

Bengali Word Detection from Lip Movements Using Mask RCNN and Generalized Linear Model

Abul Bashar Bhuiyan¹, Jia Uddin²

¹Department of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh

²AI and Big Data Department, Endicott College, Woosong University, Daejeon, Korea

Article Info

Article History:

Received Sep 10, 2023

Revised May 24, 2024

Accepted Jun 28, 2024

Keywords:

Word Detection

Lip Movements

Machine learning

Image Segmentation

Accuracy

ABSTRACT

Speech processing with the help of lip detection and lip reading is an advancing field. For this, we need proper algorithms and techniques to detect lips and movements of lips perfectly. Lip detection and configuration are the most important parts of speech recognition. In this paper, we focus on detecting the lip segment properly. Mask R-CNN (Regional Convolutional Neural Network) performs object detection and instance segmentation per video frame to detect the lip segment. The process of mask R-CNN adds only a small overhead to Faster R-CNN and is quite simple to train, running at 5 frames per second. The Mask R-CNN involves keypoint detection which helps to extract the location of the lip landmarks pixel by pixel. Once the lip region is extracted and the landmarks are highlighted, we observe how the lip landmarks change as the object's lips move over time to each Bengali word. The keypoint changes that are observed during each millisecond are then the landmarks used to train the GLM (Generalized Linear Model). In addition, we compare the performance of GLM with Naive Bayes, Logistic Regression, and Decision Tree. The GLM has exhibited the highest 91.8% accuracy, whereas the Naive Bayes, Logistic Regression, and Decision Tree show the accuracy of 87.1%, 38.3%, and 82.2%, respectively.

Copyright © 2024 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Jia Uddin, Ph.D.

AI and Big Data Department,

Endicott College, Woosong University, Daejeon, South Korea

Email: jia.uddin@wsu.ac.kr; Fax: 070-7545-9767

1. INTRODUCTION

Lip detection, a major focus in computer vision, is gaining momentum due to its integration with AI (Artificial Intelligence) and Machine Learning [1]. For facial expression recognition, real-time face and lips tracking is needed. It is a difficult task as the lip's movement varies from person to person. With the advancement of AI, like other applications, researchers are using deep learning models in lip movement recognition. Oliver *et al.* [2] introduced the LAFTER: lips and face real-time tracker for recognizing the potential of 2-D blob features for tracking facial expressions, a domain parallel to the detection and recognition of lip movements vital for precise speech recognition. Chan *et al.* [3] made strides in visual speech recognition, relying on CNN architectures, particularly the VGG-M model.

In [4], Aripin and Setiawan delved into Indonesian lip-reading recognition utilizing the LSTM (Long-Term Recurrent Convolutional Network). They emphasized the importance of carefully detecting lip movements to improve word accuracy. El-Bialy *et al.* brought forth a phoneme-based lip-reading system for silent speech recognition, again emphasizing the importance of accurate lip tracking [5].

It's important to understand the unique lip movements for different languages, as shown by Fu *et al.* in their study on Chinese lip-reading [6]. In another study, a CNN followed by a bidirectional LSTM and classifier achieved an accuracy of 84.75% for Bengali word detection from lip movement [7].

The proposed work is inspired by these foundational works and others like the work of Zhang and Lu [8] on lip-reading algorithms and Berkol *et al.* [9] who introduced a visual lip-reading dataset for Turkish. In Table 1, we show the contributions and limitations of different studies in the literature.

We used a hybrid approach for lip movement detection and word recognition. For detecting lip edges and keypoints, we extracted frames from videos and each frame process through an image-based method using Mask R-CNN due to its strong performance in object detection and segmentation and its computational efficiency. The word detection was video-based, where the Generalized Linear Model (GLM) analyzed the sequence of lip positions over time to identify words. This approach focuses on Bengali lip movement detection with Mask R-CNN, followed by word recognition using the GLM, combining the strengths of both image and video-based techniques.

Table 1. Contributions and limitations of different studies in the literature

Ref.	Contributions	Limitations
[3]	Low-resource method for lip movement detection from video.	Low accuracy, poor detection of diverse lip patterns, colors and sizes.
[4]	Developed a high-accuracy lip-reading system with LRCN, using HOG+SVM for mouth region detection.	HOG+SVM struggles with detailed lip edge detection and variations in lip shapes.
[5]	Improved lip-reading accuracy in noisy environments using phoneme-based classification.	Did not focus on detailed word-level recognition or individual lip movement analysis.
[6]	Expanded Chinese lip-reading dataset and introduced a highly efficient model with ShuffleNet and CBAM, reducing computational costs. Designed for mobile applications.	Did not mention about non-Chinese languages. Lacks generalizability to other languages and comprehensive model comparisons.
[7]	Achieved 84.75% accuracy in Bengali word detection with CNN-BiLSTM.	Low accuracy and not portable enough to train different languages easily.
[8]	Developed Efficient-GhostNet, a lightweight network for lip reading, reducing computational complexity and memory usage and the dataset variation is impressive.	Accuracy limited to 88.8% for numbers only, no training or testing on words and lack of consideration for different languages.
[10]	Detected hand gestures based on color that can be used to lip detection, achieving 86% accuracy.	Color based detection is not reliable for lip movement detection.

The advent of deep learning methods such as RCNN has brought about significant improvements in word detection using lip movements [11]. In this study, we use the current state-of-the-art, Mask RCNN [12], to detect lips and their keypoints [13], which aids in word detection from lip movements.

It aims to provide an alternative to sign languages and assist in various applications, promising improved accuracy and speed in visual speech-to-word conversion. This paper proposes an application of Mask RCNN for detecting and interpreting lip movements and then classifying the Bengali words using GLM, which can potentially be extended to other languages given appropriate datasets.

The rest of the paper is organized as follows. Section 2 describes the model and datasets. Section 3 discusses the Mask RCNN application and training and evaluates the results, and finally, Section 4 concludes the paper with limitations and future implementations.

2. PROPOSED METHODOLOGY

The detailed methodology of the model is described in the following sub-section and a flow diagram is presented in Fig. 1.

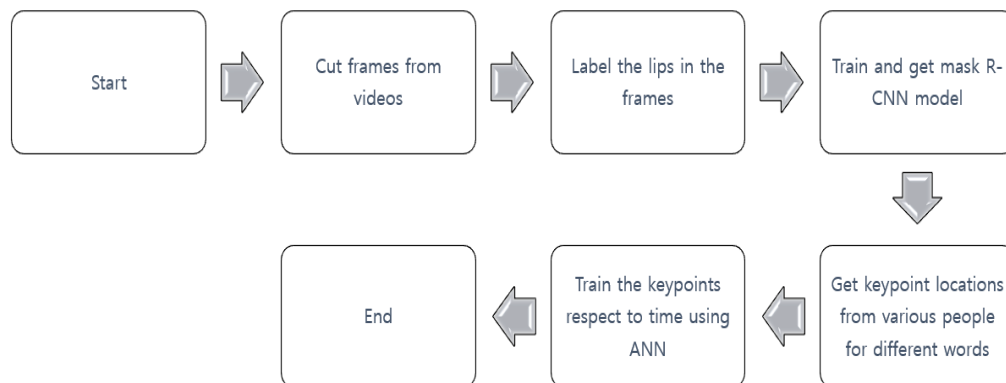


Figure 1. Step-by-step flow diagram of the proposed methodology.

2.1. Dataset Creation for Lip Detection

The dataset is composed of 800 images for training, some of the samples are shown in Fig. 2. These images are specifically used for training purposes. We have created an in-house video database from speeches recorded. The videos are further processed to make them suitable for the system.



Figure 2. A few samples of the used dataset.

2.2. Video Acquisition and Conversion

Videos of different personalities are taken ranging from male to female, young to old, and student to worker. The videos are mostly 3-5 seconds each. In each of the videos “Amar Bhasha Bangla/ আমার ভাষা বাংলা” is uttered. Here the lip movements are recorded mainly. There are different lip movements for different words and syllables. For the part “aaa/আ” from the word “Amar/ আমার”, the upper lips and lower lips move further apart. Again, for the “mar/মার” part, the distance between the upper lips and lower lips decreases at one stage and becomes zero. Similarly, for other words, there are different lip movements. Each lip movement is vital for calculations since the position of the lips and the distance between the upper lips and lower lips are our main concerns. The videos were taken in normal condition in house rooms using a mobile camera. After the videos are successfully taken, they are converted to images in jpg format. For this purpose, we have used a “Free video to jpg converter” [14]. This converts videos of all formats to jpg image format. Here basically frames are cut from the videos to turn them into a series of sequential images. The videos were usually 4-5 seconds. For each video, we have taken frames after 200 milliseconds. So, for each second of video, we get 5 frames. Like this, a total of 1000 image frames are taken from several videos.

2.3. Data Refinement

The frames were resized to focus on the region of interest (ROI) [15] and then annotated to identify keypoints and masks. We have taken data from people of various ages, 10 to 60 to ensure realistic and diverse results. A detailed breakdown of frames sourced from different age groups is shown in Table 2. All these frames are taken for training purposes. For testing purposes, we have taken another group of frames as shown in Table 3. This part is for testing. We have taken 118 images for testing purposes.

Table 2. Classification of Training Frames

Age Group	Number of Frames		Number of People	
	Male	Female	Male	Female
10-20	415	0	20	8
21-30	147	0	8	2
31-40	0	40	1	2
41-50	0	88	4	2
51-60	110	0	4	3

Table 3. Classification of Testing Frames

Age Group	Number of Frames	
	Male	Female
10-20	60	0
21-30	25	0
31-40	0	6
41-50	0	11
51-60	16	0

2.4. Frame Resizing

The frames that were cut from the videos are of the video resolution. The resolution is very high and it covers areas that are unnecessary for our calculations. As shown in Fig. 3(a), we have resized the frames into 512 x 512 pixels to cover only the region on which our research is based on or the region of interest. This operation is performed with the help of BIRME 2.0 [16]. That is Bulk Image Resizing Made Easy. The version is 2.0 and it is an online tool to resize images.

Here, we can see only the face part is taken into consideration and other parts are ignored. We perform the annotations on the resized 512 x 512 frames. The quality of the images is set to 100% and the format of the images or frames is not changed here. These resized frames are then sent for annotation.

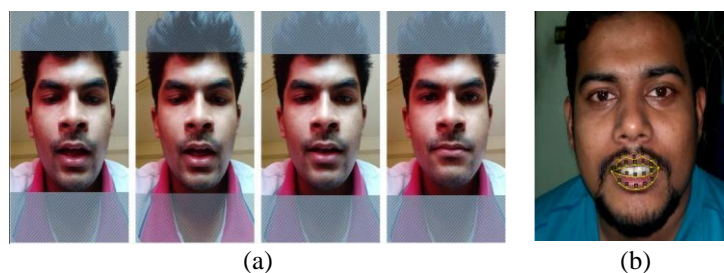


Figure 3. (a) Frame resizing, (b) Annotated image with keypoints and mask.

2.5. Annotations

The most vital part of lip movement detection is the annotation part. Here we determined the key points from the lips and then detected the mask. The change in the position of the key points determines the lip movements and then we detect words from it. For this process, we have used a tool called VGG Image Annotator (VIA) [17].

Now, first, the image frames are imported to the tool. Then we take small points on the lips since it is our region of interest. For the upper lip, the key points are taken from the leftmost side of the lip. That is keypoint 1. Then gradually keypoints are taken along the edges of the upper part of the upper lip. It stops at keypoint 7. Next, we continue the process on right most edges of the lower part of the upper lip. We stop after we have reached keypoint 12.

The key points are taken from the leftmost side of the upper lip. That is keypoint 1. Then gradually keypoints are taken along the edges of the upper part of the upper lip. It stops at keypoint 7. Next, we continue the process on right most edges of the lower part of the upper lip. We stop after we have reached keypoint 12.

So, in total 22 key points are used for the upper and lower lip. Here one of the most important parts is the vertical alignment of the points. Keypoint 1 and keypoint 7 are independent points. Keypoint (2,12,13,22) are vertically aligned. Similarly (3,11,14,21), (4,10,15,20), (5,9,16,19), and (6,8,17,18) have aligned points in their groups. All the four key points in a group fall on the same straight line.

After we are done with the keypoints we draw a polygon joining the key points of the upper lips first, as shown in Fig. 3(b). Then another polygon is drawn joining the key points of the lower lips. So, after masking we have a total of 22 key points to work with. When a speech is made the key points change their position and we identify what the person is saying.

After the annotation is done we exported the annotations of the images as JSON files. This file is used as a dataset for the process.

2.6. Further Data Collection

In our second dataset, we have taken a total of 3000 videos. Each video is 1-2 seconds. Out of 3000, 1000 is taken for the word “amar/ আমার”, 1000 is taken for “bhasha/ ভাষা” and the last 1000 is for the word “bangla/ বাংলা”. After the videos are taken they are converted to frames in the same process as the first dataset.



Figure 4. Image frames to create the second dataset

As shown in Fig. 4, a sequence of images was taken for the word “bhasha/ ভাষা”. We detected keypoints from these images and calculated the ratio of the length between points with respect to the length of the two corners of the lips and then these are sent for training so that the system can detect the lip movements more.

3. RESULTS AND DISCUSSION

The model was trained using Matterport's Mask R-CNN implementation [18], customizing it for the image dataset. We loaded our image dataset with annotations and added them to the custom dataset class. During the training, we observed almost a consistent decrease in loss values but keypoint loss was not very satisfactory. We had loss values for Keypoints, Bounding Box, Class, Mask, RPN Bounding Box, and RPN Class and all of them sum up the Total loss as shown in Table 4. As shown in Fig. 5(a) and in Table 4 all the loss values are described below.

Total Loss: After the very first epoch the total loss value starts with the value of 8.084 and it smoothly goes down after each further epoch is completed. After completing 150 epochs the loss value then goes almost flat with no major variation and it ends with the value of 2.658.

Keypoints Loss: At the time of training, initially our keypoint loss function value was 7.110 and after completing all the 160 epochs the results ended with a value of 2.611. The value shows that, gradually our keypoint loss function value is decreasing as per epoch for the training model we predict that the model is performing well for keypoint at the moment of training.

Bounding Box Loss: Training in our mask RCNN model bounding box loss function gives a very happy figure. It starts with the value of 0.2935 and after around 10 epochs the value decreases to around 0.0400. Then for the next 10 epochs, the loss function graph is a curve which means, at this stage of training in our model the dataset is performing well for the bounding box. And at the last, we get the bounding box loss of $3.8988e^{-3}$.

Table 4. Training loss values

Loss Properties	Loss Values	
	Epoch 1	Epoch 160
Total	8.084	2.658
Keypoints	7.110	2.611
Bounding Box	0.2935	$3.8988e^{-3}$
Class	0.04928	$1.4923e^{-3}$
Mask	0.2886	0.03923
RPN Bounding Box	0.3303	$2.1834e^{-3}$
RPN Class	0.01287	$1.5984e^{-4}$

Class Loss: The function value is also in a good figure in our model with a start of value 0.04928 and after 160 epochs completed the value is $1.4923e^{-3}$.

Mask Loss: At the point of starting the mask loss value is 0.2886 and up to 5 epochs the convergence of the loss function value is almost the same the value goes down by each epoch and at the end of 160 epochs it is 0.03923.

RPN Bounding Box Loss: Region Proposal Network (RPN) bounding box loss function value also shows an initial value of 0.3303 and for the first few epochs the value of the loss function is almost the same. From 40 to 120 epochs, the loss function value stands around at 0.0500. From the graph shown in Fig. 5(a), it shows that after 120 epochs RPN bounding box loss further decreased and at the last epoch the value was $2.1834e^{-3}$.

RPN Class Loss: In our model, the RPN class loss function shows a very good figure with an initial value of 0.01287 and it ends with $1.5984e^{-4}$.

The model was trained on an Intel Core i5 8400 processor with 8GB of RAM and an NVIDIA GTX 1080 graphics card. It was trained for 160 epochs and took about 33 hours to train.

3.1. Model Validation

As shown in Table 5, model validation showed satisfactory progress for all the properties except keypoint loss, as expected. The graph indicates that more data were needed for keypoint detection, as shown in Fig. 5(b) and Fig. 5(c).

Loss Properties	Loss Values	
	Epoch 1	Epoch 160
Total	7.648	4.988
Keypoints	6.683	4.666
Bounding Box	0.1988	0.03091
Class	0.08238	1.4975e ⁻⁴
Mask	0.2268	0.2155
RPN Bounding Box	0.4509	0.07436
RPN Class	6.3355e ⁻³	8.4359e ⁻⁴

Total Validation Loss: It is quite high at the beginning of epoch 1 which is 7.648 and at the end, it is 4.988 which is still quite high and it should be less than 1 for better results. The reason behind this high value is that the system could not detect the key points well but managed to detect the mask pretty well.

Keypoint Validation Loss: While validating our model, the keypoint loss did not show a good figure. After 120 epochs are completed the loss function value goes up. This is because we needed more datasets for training the model. At the point of validating our model, we see the model won't perform well for keypoint detection, we need to add more images to our dataset to overcome the overfitting issue.

Bounding Box Validation Loss: This is the validation bounding box loss graph. The value after the first epoch is 0.1988 and at the end of the 160 epochs, the value reduces to 0.03091. There is a great reduction of value at the end. This means that the system could determine the bounding boxes on the lips quite well and the difference between training and testing is very negligible.

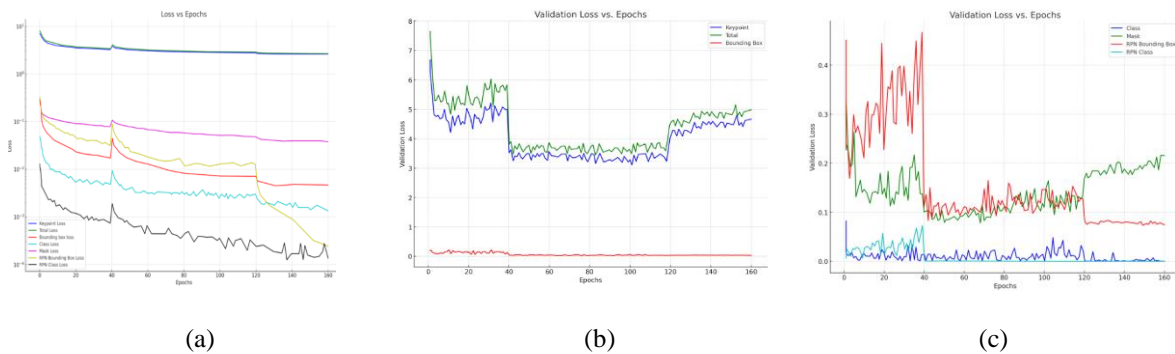


Figure 5. Visualization of validation loss: (a) over epochs-1, (b) over epochs-2, (c) over epochs (Logarithmic scale) Equations.

As shown in Fig. 5(b), Fig. 5(c) and in Table 5 all the validation loss values are described below.

Class Validation Loss: On the first epoch of class validation loss, the value is 0.08238 and at the end of the 160 epochs, the value reduces to 1.4975e⁻⁴. So, the value reduces a lot. That means that the training is done well and when the system is sent for testing, it can match the lips and detect it very well.

Mask Validation Loss: Here we can see that on the first epoch, the value is 0.2268 and at the end of 160 epochs we see it is 0.2155. Here on the graph, we see that it reduces and then rises again. This doesn't mean mask detection will be poor, in terms of mask validation the loss value will vary because there is no proper way to validate it. For the same image of lips, the model will generate different masks every time we run the model. As the value was always less than 1, we are good to go.

RPN Bounding Box Validation Loss: It validates whether the trained system can detect boxes or not. Here we can see that on our first epoch, the value is 0.4509 which is high but after 160 epochs the value is reduced to 0.07436 which is very close to 0. This indicates that the trained system is able to detect the boxes of the lips.

RPN Class Validation Loss: This is the Region Proposal Network. It is of Faster RCNN and since our system is an extended version of the Faster RCNN this is also a part of our loss validation. It is mainly the difference when we have worked with the mask and without the mask. Here we can see that on the first epoch, the value was comparatively high but after 160 epochs the value reduced a lot. This indicates that the detection was much better after 160 epochs and the system is getting trained well.

3.2. Image Testing and Predictions

Based on the data the keypoint loss value was not significantly reduced and keypoint detection was not highly accurate, as shown in Fig. 6(a). So, we decided not to use keypoint detection from the trained model. Instead, we used the Mask R-CNN model to output a binary matrix for each class detected.

3.3. Solving Keypoint Detection Issue

When we run the Mask R-CNN model to a frame or picture it returns a binary matrix for each class detected where each index of the matrix is used as a pixel value. If any index contains '0' that means the detected object's pixel is not presented there and if the index contains '1' that means the object's pixel is presented there.

If we consider a 512x512 image where we want to detect the lips we will get a binary matrix for each class. We have classified the upper lip and the lower lip differently. For the upper lip, we will get a binary matrix of size 512x512, as shown in Fig. 6(b), and the same for the lower lip.

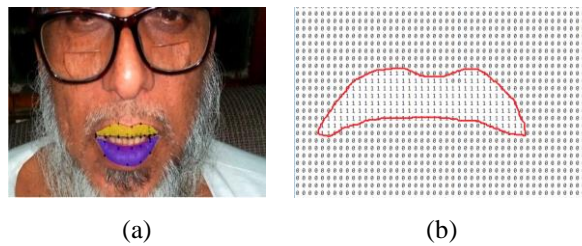


Figure 6. (a) Middle point detection from the midpoint of two top points, (b) Upper lip detection mask.

We detected the columns where there are no '0's. So, the length of the lip will be the distance from the first non-zero column to the last non-zero column as shown in Fig. 7(a). This distance will help us to detect the rotation of the lip. For distance calculation, we used a Geometrical Equation (1).

$$\text{distance} = \sqrt{(C_2 - C_1)^2 + (R_2 - R_1)^2} \quad (1)$$

Here, C_1 = First Column Index, C_2 = Last Column Index, R_1 = First Row Index, and R_2 = Last Row Index.

We find the midpoint of the lip using midpoint equation (2) and split the lip into two parts. From the two parts we detected the top points, the 'left top point' and the 'right top point' as shown in Fig. 7(a).

$$\text{midpoint} = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right) \quad (2)$$



Figure 7. (a) Upper Lip Detection Mask (b) Middle Point detection from the midpoint of two top points, (c) Additional keypoint between corner and top point, (d)

After getting two top points we again used the midpoint equation (2) and the first non-zero row of that column to get the 'mid-top point' of the lip, as shown in Fig. 7(b). We again used the midpoint equation (2) to get the midpoint between 'top points' and 'corner point', as shown in Fig. 7(c) and did the same for both the left part and right part of the lip. Also, in Fig. 7(d) it is shown that the lips are not always perfectly aligned with the axis so we calculated the angle between the lip and the axis for lip alignment using the Trigonometric function (3).

$$\theta = \tan^{-1} \left(\frac{y_2 - y_1}{x_2 - x_1} \right) \quad (3)$$

$$x_{\text{next}} = x_{\text{present}} + \cos\theta \times \text{length} \quad (4)$$

$$y_{next} = y_{present} + \sin\theta \times length \quad (5)$$

Based on the calculated angle we found other keypoints coordinates in the matrix using these equations (4), and (5), and as a result, we got perfectly located 22 keypoints on both the upper and lower lips. We used these keypoints to calculate the distance per frame between the points of the lips when the lips move. Then we calculated the ratio of the distance with respect to the distance from one corner to another corner and stored the ratio value per frame in a spreadsheet. We took 5 frames per second (200ms) per word and calculated each frame's distance ratios and fitted them in some Machine Learning algorithms for prediction.

3.4. Algorithms for the Second Dataset

We used supervised learning [19] to train the model with our own image dataset and predict words. The algorithms used included Logistic Regression [20], Generalized Linear Model [21], Decision Tree [22], and Naive Bayes [23].

3.5. Result Analysis

The comparative performance of four distinct algorithms was evaluated in our research. We have represented the comparisons visually using bar graphs as shown in Fig. 8, revealing a correlation range between 10% and 50%.

As shown in Fig. 8, the bar graph shows the respective accuracy levels of the utilized algorithms. Naive Bayes, Generalized Linear Model, Logistic Regression, and Decision Tree, with accuracy rates of 87%, 92%, 38%, and 82% respectively.

Out of all the algorithms tested, the Generalized Linear Model demonstrated the highest accuracy at approximately 91.8% as shown in Table 6. The standard deviation for this model is about +/-1.8% which means the values are not spread out from the mean value. Other noteworthy performers were the Naive Bayes and Decision Tree algorithms, with accuracy rates of 87.1% and 82.2% and standard deviations of +/-1.7% and +/-1.2%, respectively. The Logistic Regression algorithm, on the other hand, performed the poorest, with only 38.3% accuracy and a standard deviation of +/-2.7%.

The Generalized linear model (GLM) is designed to model situations where the relationship between the input features and the output variable is essentially linear. The high accuracy of GLM suggests that the relationship between the distance ratios of keypoints and the words they are trying to predict is roughly linear in nature. This means that as the keypoint ratios change, the likelihood of predicting a certain word changes in a linear fashion. The linear relationship in the data fits well with the assumptions made by this model, leading to high accuracy.

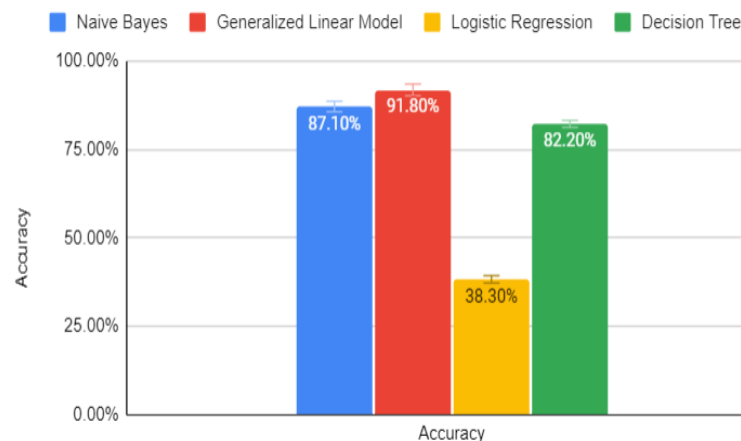


Figure 8. Accuracy comparison of different algorithms.

Naive Bayes works based on Bayes' theorem, which assumes that the predictors are independent of each other. In the context of our dataset, this means that each distance ratio (between keypoints) independently contributes to the probability of a particular word prediction. Despite its "naive" assumption, this algorithm can work surprisingly well in many real-world scenarios, which may be why we observed a relatively high accuracy with our dataset but it could not defeat GLM.

Decision trees divide the input space into regions and assign a label (or a class) to each region. Based on our provided data (distance ratios), the decision tree probably created decision boundaries based on certain thresholds of these ratios. While it performed decently, it might not have been able to capture all the details

and possible interactions between the different keypoint ratios as well as the GLM. In other words, the decision tree might not have been able to learn the complex relationships between the data points as well as the GLM.

Table 6. Comparison among the different algorithms

Algorithms	Performance	
	Accuracy	Standard Deviation
Naive Bayes	87.1%	+/-1.7%
Generalized Linear model	91.8%	+/-1.8%
Logistic Regression	38.3%	+/-2.7%
Decision tree	82.2%	+/-1.2%

Logistic regression is a statistical method that estimates the probability of a binary outcome by fitting data to a logistic curve. The model assumes that the log odds of the outcome are a linear combination of the predictors. In our dataset, the distance ratios between lip keypoints may not align with the assumptions of the logistic model. This is because the data may be too complex or multifaceted for the straightforward logistic transformation to capture accurately. As a result, the model gave us the poorest performance among all.

Table 7. Validation table of the generalized linear model

Predicting Words	Precision-Recall Data			
	Word 1 (Aamar)	Word 2 (Bhasha)	Word 3 (Bangla)	Class Precision
Word 1	227	11	11	89.72%
Word 2	4	94	5	91.26%
Word 3	1	2	106	97.25%
Class Recall	97.84%	87.85%	84.13%	

Total Accuracy: 91.83% +/- 1.8%

The precision-recall metric is used to verify the Generalized Linear Model's reliability. The three different words can be seen to have varying class precision and class recall values. As shown in Table 7, the precision of the predicted Word 1 (Aamar/ আমার) is 89.72%, Word 2 (Bhasha/ ভাষা) is 91.26% and Word 3 (Bangla/ বাংলা) is 97.25%. The class recall values are 97.84%, 87.85%, and 84.13% for the three words respectively.

4. CONCLUSIONS

Lip detection is quite a hectic task. In our paper, we have used the latest Mask RCNN algorithm which falls under the Artificial Neural Network. With the help of this, we have shown that lip detection is much easier, and better results can be achieved through this process. Our initial research consisted of around 1000 image frames and later on we added another 3000 image frames to train the system. So, our system is a well-learned system that can work on complex conditions as well. We have used the Bangla language and not much research has been previously done on this language. So, this algorithm and this type of work is comparatively new in the computer vision community. We would definitely like to expand our work in the near future in other fields as well. Our work can be very helpful in recognizing words from videos, CCTV camera footage, and physically disabled humans. In our paper, we have presented lip movements detected based on keypoints, so detection accuracy is a lot higher and we might be able to detect words in real time if improved resources are provided.

4.1. Further Implementation

Our research work is on lip movements and the detection of words from lip movements. This research can be used in a variety of processes. There can be a lot of implementations of this research. We can use this to detect words from CCTV camera footage. We can use this as an alternative to subtitles in movies or other videos. Again, we have done our work based on the language Bangla. We can further use this on other languages as well if we can manage the proper datasets of the languages.

In CCTV cameras we can see the videos only without the sound. It is not possible to know what the people are saying in those videos. So, with the help of Mask RCNN, we can detect the words from the lip movements of the objects. So, we can easily identify the sayings of the people from the videos. This might be of great help for the law-maintaining organizations as they can easily know what the people are saying. Therefore, this can be one of the most important implementations of the Mask RCNN algorithm.

In movies or TV series or any videos we mostly need subtitles. Subtitles are speech converted to text so that the videos can be understood well. Now our research is on similar things. We convert words from lip

movements. The system could be used as an alternative to subtitles. When the lip movement occurs in the videos, our algorithm will instantly translate those movements to words and work as subtitles.

Humans who are physically disabled and have problems speaking generally use sign language. The system can help these people a lot. Instead of sign language, if these people can just move their lips, others will be able to understand what they are saying. This is because the lip movement will result in written words. It is expected that this could be a good alternative to sign language.

If we take datasets of other languages and then work on the lip movements when speech is made in other languages then we will be able to detect those words as well. As a result, words can be converted from speech in any language.

ACKNOWLEDGEMENT

This research is funded by Woosong Univerity academic research 2024.




REFERENCES

- [1] S. W. Chin, K. P. Seng, L.-M. Ang, and K. H. Lim, "New lips detection and tracking system," in Proceedings of the international multiconference of engineers and computer scientists, vol. 1, 2009, pp. 18-20.
- [2] N. Oliver, A. P. Pentland, and F. Berard, "LAFTER: lips and face real time tracker," in Proc. IEEE Computer Society Conf. Comput. Vis. Pattern Recogn., San Juan, PR, USA, 1997, pp. 123-129.
- [3] Z.-M. Chan, C. Y. Lau, and K. F. Thang, "Visual Speech Recognition of Lips Images Using Convolutional Neural Network in VGG-M Model," J. Inf. Hiding Multim. Signal Process., vol. 11, pp. 116-125, 2020.
- [4] A. Aripin and A. Setiawan, "Indonesian Lip-Reading Recognition Using Long-Term Recurrent Convolutional Network," SSRN Electronic Journal, 2022. [Online]. Available: <https://ssrn.com/abstract=4444973>
- [5] R. El-Bialy et al., "Developing Phoneme-based Lip-reading Sentences System for Silent Speech Recognition," CAAI Trans. Intell. Technol., 2022.
- [6] Y. Fu, Y. Lu, and R. Ni, "Chinese Lip-Reading Research Based on ShuffleNet and CBAM," Applied Sciences, vol. 13, no. 2, p. 1106, Jan. 2023.
- [7] M. M. Rahman, M. R. Tanjim, S. S. Hasan, S. M. Shaiban, and M. A. Khan, "Lip Reading Bengali Words," in Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI '22), Sanya, China, 2023, Art. no. 22, pp. 1-6, doi: 10.1145/3579654.3579677.
- [8] G. Zhang and Y. Lu, "Research on a Lip-Reading Algorithm Based on Efficient-GhostNet," Electronics, vol. 12, no. 5, p. 1151, Feb. 2023.
- [9] A. Berkol et al., "Visual Lip-Reading Dataset in Turkish," Data, vol. 8, no. 1, p. 15, Jan. 2023.
- [10] Uddin, J., Arko, F. N., Tabassum, N., Trisha, T. R., & Ahmed, F. (2017, December). Bangla sign language interpretation using bag of features and Support Vector Machine. In 2017 3rd International Conference on Electrical Information and Communication Technology (EICT) (pp. 1-4). IEEE.
- [11] P. Bharati and A. Pramanik, "Deep Learning Techniques—R-CNN to Mask R-CNN: A Survey," in Computational Intelligence in Pattern Recognition, A. Das, J. Nayak, B. Naik, S. Pati, and D. Pelusi, Eds. Singapore: Springer, 2020, vol. 999. [Online]. Available: https://doi.org/10.1007/978-981-13-9042-5_56.
- [12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961-2969.
- [13] K. Ishizaki, K. Saruta and H. Uehara, "Detecting Keypoints for Automated Annotation of Bounding Boxes using Keypoint Extraction," 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2020, pp. 1691-1694, doi: 10.1109/CSCI51800.2020.00312.
- [14] "Free Video to JPG Converter," DVDVideoSoft, 2017. [Online]. Available: <https://www.dvdvideosoft.com/products/dvd/Free-Video-to-JPG-Converter.htm>. [Accessed: 03, July, 2023]
- [15] J.-X. Zhang, G. Wan, and J. Pan, "Is lip region-of-interest sufficient for lipreading?," in Proceedings of the 2022 International Conference on Multimodal Interaction, 2022.
- [16] "BIRME - Bulk Image Resizing Made Easy 2.0," BIRME. 2018. [Online]. Available: <https://www.birme.net/>. [Accessed: 03, July, 2023].
- [17] A. Dutta and A. Zisserman, "The VIA Annotation Software for Images, Audio and Video," in Proceedings of the 27th ACM International Conference on Multimedia (MM '19), New York, NY, USA: ACM, 2019. [Online]. Available: <https://doi.org/10.1145/3343031.3350535>
- [18] W. Abdulla, "Splash of color: Instance segmentation with mask r-cnn and tensorflow," Matterport Engineering Techblog, Mar. 20, 2018. [Online]. Available: <https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46>.
- [19] Q. Liu and Y. Wu, "Supervised Learning," in Encyclopedia of the Sciences of Learning, N. M. Seel, Ed. 2012. [Online]. Available: https://doi.org/10.1007/978-1-4419-1428-6_451.
- [20] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, Vol. 398. John Wiley & Sons, 2013.
- [21] J. A. Nelder and R. W. Wedderburn, "Generalized linear models," Journal of the Royal Statistical Society Series A: Statistics in Society, vol. 135, no. 3, pp. 370-384, 1972.




- [22] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 18, no. 6, pp. 275-285, 2004.
- [23] K. P. Murphy, "Naive bayes classifiers," *University of British Columbia*, vol. 18, no. 60, 2006.

BIOGRAPHY OF AUTHORS



Abul Bashar Bhuiyan    is a Senior Software Engineer at WPDeveloper Inc. He received a BSc. in Computer Science and Engineering from BRAC University, Bangladesh. His research interests are Machine Learning/Deep Learning prediction and detection and computer Vision, NLP for Medical.



Dr Jia Uddin    is as an Assistant Professor, Department of [AI and Big Data](#), Endicott College, Woosong University, Daejeon, South Korea. He received Ph.D. in Computer Engineering from University of Ulsan, South Korea, M.Sc. in Electrical Engineering (Specialization: Telecommunications), Blekinge Institute of Technology, Sweden, and B.Sc. in Computer and Communication Engineering, International Islamic University Chittagong, Bangladesh. He was a visiting faculty at School of computing, Staffordshire University, United Kingdom. He is an Associate Professor (now on leave), Department of Computer Science and Engineering, Brac University, Dhaka, Bangladesh. His research interests are Industrial Fault Diagnosis, Machine Learning/Deep Learning based prediction and detection using multimedia signals.