

An Adam based CNN and LSTM approach for sign language recognition in real time for deaf people

Subrata Kumer Paul¹, Md. Abul Ala Walid², Rakhi Rani Paul¹, Md. Jamal Uddin⁵, Md. Sohel Rana^{3,4},
Maloy Kumar Devnath⁵, Ishaat Rahman Dipu¹, Md. Momenul Haque¹

¹Department of Computer Science and Engineering, Bangladesh Army University of Engineering and Technology, Natore, Bangladesh

²Department of Computer Science and Engineering, Khulna University of Engineering and Technology (KUET), Khulna, Bangladesh

³Department of Electrical and Electronic Engineering, Northern University of Business and Technology Khulna, Khulna, Bangladesh

⁴Department of Computer Science and Engineering, Bangladesh University of Professionals, Dhaka, Bangladesh

⁵Department of Computer Science and Engineering, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj, Bangladesh

Article Info

Article history:

Received Feb 19, 2023

Revised May 24, 2023

Accepted Jul 13, 2023

Keywords:

Convolutional neural network

Deep learning

Gated recurrent unit

Long short-term memory

Recognition

Sign language

ABSTRACT

Hand gestures and sign language are crucial modes of communication for deaf individuals. Since most people can't understand sign language, it's hard for a mute and an average person to talk to each other. Because of technological progress, computer vision and deep learning can now be used to count. This paper shows two ways to use deep knowledge to recognize sign language. These methods help regular people understand sign language and improve their communication. Based on American sign language (ASL), two separate datasets have been constructed; the first has 26 signs, and the other contains three significant symbols with the crucial sequence of frames or videos for regular communication. This study looks at three different models: the improved ResNet-based convolutional neural network (CNN), the long short-term memory (LSTM), and the gated recurrent unit (GRU). The first dataset is used to fit and assess the CNN model. With the adaptive moment estimation (Adam) optimizer, CNN obtains an accuracy of 89.07%. In contrast, the second dataset is given to LSTM and GRU and a comparison has been conducted. LSTM does better than GRU in all classes. LSTM has a 94.3% accuracy, while GRU only manages 79.3%. Our preliminary models' real-time performance is also highlighted.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Md. Abul Ala Walid

Department of Computer Science and Engineering

Khulna University of Engineering and Technology (KUET)

Khulna-9203, Bangladesh

Email: abulalawalid@gmail.com

1. INTRODUCTION

In recent years, artificial intelligence (AI) has made people smarter by making it easier for them to process information, make decisions, and do tasks. In addition, it can help with medical diagnosis and treatment [1] as well as human development through education and training initiatives [2]. It can also be utilized to aid disabled people in various ways. For example, using an IoT-based deep learning approach to provide indoor thermal comfort for disabled people [3], assistive technologies such as text-to-speech, speech recognition, and natural language processing can benefit people who struggle with communication. People who have trouble moving around can get help with physical tasks from AI-powered robots and drones [4]. AI can also make personalized programs to help people with physical and neurological disabilities get better.

Also, chatbots, virtual assistants, and other interfaces powered by AI can make it easier for people with cognitive or developmental disabilities to get information and do tasks.

Deaf and hard-of-hearing people communicate with others or their community primarily through hand and body gestures in sign language (SL). It differs in vocabulary, meaning, and grammar from spoken and written language. Saying makes clear sounds linked to specific words and grammatical structures to send messages [5]. Visual hand and body motions are used in sign language to communicate important messages. People who are deaf get stressed because they can't use all services well because of communication problems [6]. Thirty-four million children and 432 million adults worldwide require rehabilitation for "disabling" hearing loss. A debilitating hearing loss is predicted to affect more than 700 million individuals by 2050, or 1 in 10 people. Large segments of society value sign language as an alternative means of communication. According to the International Federation of the Deaf, there are 70 million deaf people in the globe who use more than 300 sign languages [7], [8].

Each sign in a given sign language differs in terms of hand form, motion profile, and placement of the hand, face, and other body parts. Because of this, they think visual sign language is a complex area to study in computer vision [8]. Materially, it is a multidisciplinary topic that still needs to be solved in depth to fully understand what people are trying to say [9]. Several new AI technologies have been used to understand sign language in the past few years. According to recent studies, this can be performed in a few different ways [7]–[9]. One way is to use computer vision and machine learning algorithms to look at videos of people using sign language. These algorithms can be taught to recognize certain motions and signs to translate sign language into text or voice. Another method is to employ wearable technology, such as gloves or other accessories with sensors, to track the movements of the hands and fingers. These sensors can be used to track what the hands and fingers are doing, which can then be used by machine learning algorithms to recognize sure signs and signals. Due to several factors, sign language recognition is about 30 years behind voice recognition systems. One of the leading causes is that two-dimensional video signals require far more processing and recognition effort than one-dimensional audio signals. Also, the vocabulary and meanings of sign language have yet to be entirely found, and there are no dictionaries. Other than these, there are no universal definitions for many such signs. Games, virtual reality settings, robot control, and natural language conversations are just a few areas where these technologies can be successfully used [9], [10].

The sign language dataset is taken from a webcam and specifically from the droid cam client server to integrate the method with the phone [11]. Two datasets are prepared for this study, DS-1 and DS-2. The DS-1 set contains three classes, and the DS-2 set includes twenty-six types with a total 28,600 number of images. DS-1 is used to fit both long short-term memory (LSTM) and gated recurrent unit (GRU) for classifying three different classes "hello," "iamhungry," and "thanks." LSTM shows the utmost exactness on average for all performance measures (accuracy, precision, recall, and F1-score) compared to GRU. On the other hand, the DS-2 dataset has been utilized to prepare convolutional neural network (CNN) model. The 89.07% accuracy has been achieved from the CNN model generated by the proposed dataset. On the other hand, LSTM displays 94.3% accuracy.

2. LITERATURE REVIEW

In several previously published publications were dissected and analyzed. Huang *et al.* [10] talk about how the Kinect and a CNN can be used to recognize 3D sign language. The authors used 3D CNN to get spatial and temporal information from raw data to find natural features that could be used to adapt to the significant differences in hand movements. Moreover, a realistic dataset having twenty-five signs is used in their study. Pigou *et al.* [12] worked on the CNN-based recognition system and Microsoft Kinect was provided. Thresholding, background erasure, and median filtering were employed in this system's preprocessing. They used the accelerated gradient descent (NAG) optimizer that Nesterov made, which was very good at recognizing movements related to the Italian language. Wang *et al.* [13] used multidimensional hidden Markov models (HMMs) to identify the sign using a sensory globe and the bird's motion tracker. Support vector machine (SVM), k-nearest neighbor (KNN), logistic regression, and CNN are a few techniques that can be utilized to design a strategy that will make it much simpler for a non-signer to communicate with a signer, according to Priya *et al.* [14]. They contain the entire American sign language (ASL) grammar, consisting of 26 letters and ten numbers. The accuracy of the experimental results—80.30% for SVM and 93.81% for deep neural networks (DNN)—was encouraging. Srinivas *et al.* [15] said that to recognize the sign language system for the deaf and dumb, SVM, and ANN should be used for 26 different classes.

Hein *et al.* [16] implemented their way with two segments: a training segment and a classification segment. First, a webcam input video was taken for the training part. Then, using a component detection approach, we found the component. After finding the details, they use the localization method we suggested

to see where the head and hand are. Elmezain *et al.* [17] have also used a real-time system to create work based on the hidden Markov model (HMM) that can automatically recognize Arabic digits (0-9) in both isolated and continuous gestures. Other HMM topologies with separate states, such as ergodic, left-right (LR), and left-right banded (LRB), can also handle single gestures. Chakraborty *et al.* [18] put the English alphabet into groups based on how different hand gestures in the Indian sign language (ISL) show it. They did this by using Google's media pipelining API. Using this API, one can find the x, y, and z coordinates in three-dimensional space for each of the 21 landmarks on each hand. They found that by using the media pipe API, they could accurately predict the ASL and a few other SLs. In terms of accuracy, SVM has been compared to the random forest (RF), the KNN, and the decision tree (DT). SVM is the most accurate, with a 99% accuracy rate. Shankar *et al.* [19] say that object identification is the most common way computer vision is used. It is a technique for locating and modifying objects like furniture and artwork. Although numerous detection methods exist, their degrees of precision and effectiveness still need improvement. Phi *et al.* [20] developed a glove-based gesture recognition system using ten flex sensors and an accelerometer. Fang *et al.* [21] used data gloves and 3D position trackers to make a Chinese sign language recognition system that could understand 91.9% of the words in 1,500 test phrases. McGuire *et al.* [22] used a one-handed glove-based system and a hidden Markov model. This study has shown that using sensors and equipment to recognize sign language is expensive and needs to be more user-friendly to be portable.

3. METHOD

Figures 1 and 2 can be used to show how the proposed method works. The dataset for the first proposed model diagram was obtained from the media pipe holistic pose estimation. Essential points are collected by following the process described in Figure 1. Using a holistic media pipe library, points are accumulated from hand, face, and pose landmarks. After preprocessing several frames, the arrays are transferred into LSTM and GRU models for the training and testing phases, where the trained model is used to assess the prediction made from the test data.

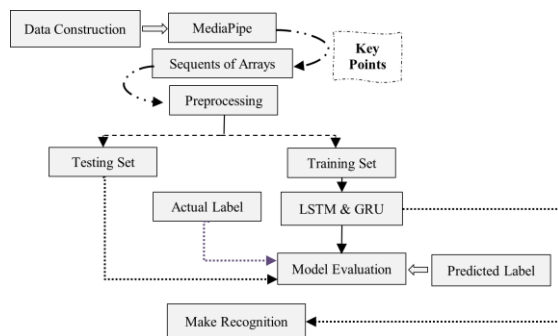


Figure 1. Block diagram of the proposed model for LSTM and GRU

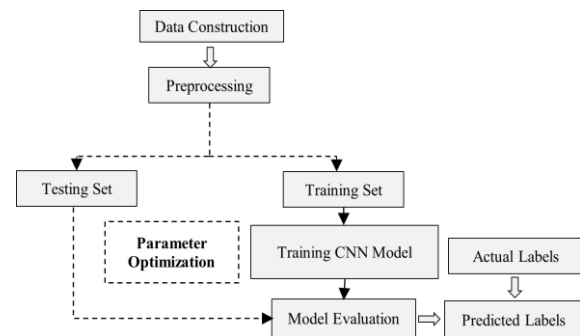


Figure 2. Block diagram of the proposed model for CNN

Our model has been introduced with several epochs for accurate recognition to gather the expected results acquired in our second walkthrough. Two different epochs with different sizes for our LSTM models and batch normalization have been added before the top layer. Our LSTM follows the CNN-time-series divided parasitic capacitance (TSDPC)-LSTM framework proposed by Tang *et al.* [23]. But instead of passing important frames from the video, LSTM takes a series of essential points in a 2D array. On the other hand, our proposed GRU follows the architecture of Haque *et al.* [24], where the sequence of a 2D array of critical points is also passed, excepting features of the frames. Dropout layers with a dropping rate of 0.5 were added to prevent overfitting the model. Additionally, the hidden and output layers defined the sigmoid and relu activation functions. The "SoftMax" activation function has been added to the output layer, and sparse categorical cross-entropy has been chosen as the loss function. In our model, adaptive moment estimation (Adam) [25] is regarded as an optimization method. The input shape is computed as (30 and 1660). For data acquisition, droid cam and 120×720-pixel camera devices have been used to acquire the images. But the number of pixels in the droid camera is higher because it comes from a phone. The method shown in Figure 2 has been used to get the results we wanted for classifying the 26 letters of the alphabet. The images are resized to 50×50 pixels, and several preprocessing steps are followed. Also, each of the photos is segmented through the collected images. After finishing the preprocessing steps, the images are fed

into a CNN model and split between train and test with the “processed train” dataset. Finally, the recognition is made using the independent test dataset. The architecture of the proposed CNN model is inspired by ResNet50 [26]. However, for our proposed CNN model, we used the architecture of improved ResNet50 introduced by Wu *et al.* [27].

3.1. Dataset description

The proposed method depends on deep learning techniques known as LSTM and CNN. The LSTM method is tested on our datasets of pose estimation data from different webcams. Three types of images are stored in three folders, being created named “hello,” “iamhungry,” and “thanks” to export the data to be stored as NumPy arrays (x, y, and z-axis of hand pose) and create labels. This dataset is recognized as a DS-1 set. Therefore, the DS-1 set is about the sequence of frames related to each category.

The second method is based on a convolutional neural network, where a dataset of 28,600 images is also collected by us using several webcams, each consisting of 26 different categories stored in 26 other folders with A-Z alphabets. Again, each folder contains two subfolders or subsections. The first section of the folders includes the raw image files, and the second section holds the preprocessed images. This dataset is recognized as a DS-2 set. There are images of 26 alphabets in the DS-2 collection.

3.2. Long short-term memory

The LSTM is made to reduce back-flow issues [28]. Hochreiter and Schmidhuber [28] created the LSTM algorithm, a modified recurrent neural network (RNN), and resolve the aforementioned error backpropagation problems. Cells, input gates, and output gates were the only components of the LSTM’s initial implementation.

Yet, the effectiveness of RNN [29] declines as the gap duration increases. Making a choice on the removal of unneeded information from the cell state is important for the first stage. The “forget gate layer,” one of the sigmoid layers, handles such decisions. In making decision, x_t and C_{t-1} are depicted in Figure 3, and the results for all of the numbers in C_{t-1} cells might be any value between 0 and 1. If the output is a ‘1’, it means that the information needs to be saved, whereas a ‘0’ means that it needs to be deleted. After that, it is now necessary to plan what data should be kept in the cells. There are two steps in this process. First, the values that need to be updated are resolved by the gate layer, which is also a sigmoid coat layer. Second, in this state, a tanh layer created just for adding generates a new character’s vector, i.e., t . This step involves execution once all planning and decision-making has been completed [30]. The following (1)-(3) can be used to compute the f_t , i_t and C_{t_hat} [28].

$$f_t = \sigma(W_f \cdot [hidden_{t-1}, dm_{x_t}] + b_f) \quad (1)$$

$$i_t = \sigma(W_i \cdot [hidden_{t-1}, dm_{x_t}] + b_i) \quad (2)$$

$$C_{t_hat} = \tanh(W_c[hidden_{t-1}, dm_{x_t}] + b_c) \quad (3)$$

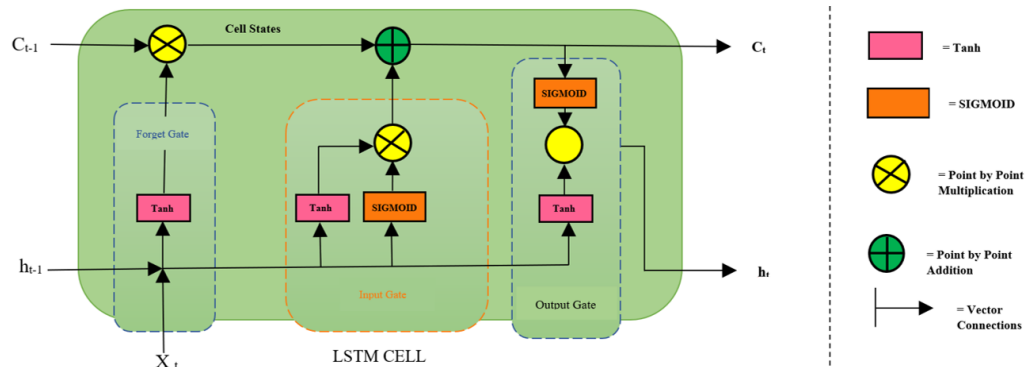


Figure 3. The basic block diagram of LSTM memory cell

3.3. Gated recurrent unit

GRU is a type of RNN that addresses the vanishing gradients problem by introducing gating mechanisms to control the flow of information within the network. It is also used for modeling sequential

data such as speech or text. It is similar to the more popular LSTM network but has fewer parameters and is easier to train. The GRU has a hidden state vector h that is updated at each time step t based on the current input X_t and the previous hidden state h_{t-1} . It also has two gating mechanisms: the reset gate r_t and the update gate z_t . The reset gate determines how much of the previous hidden state should be forgotten, while the update gate determines how much of the current input should be used to update the hidden state. The GRU's reset gate r_t , update gate z_t , candidate hidden state h_t' , and hidden state h_t can be computed by (4)-(7):

$$r_t = \text{sigmoid}(W_{x_r} * [h_{t-1}, X_t]) \quad (4)$$

$$z_t = \text{sigmoid}(W_{x_z} * [h_{t-1}, X_t]) \quad (5)$$

$$h_t' = \text{tanh}(W_{x_h} * [r_t * h_{t-1}, X_t]) \quad (6)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * h_t' \quad (7)$$

where W_{x_r} , W_{x_z} , and W_{x_h} are weight matrices, sigmoid is the sigmoid activation function, tanh is the hyperbolic tangent activation function, X_t is the current input, h_{t-1} is the previous hidden state, r_t is the reset gate, z_t is the update gate, h_t' is the candidate hidden state, and h_t is the updated hidden state. These equations show how the GRU gates work together to control the flow of information from the current input and the previous hidden state to the current hidden state. The reset gate r_t determines which parts of the previous hidden state should be forgotten, and the update gate z_t determines how much of the current input should be used to update the hidden state. The candidate hidden state h_t' is calculated based on the reset-gated previous hidden state and the current input, and is used to update the hidden state based on the update gate z_t . The structure of a memory cell is illustrated as a circuit diagram in Figure 4 [31].

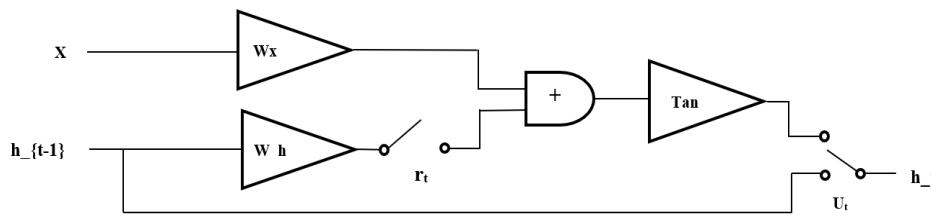


Figure 4. Structure of GRU memory cell

4. RESULTS AND DISCUSSION

For calculating some of the performance matrices like average precision, recall, fl-score, and overall accuracy, initially, the confusion matrix is produced. The following has been provided as the calculation formula for each of these phenomena. True positive (TP) results are those where the model accurately identified one class or positive class. A true negative (TN) is a result for which the model correctly predicts another class or negative class, much as a genuine positive. When the model predicts the positive class incorrectly, a false positive (FP) is produced. A false negative (FN) is a result that occurs when the model incorrectly predicts the negative class. In (8) to (11) are used to compute precision, F1 score, and accuracy from the confusion matrix. Tables 1-5 display the performance of several models created for this study. While precision frequently employs (9) to calculate the TP rate, recall frequently uses (11), which focuses on reducing FN rates [32]. A confusion matrix for LSTM and GRU is a 3×3 matrix [33] for showing how many positive and negative predictions are correct and incorrect for three classes “hello” (0), “iamhungry” (1) and “thanks” (2). Its confusion matrix illustrates whether the classification algorithm correctly or incorrectly classified the records into positive and negative classes.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

$$\text{Precision Score} = \frac{TP}{TP+FP} \quad (9)$$

$$\text{F1 - Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

$$\text{Recall} = \frac{TP}{FN+TP} \quad (11)$$

Each of the scores is evaluated for the measurement of further analysis: the confusion matrix of the LSTM, GRU, and CNN model is shown to calculate the model’s assessment in terms of accuracy, precision, and F1-score [34]. The classwise tested results of the LSTM and GRU models are listed in Tables 1 and 2, respectively based on SGD [29] and Adam [25] optimizing methods. These are basically classwise precision, recall, F1-score, and accuracy. Classwise TP, FP, TN, and FN scores are also shown in the form of a 3×3 confusion matrix in Figures 5 and 6.

Table 1. Classwise accuracy, precision, recall, and F1-score for LSTM

Optimizing method	Classes	Precision (%)	F1-score (%)	Accuracy (%)	Recall (%)
SGD	hello	92	90	85	88
	iamhungry	82	87.5	90	94
	thankyou	85	84	81	83
Adam	hello	100	100	93	100
	iamhungry	80	89	96	100
	thankyou	100	93	94	88

Table 2. Classwise accuracy, precision, recall, and F1-score for GRU

Optimizing method	Classes	Precision (%)	F1-score (%)	Accuracy (%)	Recall (%)
SGD	hello	74	61	81	52
	iamhungry	78	85	79	93
	thankyou	70	75	76	80
Adam	hello	75	60	82	50
	iamhungry	80	89	72	100
	thankyou	70	77	79	88

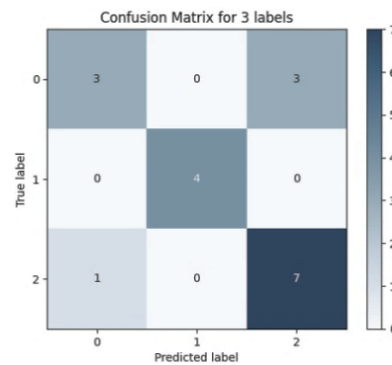
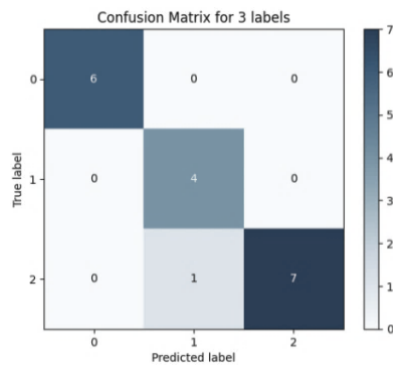


Figure 5. Confusion matrix for LSTM model Figure 6. Confusion matrix for GRU model

Figure 5 represents the confusion matrix of the LSTM model where the following record represents the records following “0: hello”, “1: iamhungry” and “2: thank you” signs considering 6, 4, and 7 true positive records respectively. Again, Figure 6 represents the confusion matrix of the GRU model where the following record represents the records following “0: hello”, “1: iamhungry” and “2: thank you” signs considering 3, 4, and 7 true positive records respectively. According to the investigation, the GRU model is not reliable for classifying the “hello” class whereas the LSTM model shows robustness for all of the classes. In Table 3, the overall performance of both of the models is analyzed and the average value of each performance measure is stored. The GRU model performs substantially worse on average than the LSTM model and LSTM is showing utmost performance with the Adam optimizing method which can be abundantly identified from Figure 7.

The 26×26 confusion matrix is calculated for the CNN model using predicted and actual outcomes. Again, the matrix includes TP, TN, FP, and FN values for all 26 classes that have been used to calculate classwise accuracy, precision, and recall for each of the 26 alphabet classes displayed in Table 4. Additionally, Table 5 gives the CNN model’s overall performance report based on optimizing methods namely SGD, RMSprop [29], and Adam [34]. Results can be clear from Figure 8 where the CNN model with Adam optimizing method is showing an increasing amount of precision but lowering the score of F1 measure. On average, accuracy of CNN model is 89.07% with Adam optimizing method. In contrast, the model optimised with SGD demonstrates the lowest levels of accuracy and F1-score.

Table 3. Comparative analysis of the overall performance of LSTM and GRU model

Model	Precision (%)	F1-score (%)	Accuracy (%)	Recall (%)
LSTM	93.3	94	94.3	96
GRU	75	75	77.6	79.3

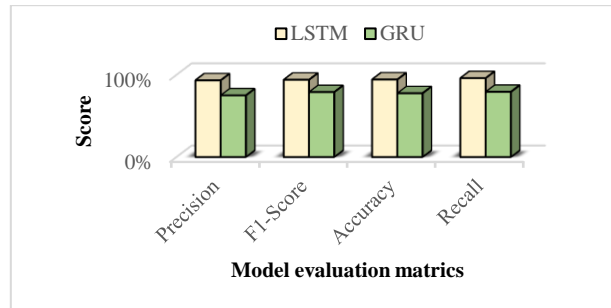


Figure 7. Performance comparison between LSTM and GRU

Table 4. Classwise accuracy, precision and recall for CNN model

Classes	Accuracy (%)	Precision (%)	Recall (%)	Classes (%)	Accuracy (%)	Precision (%)	Recall (%)
A	93	96	100	N	93	96	87
B	94	77	87	O	97	96	93
C	100	100	93	P	85	87	94
D	82	93	93	Q	85	100	98
E	87	97	96	R	77	88	83
F	72	95	87	S	100	87	85
G	88	88	100	T	87	93	84
H	100	85	78	U	96	99	85
I	87	76	89	V	93	93	73
J	81	72	66	W	93	91	93
K	85	100	92	X	72	100	87
L	82	84	100	Y	93	100	75
M	100	89	100	Z	94	92	93

Table 5. Performance report of CNN model

Model	Optimizing method	Precision (%)	F1-score (%)	Accuracy (%)	Recall (%)
CNN	SGD	82	86	88	90
	RMSprop	84	86.4	89	89
	Adam	91.3	88.37	89.07	88.88

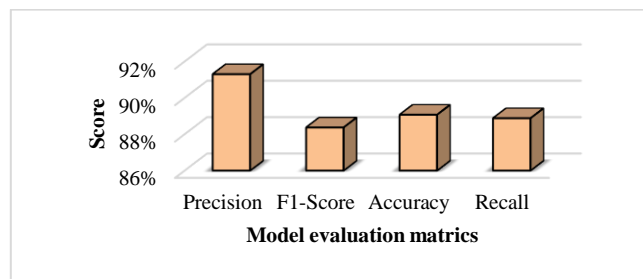


Figure 8. Performance analysis of CNN

In Table 6, the performance comparisons among several existing methods are highlighted. From our analysis, it can also be included that although the gloves and devices give good results and a better understanding of recognizing the signs such as implementing cyber gloves [13] with the Hidden Markov model, but it is not cost-friendly for lots of people. Using a vision-based approach we can reduce the cost by only the trained dataset and model to recognize the sign with only captured devices such as a webcam or by using a droidcam (a third-party mobile camera to a webcam app) to capture from a device.

Table 6. Comparative analysis of the related and existing approaches

Ref.	Used method(s)	Accuracy (%)
[13]	Cyber gloves and HMMs	95
[16]	Leap motion devices	90.82
[20]	Flex sensors	90.34 (precision score)
[21]	Data gloves and 3D position trackers	91.9
[22]	One-handed glove-based system and HMMs	94
Proposed work	LSTM	94.3
	GRU	76
	CNN	89.07

5. REAL-TIME ANALYSIS

In this section, it will be determined how our method does all of the operations that have been carried out in the actual scene in real-time analysis. Real-time analysis is necessary because without monitoring in real time it can not be figured out how the model will perform in the deployed states that the feature can be combined to form integrated benefits to the users. Figure 9 gives a glimpse of recognized sign language from the segmented images. When the hand is set to the region of the boxes, it captures and identifies the 50×50 images from the region where the green box is located.



Figure 9. Sign language recognition using CNN model

Then it converts or deciphers the hand sign language into an interpretable American sign alphabet which contains the letters from A to Z. Each time our system shows different signs on the region of the box, it takes the sign and interprets it continuously. This recognition has been performed using our proposed improved ResNet50-based CNN model. On the other hand, Figures 10-12 describe about real-time three significant types of sign language recognition via our proposed method assembled with LSTM method. Figure 10 demonstrates how our system reacts to the sign language action “hello”. Figure 12 displays how our system reacts to the sign language action “thanks”. Again, our system reacts to the feeling of hunger in the sign language action in real-time demonstrated in Figure 11. Moreover, our proposed system successfully recognizes the sign language actions in each of the three scenarios in real-time.

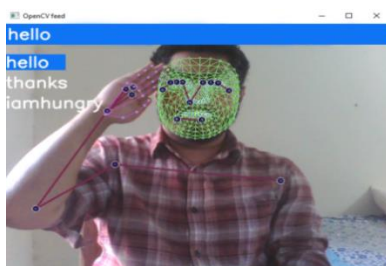


Figure 10. Detecting sign language of “hello”

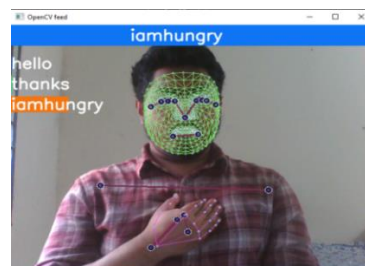


Figure 11. Detecting sign language of “iamhungry”

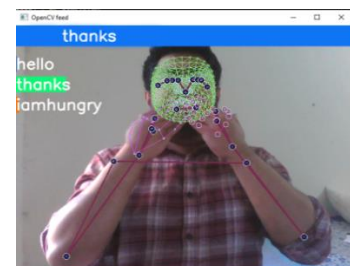


Figure 12. Detecting sign language of “thanks”

6. CONCLUSION

In this study, a CNN and LSTM method-based sign language recognition system has been introduced. This system may help people who are deaf, mute, or do not know sign language communicate much better. Two datasets are created in this motive: one has 26 signs, and the other has three important

symbols with the necessary sequence of frames or movies for normal conversation. The improved ResNet-based CNN, LSTM, and GRU are examined in this paper. The Adam optimizer produces satisfactory results across the board. Using the first dataset to fit and evaluate the CNN model yields an accuracy of 89.07%. In contrast, the second dataset is given to LSTM and GRU, and a model evaluation comparison study is conducted. LSTM outperforms GRU in every discipline. LSTM has an accuracy of 94.3%, while GRU only achieves a success rate of 79.3%. The proposed system can accurately recognize and translate sign language into speech or text. This makes sign language recognition easier to use and more common, giving people in need a vital way to talk. As technology keeps getting better and more languages are added, the chances that these systems will improve the lives of deaf and mute people will only go up. In the future, adding more languages and using sign-to-text and text-to-speech tools will be possible.




REFERENCES

- [1] F. S. Baji, S. B. Abdullah, and F. S. Abdulsattar, "K-mean clustering and local binary pattern techniques for automatic brain tumor detection," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 3, pp. 1586–1594, Jun. 2023, doi: 10.11591/eei.v12i3.4404.
- [2] M. A. A. Walid, S. M. M. Ahmed, M. Zeyad, S. M. S. Galib, and M. Nesa, "Analysis of machine learning strategies for prediction of passing undergraduate admission test," *International Journal of Information Management Data Insights*, vol. 2, no. 2, p. 100111, Nov. 2022, doi: 10.1016/j.ijimei.2022.100111.
- [3] B. Brik, M. Esseghir, L. Merghem-Boulahia, and H. Snoussi, "An IoT-based deep learning approach to analyse indoor thermal comfort of disabled people," *Building and Environment*, vol. 203, p. 108056, Oct. 2021, doi: 10.1016/j.buildenv.2021.108056.
- [4] X. Liu, X. He, M. Wang, and H. Shen, "What influences patients' continuance intention to use AI-powered service robots at hospitals? The role of individual characteristics," *Technology in Society*, vol. 70, p. 101996, Aug. 2022, doi: 10.1016/j.techsoc.2022.101996.
- [5] Z. Chen, X. Liu, M. Kojima, Q. Huang, and T. Arai, "A Wearable Navigation Device for Visually Impaired People Based on the Real-Time Semantic Visual SLAM System," *Sensors*, vol. 21, no. 4, Feb. 2021, doi: 10.3390/s21041536.
- [6] W. C. Stokoe, "Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf," *Journal of Deaf Studies and Deaf Education*, vol. 10, no. 1, pp. 3–37, Jan. 2005, doi: 10.1093/deafed/eni001.
- [7] R. Rastgoo, K. Kiani, and S. Escalera, "Sign Language Recognition: A Deep Survey," *Expert Systems with Applications*, vol. 164, Feb. 2021, doi: 10.1016/j.eswa.2020.113794.
- [8] R. Elakkiya, "RETRACTED ARTICLE: Machine learning based sign language recognition: a review and its research frontier," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 7, pp. 7205–7224, Jul. 2021, doi: 10.1007/s12652-020-02396-y.
- [9] A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs," *Neural Computing and Applications*, vol. 32, no. 12, pp. 7957–7968, Jun. 2020, doi: 10.1007/s00521-019-04691-y.
- [10] J. Huang, W. Zhou, H. Li, and W. Li, "Sign Language Recognition using 3D convolutional neural networks," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*, Jun. 2015, pp. 1–6, doi: 10.1109/ICME.2015.7177428.
- [11] K. Aggarwal and A. Arora, "An Approach to Control the PC with Hand Gesture Recognition using Computer Vision Technique," in *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, Mar. 2022, pp. 760–764, doi: 10.23919/INDIACom54597.2022.9763282.
- [12] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, "Sign Language Recognition Using Convolutional Neural Networks," in *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I 13*, 2015, pp. 572–578, doi: 10.1007/978-3-319-16178-5_40.
- [13] H. Wang, M. C. Leu, and C. Oz, "American Sign Language Recognition Using Multi-dimensional Hidden Markov Models," *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING*, vol. 22, pp. 1109–1123, 2006.
- [14] U. H. Priya, S. K. Prasad, M. M. Jacob, R. R. Krishna, and P. R. Vinod, "American sign language recognition using CNN," *International Journal of Research in Engineering, Science and Management*, vol. 3, no. 7, pp. 333–336, 2022.
- [15] L. V. Srinivas, C. Raminaidu, D. Ravibabu, and S. S. Reddy, "A framework to recognize the sign language system for deaf and dumb using mining techniques," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 29, no. 2, pp. 1006–1016, Feb. 2023, doi: 10.11591/ijeecs.v29.i2.pp1006-1016.
- [16] Z. Hein, T. P. Htoo, B. Aye, S. M. Htet, and K. Z. Ye, "Leap Motion based Myanmar Sign Language Recognition using Machine Learning," in *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, Jan. 2021, pp. 2304–2310, doi: 10.1109/ElConRus51938.2021.9396496.
- [17] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis, "A Hidden Markov Model-based continuous gesture recognition system for hand motion trajectory," in *2008 19th International Conference on Pattern Recognition*, Dec. 2008, pp. 1–4, doi: 10.1109/ICPR.2008.4761080.
- [18] S. Chakraborty, S. Banerjee, N. Bandyopadhyay, Z. Sarkar, and P. Chakraverty, "Indian Sign Language Classification (ISL) using Machine Learning," *American Journal of Electronics & Communication*, vol. 1, pp. 17–21.
- [19] R. S. Shankar, L. V. Srinivas, P. Neelima, and G. Mahesh, "A Framework to Enhance Object Detection Performance by using YOLO Algorithm," in *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, Apr. 2022, pp. 1591–1600, doi: 10.1109/ICSCDS53736.2022.9760859.
- [20] L. T. Phi, H. D. Nguyen, T. T. Q. Bui, and T. T. Vu, "A glove-based gesture recognition system for Vietnamese sign language," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, Oct. 2015, pp. 1555–1559, doi: 10.1109/ICCAS.2015.7364604.
- [21] G. Fang, W. Gao, and D. Zhao, "Large-Vocabulary Continuous Sign Language Recognition Based on Transition-Movement Models," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 1, pp. 1–9, Jan. 2007, doi: 10.1109/TSMCA.2006.886347.
- [22] R. M. McGuire, J. Hernandez-Rebollar, T. Stamer, V. Henderson, H. Brashear, and D. S. Ross, "Towards a one-way American Sign Language translator," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, 2004, pp. 620–625, doi: 10.1109/AFGR.2004.1301602.
- [23] H. Tang, L. Ding, S. Wu, B. Ren, N. Sebe, and P. Rota, "Deep Unsupervised Key Frame Extraction for Efficient Video Classification," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 19, no. 3, pp. 1–17, Aug.




- 2023, doi: 10.1145/3571735.
- [24] M. Haque, S. Afsha, and H. Nyeem, "Developing BrutNet: A New Deep CNN Model with GRU for Realtime Violence Detection," in *2022 International Conference on Innovations in Science, Engineering and Technology (ICISSET)*, Feb. 2022, pp. 390–395, doi: 10.1109/ICISSET54810.2022.9775874.
- [25] Y. Xue, Y. Tong, and F. Neri, "An ensemble of differential evolution and Adam for training feed-forward neural networks," *Information Sciences*, vol. 608, pp. 453–471, Aug. 2022, doi: 10.1016/j.ins.2022.06.036.
- [26] K. C. Sharmili, G. P. Suja, E. Pandian, M. A. A. Walid, S. Arunachalam, and G. C. Babu, "An Effective Diagnosis of Alzheimer's Disease with the Use of Deep Learning based CNN Model," in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, May 2023, pp. 443–448, doi: 10.1109/ICICCS56967.2023.10142306.
- [27] D. Wu, Y. Ying, M. Zhou, J. Pan, and D. Cui, "Improved ResNet-50 deep learning algorithm for identifying chicken gender," *Computers and Electronics in Agriculture*, vol. 205, p. 107622, Feb. 2023, doi: 10.1016/j.compag.2023.107622.
- [28] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [29] A. Kumar, S. Sarkar, and C. Pradhan, "Malaria Disease Detection Using CNN Technique with SGD, RMSprop and ADAM Optimizers," 2020, pp. 211–230, doi: 10.1007/978-3-030-33966-1_11.
- [30] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning," *Decision Analytics Journal*, vol. 3, p. 100071, Jun. 2022, doi: 10.1016/j.dajour.2022.100071.
- [31] S. Dhulipala, F. F. Adedoyin, and A. Bruno, "Sign and Human Action Detection Using Deep Learning," *Journal of Imaging*, vol. 8, no. 7, p. 192, Jul. 2022, doi: 10.3390/jimaging8070192.
- [32] M. A. A. Walid, S. M. Masum Ahmed, and S. M. S. Sadique, "A Comparative Analysis of Machine Learning Models for Prediction of Passing Bachelor Admission Test in Life-Science Faculty of a Public University in Bangladesh," in *2020 IEEE Electric Power and Energy Conference (EPEC)*, Nov. 2020, pp. 1–6, doi: 10.1109/EPEC48502.2020.9320119.
- [33] M. A. A. Walid *et al.*, "Adapted Deep Ensemble Learning-Based Voting Classifier for Osteosarcoma Cancer Classification," *Diagnostics*, vol. 13, no. 19, p. 3155, Oct. 2023, doi: 10.3390/diagnostics13193155.
- [34] S. K. Paul and R. R. Paul, "Speech Command Recognition System using Deep Recurrent Neural Networks," in *2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, Nov. 2021, pp. 1–6, doi: 10.1109/ICEEICT53905.2021.9667795.

BIOGRAPHIES OF AUTHORS






Subrata Kumer Paul    completed his B.Sc. and M.Sc. Engineering from Rajshahi University, in Computer Science and Engineering in 2016 and 2018, respectively. Now, he is working as an Assistant Professor at the Bangladesh Army University of Engineering and Technology (BAUET), Qadirabad Cantonment, Natore-6431, Bangladesh. Recently, he has been pursuing the MPhil Program at Rajshahi University. His research fields are speech signal processing, data mining, and machine learning. He can be contacted at email: sksubrata96@gmail.com.






Md. Abul Ala Walid    received his B.Sc. Engineering degree from Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj, Bangladesh, in Computer Science and Engineering, and completed Masters in Applied Statistics and Data Science from Jahangirnagar University, Savar, Dhaka-1342, Bangladesh. His areas of interest in research include computer vision, biomedical image processing, clinical data analysis, machine learning, and deep learning. He can be contacted at email: abulalawalid@gmail.com.






Rakhi Rani Paul    graduated her B.Sc. and M.Sc. Engineering from Rajshahi University, in Computer Science and Engineering in 2017 and 2019, respectively. Now, she is working as a Lecturer at Bangladesh Army University of Engineering and Technology (BAUET), Qadirabad Cantonment, Natore-6431, Bangladesh. Her research field is speech signal processing. She can be contacted at email: rakhipaul.cse@gmail.com.






Md. Jamal Uddin    received the B.Sc. and M.Sc. degrees in computer science and engineering from Rajshahi University, Bangladesh. He is currently an Assistant Professor at the Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj. His research interests include artificial intelligence, machine learning, deep learning, and computer vision. He can be contacted at email: jamal.bsmrstu@gmail.com.






Md. Sohel Rana    graduated from Prime University (PU), Bangladesh, with a Bachelor of Science in Electrical and Electronic Engineering degree in 2015. In 2018, he graduated with a Master of Science in Telecommunication Engineering from the UITS in Bangladesh. In the year 2022, a second M.Sc. Engineering degree was awarded by the BUP in Bangladesh. Some of his research interests are microstrip patch antenna, wireless communication, image processing, digital signature, cyber security, renewable energy, solar cell, biomedical engineering, control systems, and power electronics. Mr. Rana authored and co-authored in 39 papers regarding to his research works. At present, he is working as a Lecturer at NUBTK, Khulna, Bangladesh. He can be contacted at email: sohel.rana@uits.edu.bd.






Maloy Kumar Devnath    is a doctoral student in Information Systems at the University of Maryland, Baltimore County. His research interest is mainly in applied machine learning, graph-based ML, and sensor computing. Prior to his doctoral studies, Maloy served as an Assistant Professor at Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj. He has completed his B.Sc. Engineering degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology. He can be contacted at email: at maloy.cse.buet@gmail.com.



Ishaat Rahman Dipu    received his B.Sc. degree from Bangladesh Army University of Engineering and Technology, in Computer Science and Engineering in 2022. His specialization is in data science, deep learning, and computer vision. He can be contacted at email: ishaat48@gmail.com.



Md. Momenul Haque    finished his B.Sc. from Bangladesh Army University of Engineering and Technology (BAUET), in Computer Science and Engineering in 2022. Now, he is pursuing his M.Sc. from Rajshahi University of Engineering & Technology (RUET). At present, he is working as a lecturer under the Department of Computer Science and Engineering (CSE) at Rabindra Maitre University (RMU), Kushtia-7000, Bangladesh. His specializations are in data science, deep learning, and computer vision. He can be contacted at email: mominulhaquemim13@gmail.com.