pattern Recognition Group RWTH Aachen University, Germany, 2017

[5] Tayyip Ozcan and Alper Basturk, "Lip Reading Using Convolutional Neural Networks with and without Pre-Trained Models," Balkan Journal of Electrical and Computer Engineering, Vol. 7, No. 2, April 2019.

[6] Amit Garg, Jonathan Noyola, Sameep Bagadia, “Lip reading using CNN and LSTM,” 2016

[7] Abiel Gutierrez and Zoe-Alanah Robert, "Lip Reading Word Classification", Stanford University, 2017

[8] Pooventhiran G, Sandeep A, Manthiravalli K, Harish D, Karthika and Renuka D, “Speaker-Independent Speech Recognition using Visual Features,” International Journal of Advanced Computer Science and Applications, Vol. 11, No. 11, 2020

[9] Themos Stafylakis and Georgios Tzimiropoulos, “Combining Residual Networks with LSTMs for Lip reading,” Computer Vision Laboratory University of Nottingham, UK, arXiv:1703.04105v4 [cs.CV]

[10] http://www2.cmp.uea.ac.uk/~bjt/avletters/

[11] https://sites.google.com/site/achrafbenhamadou/-datasets/miracl-vc1

[12] https://www.robots.ox.ac.uk/~vgg/data/lip_reading/lrw1.html

[13] Vahid Kazemi and Josephine Sullivan, “One Millisecond Face Alignment with an Ensemble of Regression Trees,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014

[14] Mahmood, Ammar et al. “Automatic Hierarchical Classification of Kelps Using Deep Residual Features.” Sensors (Basel, Switzerland) vol. 20,2 447. 13 Jan. 2020, doi:10.3390/s20020447

[15] https://drive.google.com/drive/folders/12rss_0g4TKD8dqjmBPwaWvLfLD9Zr38g?usp=drive_link