Silent voice: Harnessing Deep Learning for Lip-Reading in Bangla

Munia Shaheen
Department of Computer Science & Enginnering
BRAC University
Dhaka, Bangladesh
munia.shaheen@g.bracu.ac.bd

Akib Zabed Ifti
Department of Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
akib.Zabed.ifti@g.bracu.ac.bd

Ariful Hassan
Department of Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
ariful.hassan@g.bracu.ac.bd

Junaed Hossain
Department of Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
junaed.hossain@g.bracu.ac.bd

***Abstract*—Understanding speech just through lip movement is known as lipreading. It is a crucial component of interpersonal interactions. The majority of the previous initiatives attempted to address the English lipreading issue. However, our goal is to build a deep neural network for the Bangla language that can produce comprehensible speech from silent videos by capturing the speaker's lip movements. Even though there is research on this topic in various languages, Bangla does not currently have a study or a suitable corpus to conduct research. Hence, we created a dataset of 4000 videos where we selected 20 Bangla words and these words were pronounced by 65 different speakers. Then we implemented models based on CNN-RNN architecture. Two models LipNet and autoencoder-decoder were used in previous research and two custom models were implemented as a part of our experiments. Finally, Lip-Net exhibits a reasonable level of performance with an accuracy of 62%, while Auto Encoder-Decoder performs poorly with an accuracy of 49.65%. Custom Model-1 shows a substantial rise in accuracy with 70.86%, and Custom Conv-LSTM exhibits the best overall performance with a maximum accuracy of 76.24%.**

***Index Terms*—Lipreading. Deep learning, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Lip feature extraction, Lip region localization.**

I. Introduction

Lipreading is the technique of comprehending what someone is saying only by their lip movement. Humans have long been able to read lips manually. However, automating lipreading is a significant goal since human lipreading performance is weak and constrained. Initially, systems for reading lips were created using traditional machine-learning techniques. Deep learning applications' prominence, particularly in recent years, has caused this topic to be studied more than in the past. The main components of automated lip-reading include face recognition, lip localization, feature extraction, classifier training with corpus, and word/sentence recognition by lip movement.

Nevertheless, due to challenges in the extraction of spatiotemporal features, generalization across speakers, and environmental noise, lip-reading is an extremely difficult task. Additionally, there are a lot of homophones, which implies that the same lip movement can produce a variety of different characters. Furthermore, some of the phonemes are created inside the throat and mouth and cannot be identified by simply observing a speaker's lips. Also, lip-reading from video without audio or text data depends on a variety of factors, including lighting, recording distance, and the speaker's speaking style. The endeavor becomes more challenging by external noises, mumbling sounds, and guttural sounds. Therefore, lipreading is a challenging task for both humans and machines.

Even though lip reading is a challenging technique to process, it is essential in a variety of fields, including accessibility, noise-sensitive communication, human-computer interaction, multimodal communication, security, and numerous more fields. Lip reading and voice recognition technologies can greatly improve accessibility for the deaf. Additionally, in noisy situations where speech could be hard to hear, such as crowded public spaces or industrial settings, lip reading technology might improve communication. These technologies enable individuals to understand speech even in challenging acoustic environments, improving the effectiveness and precision of communication in commonplace applications. Recent years have seen an increase in enthusiasm for studies on the security of merging lipreading with biometric data, such as biological fingerprints or facial recognition. Studies that use lip movement monitoring for individuals include printing messages to smartphones in loud circumstances using visual data instead of auditory data and implementing various security measures utilizing visual silent passwords [1]–[3]. The applications of lip-reading technology are mentioned below:

- Helping hearing-impaired people.
- Transcribing old silent movies or videos.
- Facilitating the alert for public safety.
- Improving audio in existing videos.

- Enabling video conversation in noisy or silent places, like libraries.
- Identifying utilizing biometrics.
- Synthesizing simultaneous voice from multiple speakers.
- Enhancing automatic speech recognition's overall
- Dictating information or messages into a phone while surrounded

II. Literature Review

In automated lip reading, deep learning is needed as such tasks necessitate intensive preprocessing for an image or video frame extraction or other forms of manually created vision pipelines [4] [5]. However, deep learning is seldom used in lipreading and only performs classification, not sentence-level sequence prediction in most of the research.

The paper [6], is the first known sentence-level lipreading model that uses the GRID corpus [7] consisting of 1000 sentences of audio and video by 34 speakers. In this paper, the DLib face detector, iBug face landmark predictor [8], and affine transformation were used for preprocessing. Here, they introduced a new architecture called, LipNet. In the LipNet architecture, the input is processed by 3 layers of STCNN (Spatiotemporal convolutional neural networks) which is followed by a spatial max-pooling layer. Two Bi-GRUs then follow the retrieved features. Then at each time step, a linear transformation is applied before performing a softmax over the vocabulary augmented by the CTC (Connectionist Temporal Classification) blank, followed by the CTC loss. The authors evaluated the performance of the LipNet model using the word error rate (WER) and character error rate (CER). The LipNet model is the first sentence-level lip reading that gives 95.2% accuracy at the sentence level outperforming a human lipreading baseline and exhibiting better performance than the word-level state-of-the-art in the GRID corpus. However, this paper has some limitations, the model only works for sentences consisting of 6 words and to improve their research they need larger datasets.

The paper [9], used the same GRID corpus [7] used in the paper LipNet [6]. To begin with, they trained an autoencoder of audio files. A CNN-LSTM architecture is connected to a decoder of a pre-trained autoencoder, while the main network uses a 7-layer 3D convolutional structure to extract spatiotemporal information from the input video sequence. For evaluation, they compared their result with Vid2Speech [10] in terms of PESQ(Perceptual Evaluation of Speech Quality) [11], Corr2D, and STMI(Spectro Temporal Modulation Index) [12] and gave better results. This paper showed ways to improve the audio quality for better prediction with a 98% correlation. However, they only used the GRID dataset, which contained only 6 sentences with a small word probability, to evaluate their model. For future work, more train data needs to be gathered, speech reconstructions need to include emotions, and an end-to-end framework needs to be built.

The first sentence-level Chinese lipreading was introduced in the paper [13]. In this study, they created their own dataset from CCTV News and Logic Show, labeling it with Hanyu Pinyin (a phonetic interpretation of Chinese), and end up with 349 classes and 1705 characters. They present a unique two-step network for sentence-level Mandarin lipreading where predicting the probability of Hanyu Pinyin is the first step. A max-pooling layer is applied after a 3D convolution that suggests a series of frames as input. Then, more visual features are extracted using DenseNet. The visual features are processed using a two-layer resBi-LSTM, which is subsequently accompanied by linear and softmax layers. The complete network is then trained using the CTC loss function. The second phase involves using a stack of multi-head attention to translate Hanyu Pinyin into Chinese characters. The suggested network has an absolute 13.91% and 14.99% rise in Hanyu Pinyin and Chinese character accuracy compared to LipNet [6], whereas ResNet [14] technique has an increase of 4.68% and 4.74%. The problem that remains is that for different words Hanyu Pinyin might be the same, but Chinese characters would be different due to the different contexts which indicates that Hanyu Pinyin is unable to capture context.

In the paper [15], two new datasets were created for the Turkish language, one with 111 words and the other with 113 sentences. In the paper [15], the Media-pipe framework determines the lip position and removes it from the image. Using a convolutional neural network (CNN), features are first retrieved to perform lip detection and then the classification process is completed using bidirectional long short-term memory(Bi-LSTM). The results of the experiment demonstrate that, for words and sentences, respectively, ResNet-18 and Bi-LSTM pair offers the best results, with accuracy scores of 84.5% and 88.55%. However, the limitation of the paper is that each speaker is positioned at 1.5 m in these photos, and the video clips taken to generate the dataset were captured in identical lighting and ambient conditions.

III. Dataset Analysis

In the context of our research, data visualization played a vital role in providing a comprehensive understanding of the modular structure of our dataset. To demonstrate how the videos are distributed throughout the 20 distinct folders (each with a unique Bengali title). With each folder holding precisely 200 videos, this visualization demonstrated the distribution of our dataset in a simple and understandable manner. Bangla terms were used to label the folders, which not only enriched our dataset culturally but also demonstrated our dedication to maintaining linguistic variety in data gathering and annotation.

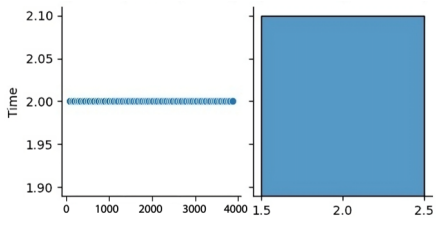

Fig. 1. Average length of videos

There are 4000 video samples and the average length of those videos is 2 seconds shown in Figure 7.

IV. Methodology

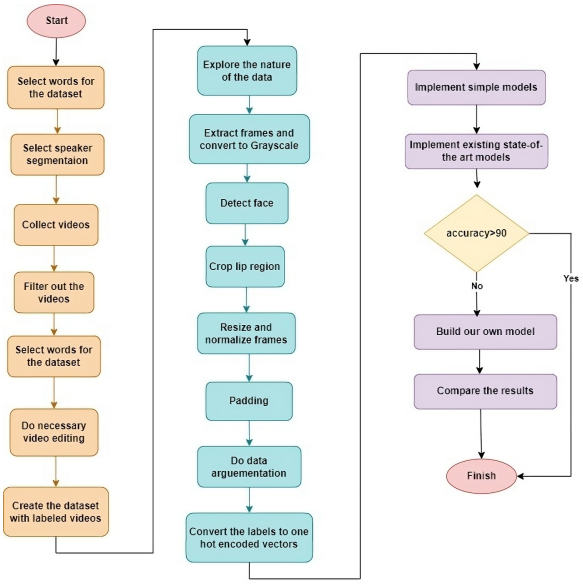

Fig. 2. Work Plan

A. *Data collection*

As there was no available video corpus for Bangla language lip reading, we have created our own corpus. The process of collecting video data from numerous individuals recording themselves speaking specific words typically involves several key steps. Firstly, a well-defined set of words or phrases is chosen for the participants to speak. Next, a diverse group of participants is recruited, ensuring variability in age, gender, and other relevant factors to capture a wide range of linguistic and vocal characteristics. For our data collection, we have selected 20 Bangla words and these words were pronounced by 65 different speakers.

B. *Data Preprocessing*

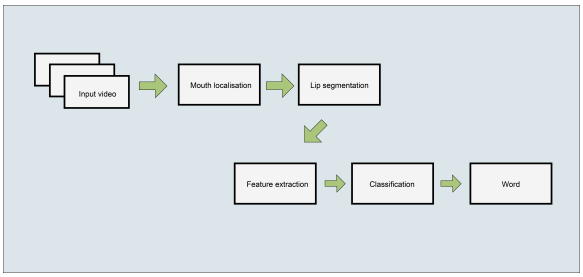

Fig. 3. Lip Reading Process

i. Video editing

When collecting videos from speakers, it's not uncommon to encounter variations in lighting conditions and video file sizes. Videos captured under poor lighting conditions can result in dark, grainy footage that hinders visual clarity. In such cases, video editing software can be employed to adjust brightness, contrast, and color balance to enhance the overall visual quality. This ensures that all videos in the dataset are visually consistent and suitable for analysis or presentation. Moreover, we managed the large-size video by doing some actions like trimming and cutting unnecessary parts of the video, such as long pauses or redundant sections, and irrelevant content of the video.

ii. Extract Frames and Grayscale

Lipreading is the process of identifying spoken words and phrases by evaluating the movement and form of the speaker's lips over time. Frames can be extracted such that the video can be divided into discrete time steps or moments, with each frame corresponding to a different lip visual configuration. Using a Python video processing library like OpenCV, we were able to turn a video into frames. First, we open the video file using the library's video capture function. Then, we read each frame sequentially from the video using a loop. Once we have a frame, we save it as an image file. By iterating through all the frames, we can effectively convert the video into a sequence of individual image frames. In our case, we took a constant number of 30 frames for each video. Along with converting videos into frames, we converted our videos into grayscale to simplify the data and reduce computational complexity, making it easier and faster to detect faces.

iii. Face Detection & Crop Lip Region

Lip reading focuses on the lip movement of a human face which requires only the lip region. Other than lip region - face, eyes, and expression are not required for lip reading. We removed all the unnecessary regions from the frames and cropped the lip region from the image. For this, we first detected the face and then cropped the lips. It begins by using the 'dlib' library to initialize a face landmark predictor and a frontal face detector. The 'face_detector', identifies the location of faces in the input frame.

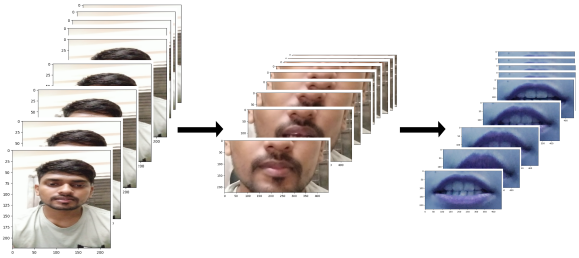

Fig. 4. Lip Segmentation

Once a face is detected, the 'landmark_predictor' is employed to obtain the facial landmarks, which are specific points on the face, including those around the lips. In this case, points 48 to 59 correspond to the lip landmarks. These lip landmarks' coordinates are extracted and used to

determine the position and size of the lip region within the face.

iv. Resize & Normalize Frames

The videos we have collected from the speaker are from different distances and angles from the camera. Some have uplifted the camera, and some have captured a little from the side. As a result, the lip regions have different sizes in every video or every frame. Hence, the cropped frames that have been collected are not equal in every frame or for every speaker. To solve this, all the images have been resized to a fixed height and width of 200,100.

When working with images, it's common to ensure that pixel values fall within a certain range so we used normalization. By dividing each pixel value for every frame in 255, we are rescaling the pixel values so that they are in the range [0, 1]. This is useful because many image processing and machine learning algorithms work better when pixel values are within this range.

v. Padding

Having a different number of frames in the dataset can affect the model's accuracy. It's important to maintain consistency in frame counts for training and testing to achieve more reliable and accurate results. Thus, we have set 30-frame standards for our video inputs.

If the frame length is more than 30, then it calculates how many frames should be added at regular intervals to create sequences of the desired length then it will add frames at least one frame at a time to maintain consistency. However, if the number of frames for a video is less than 30, it pads with duplicate frames.

vi. Data Augmentation

For complicated tasks like lip reading, we need an enormous amount of data. Thus we used data augmentation to create variations of the original video frames to enhance the diversity of the training dataset. For the augmentation part argumentation has been used which is a built-in library in Python. In our research, Albumentation has been applied. Here shifts in both the horizontal and vertical axes were applied to mimic different orientations, and also established a rotation range of 20 degrees to simulate different angles. Furthermore, providing the ability to flip the photographs horizontally, which broadens the range of viewpoints included in the training set. When these adjustments are used, the grayscale frames get subtle differences that successfully simulate various situations and viewpoints of the same subject.

vii. Convert the categorical values

In our dataset, the labels were the 20 words we selected for our classification. These labels were in categorical values and we converted categorical labels to one-hot encoded labels to make them suitable for training, ensuring compatibility, distinct class representation, and proper loss calculation.

C. Model Training:

For our model, we have used an architecture called CNN-RNN. It is a combination of CNN and RNN where the output of CNN is used as the input of RNN. We chose this architecture because it is ideal for applications like video classification because CNN helps extract spatial information and RNN helps extract temporal features to maintain sequence. We have implemented a total of four models, two based on previous best research models, and the other two as a part of our experiment.

Lipnet

Here we tried to implement the LipNet model on our Bangla corpus. Our model takes as input video data with dimensions (30, 100, 200, 1), which represents videos with 30 frames, each frame having a size of 100 x 200 pixels in grayscale. We used Keras, a high-level neural networks API, with a TensorFlow backend, for building and training the model. To implement this, we used 3 layers of time-distributed Conv2D with 2x2 max pooling. Then, the dropout layer is applied after each MaxPooling2D layer and helps to prevent overfitting by randomly setting a fraction (20%) of input units to zero during training. Then flattening layer converts the output from the convolutional layers into a 1D vector which is necessary before feeding the output into recurrent layers. The bidirectional GRU layers capture temporal dependencies in both forward and backward directions. A dense layer with 512 units and relu activation is then used which adds non-linearity to the features extracted by the previous layers.

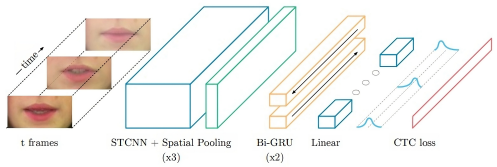

Fig. 5. LipNet Architecture [6]

Finally, the output layer with 'softmax' activation is applied. It produces class probabilities for multi-class classification (20 classes in this case). Each unit corresponds to a class, and the softmax function converts raw scores into probability scores. We used categorical cross-entropy as CTC loss was computationally expensive, and used when the number of input and output is unknown.

Autoencoder Decoder

Autoencoder-Decoder combines CNNs and LSTM networks and utilizes an auditory spectrogram for reconstructed speech. Here the output layer produces bottleneck features for the decoder. However, our research primarily focuses on correctly identifying the spoken words, not generating audio. Hence the decoder is not used in our case.

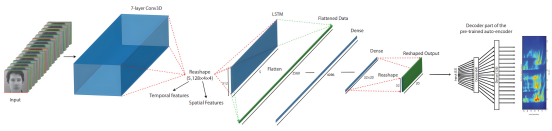

Fig. 6. Autoencoder-Decoder [10]

In our implemented model, there are 7 convolutional layers in this model. Each convolutional layer uses a 3D convolution operation (Conv3D). The kernel size for each convolutional layer is [3, 3, 3]. The number of filters starts at 32 and increases to 128 as we go deeper into the network. Batch normalization is applied after each convolutional layer to stabilize and speed up training. LeakyReLU activation is used after each convolutional layer except for the last one, where ELU (Exponential Linear Unit) activation is used. L2 regularization with a regularization strength of 0.0005 is applied to the convolutional layers. Max-pooling is performed in the spatial dimensions (width and height) with a pool size of (2, 2, 1), where the third dimension (depth) remains unchanged. After time-distributed flattening, there is a single LSTM layer, a dense (fully connected) layer with 512 units, and ReLU activation following the LSTM layer. The final layer is a dense layer with a softmax activation function. The number of units in this layer corresponds to the number of target classes, which is 20 in this case. Categorical cross-entropy is used as the loss function, which is appropriate for multi-class classification tasks. Moreover, the Adam optimizer is used with a learning rate of 0.0001.

Custom CNN-LSTM

Our first custom model follows CNN-RNN architecture similar to the LipNet model. The first layer in this model is a TimeDistributed(Conv2D) layer which applies a convolutional operation with a (3, 3) kernel size to each frame of the input video. Following the convolutional layer, there is a TimeDistributed MaxPooling2D layer with a (2, 2) pool size. This layer performs max-pooling for each frame separately, reducing spatial dimensions. Then TimeDistributed Flattening Layer flattens the output for input into the subsequent LSTM layers. A Dropout layer with a dropout rate of 0.5 follows, which helps prevent overfitting. Finally, a Dense layer with 256 units and ReLU activation follows, capturing high-level features from the LSTM output.

Custom Conv-LSTM

Our second custom model uses a variant of the Convolutional Long Short-Term Memory (ConvLSTM) network, an extension of the traditional LSTM model, which is adept at handling sequential data with spatial dimensions. The ConvLSTM layer was designed to concurrently handle the temporal and spatial correlations within the data. Both a ConvLSTM layer and a sequence of a convolutional layer followed by flattening and then an LSTM layer offer two different approaches to dealing with spatiotemporal data in neural networks. In ConvLSTM, the spatial information is preserved throughout the LSTM processing. This is particularly useful for tasks like video processing or any sequence prediction task where both the spatial features (shape, texture, etc.) of individual frames and the temporal dynamics between frames are important. Moreover, ConvLSTM layers are generally more parameter-efficient when dealing with high-dimensional input data, as they do not require the data to be flattened before processing. Consequently, in the Convolutional Layer followed by the Flattening and LSTM Layer, when the output of the convolutional layers is flattened, the spatial structure of the feature maps is lost. This means the LSTM layers do not have access to the original spatial relationships of the features.

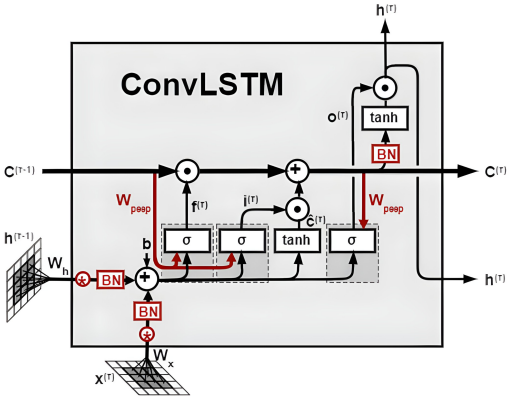

Fig. 7. Convolutional-LSTM architecture [16]

To implement this model, multiple ConvLSTM2D layers with varying numbers of filters (32, 64, 128, 256) and kernel sizes (3, 3) are used. The same padding and the tanh activation function are applied. The TimeDistributed layer is used with MaxPooling2D layers following a few ConvLSTM2D layers. For lowering the number of parameters and the spatial dimensions (height and width) of the model, this combination allows the model to apply max pooling to each frame individually, contributing in the control of overfitting. Here in this model, the GlobalAveragePooling3D layer lowers the output's dimensionality from the earlier layers. The model then consists of 512 and 1024 unit fully connected (Dense) layers, each of these layers followed by a Dropout layer with a 0.5 dropout rate. The final Dense layer indicates that the model is designed for a twenty-class classification problem since it contains three units with a softmax activation function.

V. Result Analysis

LipNet

The figure-3 indicates the 'Model Accuracy' for the Lipnet architecture, where the training accuracy is represented by the blue line. This line climbs significantly across the epochs, showing better performance on the training set. On the other hand, the orange line represents the validation accuracy, which shows a slower rate of rise and notable volatility, suggesting less steady performance on the validation set.

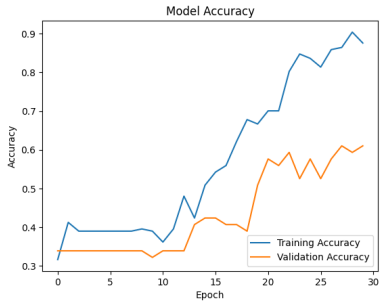

Fig. 8. Evaluation graph for Lip-Net

Autoencoder Decoder

In fig-4 the training accuracy is steadily increasing, as seen by the 'Model Accuracy' graph, indicating that the Auto Encoder-Decoder model gets better with time on the training set. Even though it is getting better, the validation accuracy is inconsistent and typically lower than the training accuracy, which suggests that the model might not be effectively generalizing to new data.

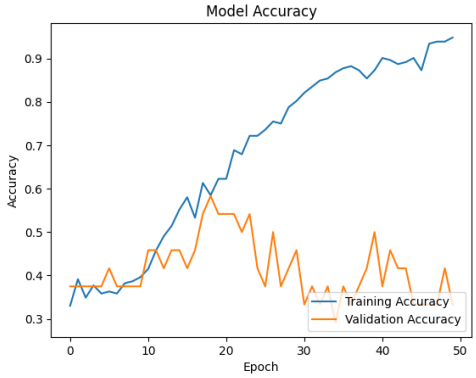

Fig. 9. Evaluation graph for Lip-Net

Custom CNN-LSTM

In fig-5 The 'Model Accuracy' graph shows that validation accuracy is increasing as well, but it is doing so with more instability. Training accuracy is increasing consistently,

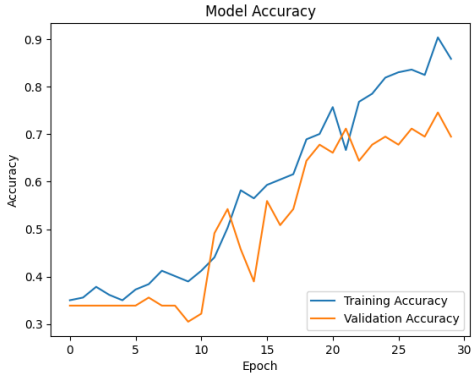

Fig. 10. Evaluation graph for CM-1

Custom Conv-LSTM

Fig-6 shows evaluation graphs of the 2nd Custom model. In the "Model Accuracy" graph, training accuracy steadily increases over 40 epochs. Validation accuracy, on the other hand, fluctuates, which may indicate that the model is overfitting or that more effective generalization methods are required.

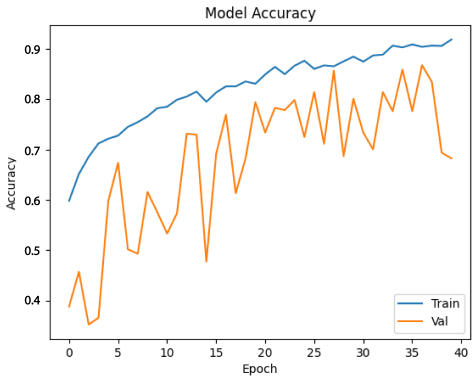

Fig. 11. Evaluation graph for CM-2

VI. Model Comparison

Various categorization reports were produced in this research to assess the performance of the suggested model. These reports provide a thorough evaluation by using multiple metrics. Accuracy shows the percentage of all predictions that have been generated correctly. It gives a clear and concise indicator of the model's accuracy rate for every class. Whereas, the F1 score is the harmonic mean of recall and accuracy, where recall is the ratio of all actual positives (including false negatives) to all true positives and precision is the ratio of true positives to all positives (including false positives).

Proposed Architecture	Accuracy	F1-Score	Top-1	Top-10
Lip-Net	62%	64%	62%	88.02%
Auto Encoder-Decoder	49.65%	46%	49.65%	66%
Custom Model 1	70.86%	77%	70.86%	95.89%
Custom Conv-LSTM	76.24%	78.11%	76.24%	98.88%

Fig. 12. Model Comparison Table

The fig-22 displays the comparison between various architectures that have been proposed for lip-reading Bangla words. These designs are assessed using multiple criteria, including accuracy, F1-Score, top-1 accuracy, and top-10 accuracy. Firstly, Lip-Net performs moderately, exhibiting a 62% accuracy and a comparable Top-1 Accuracy, indicating that it successfully selects the correct word 62% of the time. With a somewhat higher F1-Score of 64%, it shows a respectable balance between recall and precision. The Top-10 Accuracy is significantly higher at 88.02%, meaning that 88.02% of the time the correct word is among the top 10 predictions made by the model. Secondly among all the 4 architectures,

Auto Encoder-Decoder performs poorly, with an accuracy of 49.65%, matching F1-Score and Top-1 Accuracy. Though still less than other models, the Top-10 Accuracy is 66%, which is comparatively better than its Top-1 Accuracy. Again, with an accuracy of 70.86% and the highest F1-Score of 77%, Custom Early-Fusion shows a notable increase. This indicates a solid balance between recall and precision and, consequently, an efficient classification of the correct word. Additionally high are the Top-1 and Top-10 Accuracies, particularly the latter at 95.89%, which shows that the right word is nearly always among the top 10 guesses. FInally, with an F1-Score of 78.11% and the maximum accuracy of 76.24%, Custom Conv-LSTM performs the best overall. Its Top-1 Accuracy is consistent with the overall accuracy, and its remarkable Top-10 Accuracy of 98.88% is the greatest of all the architectures and indicates that the correct word is almost always among the first 10 predictions.

VII. Conclusion

In conclusion, lipreading is the skill of understanding a speaker's words solely from the movement of their lips which is crucial in many different disciplines like accessibility, noise-sensitive communication, human-computer interaction, multimodal communication, security, and a great deal more. Despite the fact that the English language has been the subject of numerous studies in this area, the Bangla language has not yet been the subject of any studies. CNN, Bi-LSTM, Bi-GRU, and CTC were frequently employed for English lipreading. However, after reviewing some studies, we discovered that the lipreading models utilized for English were inadequate for other languages. Therefore, it is imperative that we carry out research on Bangla lip-reading. However, there isn't yet a corpus of Bangla that can be used for lipreading. So, for our research a small video corpus has been created, where at most 200 videos for all the 20 words collected from different speakers. Then after necessary preprocessing of the dataset 4 different models have been applied into the features of every video keeping their labels aside which are basically the target variables. Finally, one of the custom models stands out among the others in terms of the accuracy and correctly classified 78% each time it has been executed into our test data.
However, due to lack of sufficient videos in the dataset the model performance were below 90%. Hence, in our future work, we will keep increasing the Bangla video corpus for greater accuracy.

References

[1] T. Afrouas, J. Chung, and A. Zisserman, "Deep lip reading: a comparison of models and an online application," in *ArXiv*, 2018.

[2] S. Chung and A. Zisserman, "Lip reading in the wild," in *Asian Conference on Computer Vision*, 2016.

[3] M. Miled, M. Messaoud, and A. Bouzid, "Lip reading of words with lip segmentation and deep learning," *Multimedia Tools and Applications*, vol. 82, no. 1, pp. 551–571, 2022.

[4] T. F. Matthews, J. A. Cootes, J. Bangham, S. Cox, and R. Harvey, "Extraction of visual features for lipreading," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 198–213, 2002.

[5] M. Zhao, M. Barnard, and M. Pietikainen, "Lipreading with local spatiotemporal descriptors," *IEEE Transactions on Multimedia*, vol. 11, no. 7, pp. 1254–1265, 2009.

[6] Y. M. Assael, "Lipnet: End-to-end sentence-level lipreading," *arXiv preprint arXiv:1611.01599*, 2016.

[7] J. Cooke, S. Barker, D. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006.

[8] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397–403.

[9] H. Akbari, "Lip2audspec: Speech reconstruction from silent lip movements," *arXiv preprint arXiv:1710.09798*, 2017.

[10] A. Ephrat and S. Peleg, "Vid2speech: Speech reconstruction from silent video," *arXiv preprint arXiv:1701.00495*, 2017.

[11] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (pesq), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs," ITU-T Recommendation, Tech. Rep., 2001.

[12] M. Elhilali, T. Chi, and S. Shamma, "A spectrotemporal modulation index (stmi) for assessment of speech intelligibility," *Speech communication*, vol. 41, no. 2, pp. 331–348, 2003.

[13] X. Chen, J. Du, and H. Zhang, "Lipreading with densenet and resbi-lstm," *Signal, Image and Video Processing*, vol. 14, no. 5, pp. 981–989, 2020.

[14] T. Stafylakis and G. Tzimiropoulos, "Combining residual networks with lstms for lipreading," in *conference of the international speech communication association*, 2017, pp. 3652–3656.

[15] U. Atila and F. D. Sabaz, "Turkish lip-reading using bi-lstm and deep learning models," *Engineering Science and Technology, an International Journal*, vol. 35, p. 101206, 2022.

[16] Author(s). (Year) An introduction to convlstm. Retrieved from Medium. [Online]. Available: https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7