Open Access Journal on Engineering Applications
ISSN: 3105-7004
www.oajea.hafizlab.com

Real Time English Alphabet Recognition Through Hand Gestures on Air Using Deep Learning and OpenCV

Fahmida Islam[1] **and Prome Saha Resha**[2,*]

[1,2]Department of Computer Science and Engineering, The People's University of Bangladesh, Dhaka, Bangladesh

Article Info
Received: 1 March 2026
Revised: 10 March 2026
Accepted: 15 March 2026
Published: 18 March 2026
Volume No: 01
Issue No: 02
Page No: 34-41

Corresponding author:
Prome Saha Resha
promesaharesha5678@gmail.com

DOI:
10.64886/oajea.0102.004

Check for updates

Abstract

Pattern recognition, computer vision, and image processing all are benefitted from hand-written alphabet recognition and categorization. A profusion of applications based on this domain have been created in the last few decades, such as sign identification, multilingual learning systems, and so on. This research shows how neural networks may be used to create a system that recognizes handwritten English alphabets in the air using hand gestures. Because of the acoustic similarities between the letters of the alphabet, this is a challenging undertaking to complete. The main problem is dealing with enormous different ways to write used by multiple peoples. There are a variety of alphabet-writing approaches in these complicated handwritten styles. The recognition of handwritten English alphabets has been the subject of several research studies. Several studies have been conducted on this subject, but none have proven effective in detecting English alphabets instantly moving your fingers in the breeze. Therefore, this article explains how to create an English Alphabet model of awareness that uses a Convolution Neural Network (CNN) to identify English alphabets based on hand motions the gap in the air. After a full analysis, this recommended approach achieved 93.08% accurate responses over the EMNIST dataset.

Keywords:
English alphabets, Sign Language, TensorFlow, CNN, OpenCV

1. Introduction

The goal of handwriting recognition or HWR is to translate language given in a spatial plot representations into symbolic pictures [1]. Due to the many types of handwriting, identification without help of handwritten letters are a time-consuming job [2]. These systems are broken down into two groups: online as well as offline character recognition. Data is entered into online handwriting recognition systems utilizing a pad and a digital pen. Moreover, writing strikes are depending on keystrokes, velocity, duration, and other factors. As a result, the image is characterized by writing speed, various pen motions, and so on. Offline data, on the contrary, is obtained through scanning a document and displayed as a 2D matrix. Online handwriting recognition applications include interfaces for a variety of computer devices, including tablet PCs, PDAs, and smart phones [3]. Furthermore, the HWR online is pretty important in the signature authentication procedure. Also, offline identification tasks are beneficial in a variety of situations, such as processing bank checks, offering learning aids for blind people, and automated postal mail sorting. As a result, online HWR is a critical field in which great progress has been made. However, employing machine learning also deep learning, there is still a lot of room to further the research.

After Standard Chinese and Spanish, English is the most commonly used language in the world. It is also the third most spoken native global language [4,5]. It is the most extensively studied second language, with about 60 sovereign states using it as their dominant language or even one of their official languages. Following table 1 shows the the English alphabet, including upper and lower case and their corresponding output provided by the system:

Table 1: The English alphabet, including upper and lower case also their corresponding output

English Alphabet	Output of the System
A, a	a
B, b	b
C, c	c
D, d	d
E, e	e
F, f	f
G, g	g
H, h	h
I, i	i
J, j	j
K, k	k
L, l	l
M, m	m
N, n	n
O, o	o
P, p	p
Q, q	q
R, r	r
S, s	s
T, t	t
U, u	u
V, v	v
W, w	w
X, x	x
Y, y	y
Z, z	z

Human-computer interaction necessitates the use of vision-based technologies to recognize hand movements or in short HCI. A gesture is a visual representation of a physical activity or a feeling. Gestures may be used to interact between humans and computers. For decades, numerous researchers have been focused on hand motion detecting technologies. A lot of apps depend on heavily on recognizing hand gestures, including sign language identification, augmented and virtual reality, sign language interpreters for the visually impaired, and robot control, to name a few. There are several research papers on this issue, including some for the English language, but none of them has been done for real-time English alphabet identification using hand gestures. Furthermore, recognizing handwritten letters in English seems to be more challenging than printed versions of characters for the reasons given below:

- Alphabets written through various people are not just unique., although they are often varying in size also shape.
- Each letter of the alphabet can be written in a variety of ways, making identification challenging.
- Overlaps, which are similar alphabets in different forms, make it even more difficult to recognize characters.

In recognizing applications like these, machine learning techniques have a lot of potential. It is built on AI, which allows a system that learn and develop automatically [6] based on prior experiences. To summarize, it is difficult to adequately describe handwritten alphabets due to the wide range of kinds of writing and the difficulty of properties of handwritten numerals. There are twenty-six letters in a standard English alphabet. It also has plenty of numbers that all of them the same format. This builds it hard to improve performance using a basic classification strategy and obstructs the development of a reliable English a system for recognizing handwritten letters. However, the dearth of these real-time works in English, as well gestures, indicates the potential for further research in this fascinating topic. However, because handwriting differs from person to person, there is still promise for developing a more sophisticated handwritten character recognition system. The writing style, the form of the alphabets, and their sizes all contribute to the difficulty of recognizing the characters. In this study, I proposed a real-time English alphabet a technique for identifying people based on their hand motions. To achieve these goals, we constructed the CNN model to guess English characters in order to use the biggest model for making predictions in English. Using OpenCV, well-known image classification approach, uses cameras to detect movement of the hand in real time. There have just a little open-source datasets for English handwriting available online. The EMNIST dataset was used to build the model.

Overview of the System

- First of all, the CNN model is prepared and trained for EMNIST dataset.
- After running the system, the webcam of the pc will open as a video frame to capture alphabets real time written on air.
- It will start capturing the data when it detects the blue object and finish capturing when the hand gesture is stopped.
- After capturing the digit, it will try to classify using adopted CNN model.
- Finally, the detected digit will be shown as output.

2. Related Work

There have been a few tries made in the field of real-time English Alphabet recognition identification by hand signals. Even though they were designed specifically for English sign language, handwritten characters or digit recognition are available for other languages.

This section reviews the literature on the ability to read handwritten characters utilizing SVM, KNN, neural networks, and other learning and classification approaches [7]. HWR systems are separated into two groups: systems that are online and systems that are not online [8], each with a distinct function to play in various domains, like reading data from smart phones and PDAs.

Existing literature [9] provides a complete explanation of different data input techniques, data format, and application domains in these two categories. For identifying offline handwriting data, H. Zhang et al. suggested a hybrid classification approach in 2006 [10]. The color, texture, and form of the text pictures are used to identify them. When compared to KNN, the computational cost of SVM is shown to be higher. The suggested model has the ability to classify several sets of data.

Authors Jamil Abedalrahim Jamil Alsayaydeh et al. [11] suggested an immediate HTR system based on a CNN structure that runs on a Raspberry Pi 4B platform using OpenCV and TensorFlow. Their technique, which used the EMNIST dataset for training, got a high ability to recognize of 91.58% for English letters and numbers. This was far better than other models like SVM, CTC, and RNN. Mubashir Ali et. al. [12] built a system that can recognize New Zealand Sign Language (NZSL) in real time using MediaPipe and OpenCV to pull hand landmark features from a custom dataset with 100,000 records. They looked at a lot of different machine learning algorithms and found that the Random Forest Classifier was the best one. It was able to correctly identify 99.52% of letters and digits.

Aseel Qedear et al. [13] created an updated Arabic air-writing detection system that uses a 7-layer CNN structure trained on a filtered dataset consisting of 31,349 photos. The system had a high evaluation accuracy of 96.40% and a real-time accuracy for recognition of 84%. This made it a good and interactive way for kids to acquire the Arabic alphabet. Vattikuti Shravya and his team [14] came up with an air writing system that uses MediaPipe and OpenCV to recognize letters and numbers and doodles without the need for hardware sensors. The system used CNN, RNN, and LSTM models to reach peak assessment accuracy of 98% for integers and characters 97.01% for play. This made it a fun way for dyslexic kids to learn.

Khalid M. O. Nahar et al. [15] came up with a hybrid model for recognizing Arabic air-writing. It uses CNN architectures like VGG16, VGG19, and SqueezeNet to find deep features and then machine learning and OCR approaches to sort them. When tested on the AHAWP dataset, the system reached a high recognition accuracy of 88.8% by employing a Neural Network (NN) with VGG16 includes and basic grid search optimization.

3. Methodology

The design of the model has two parts: The first stage is to get ready the digital photographs taken by a camera that records hand movements in real time. The second step is to use the CNN model to sort the pictures. To improve the classification accuracy of the proposed CNN model, I added a preprocessing module that comprises segmentation, morphological dilation and erosion, a Gaussian filtering, clipping, alongside normalization. To understand the numbers in real time, the model was saved and loaded into OpenCVs. Figure 1 shows its block diagram.

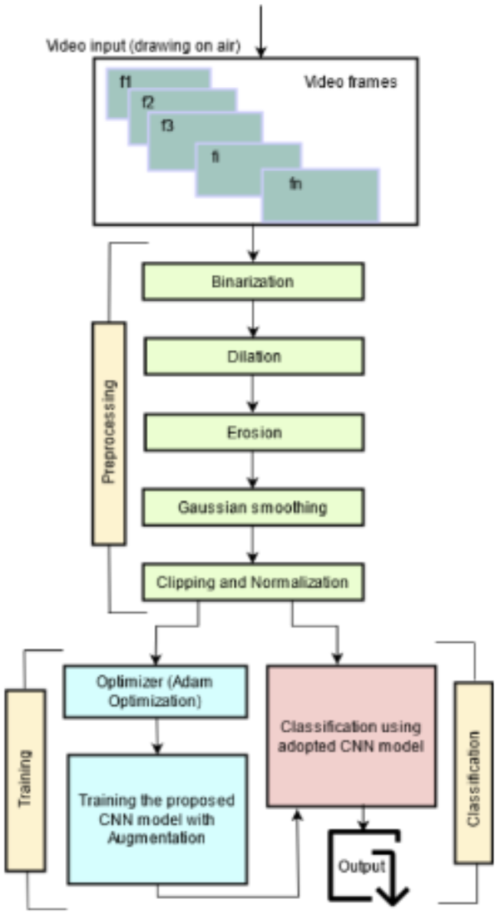

Figure 1: Block diagram of proposed system

To complete our task, we used the EMNIST dataset that was readily available. It's a single dataset of handwritten English characters. The MNIST collection of handwritten digits has 60,000 samples for training and 10,000 for testing. It's a smaller part of the NIST's larger collection. Steps included below will explain our full methodology,

Step 1. Data Preprocessing: To train the model, we first collected the dataset. The EMNIST dataset is a set of handwritten character digits taken from NIST Special Database 19 and converted into 28x28 pixel images. It has the same structure as the MNIST dataset.The EMNIST Letters dataset combines a balanced collection of upper and lowercase letters into one job with 26 classes with 145,600 characters divided into 26 balanced classes.

Step 2. Data Preprocessing: We began by fixing dataset's incorrect labeling photos, then deleting some of the incorrect photos, such as missing images. Edge thickening, flipping the forefront and background colors, and cutting down on noise with a median filter were included in the EMNIST dataset since it already includes inverted images. To make a binary picture, this RGB image must be transformed to grayscale. Inverted images, on the other hand, are already contained in the collection. As a result of the resizing, the image may lose some information. After resizing the images, we used the Lanczos interpolation algorithm after resizing the images.

Step 3. Implementation Requirements: A deep learning model requires a high-end PC with a GPU and other tools. Below is a list of the tools needed for this model. Hardware and Software:

- Intel Core i3 5th generation with 4GB RAM
- 1 TB HDD
- Camera (Webcam)
- Google Colab with 12GB GPU and 350 GB RAM
- Development tool: Python, Tensorflow, Keras, OpenCV, Pandas, Numpy.

Step 4. Model Development: Neural networks are commonly used to solve pattern recognition and image analysis problems. The most promising tool for doing so is CNN, known as deep learning method that leverages the way neurons interact in the brains of animals as a model.

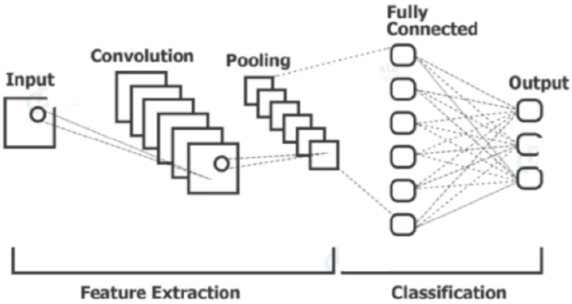

Figure 2: CNN for English Alphabet recognition

The overall concept of a CNN is shown in Figure 2. It has two primary parts: a feature extractor as well a classifier. The output from each layer (the input layer) in the feature extraction unit goes to the middle layer below it, which is the output layer for the previous layer. The next layer gets the current output as input, and the categorization section uses the input data to guess what will happen.

The inputs go via a nonlinear function called ReLU before they are processed through a succession of kernels. Assume that f stands for ReLU's activation function. If x_i^{l-1} receives input from the $(l-1)^{\text{th}}$ layer, k_{ilj}^l are the kernels of the l^{th} layer. With b_j^l, the l^{th} layer's preference is demonstrated. The following is how the convolution operation is expressed:

$$x_j^l = f\left(x_i^{l-1} * k_{ilj}^l\right) + b_j^l \tag{1}$$

For example, a 2×2 down sampling kernel's output dimensions are half the size of their corresponding input dimensions. An example of a pooling operation is as follows:

$$x_j^l = \text{down}\left(x_i^{l-1}\right) \tag{2}$$

However, in order to enhance the amount of characteristics mapped, it is often increased to choose or map the most acceptable input picture characteristics. The inputs to the fully connected network come from the CNN's last layer outputs. It uses the Softmax approach to provide classification results. The Softmax operation for the i^{th} class may be stated for any input sample x, any weight vector w, and different linear functions k:

$$P(y = i \mid x) = \frac{\exp(x^T)}{\sum_{k=1}^{K} \exp(x^T w_k)} \tag{3}$$

Table 2: The proposed CNN model summary

Layer (type)	**Output Shape**	**Param**
Conv2D	(None, 26, 26, 32)	320
Conv2D	(None, 24, 24, 64)	18496
MaxPooling2D	(None, 12, 12, 64)	0
Dropout	(None, 12, 12, 64)	0
Flatten	(None, 9216)	0
Dense	(None, 128)	1179776
Dropout	(None, 128)	0
Dense	(None, 26)	3354

4. Result and Discussion

For training the designed model, I utilized 20 epochs. The training came to an end once 20 epochs had been completed. Validation and test accuracy are inferior to training accuracy. Because our model was trained to distinguish rotated, shifted, zoomed, superimposition, and occluded pictures, our technique can detect them. While modifying hyperparameters, I use it to evaluate model skill. Table 3 displays the training loss, training accuracy, validation loss, along with accuracy. Figures 3 and 4 indicate how accurate our model's training and validation are. Figure 3 shows the loss of training and validation. in Figure 4.

Table 3: Train, Validation loss, and Train, Validation Accuracy for each epoch

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	1.0733	0.6407	0.6124	0.7913
4	0.4495	0.8510	0.2514	0.9163
7	0.3234	0.8932	0.2019	0.9320
10	0.2939	0.9046	0.1766	0.9380
14	0.2473	0.9181	0.1686	0.9441

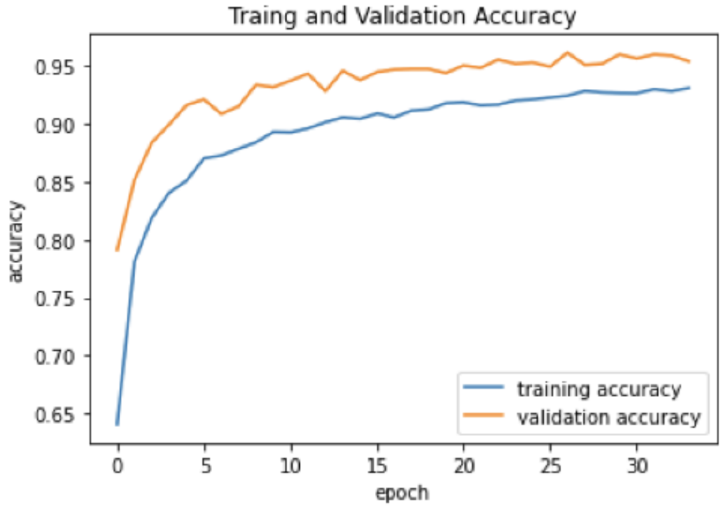

Figure 3: Training and validation accuracy

Figure 4: Training and validation loss

4.1 System Simulation and Output

We loaded the saved trained model into TensorFlow as a backend. While Python was used as the programming language and the editor, OpenCV was utilized to read the footage frame by a frame from a digital camera in an instantaneous loop. Our proposed system, as shown in Figure 5, displays recognized English alphabets on the screen.

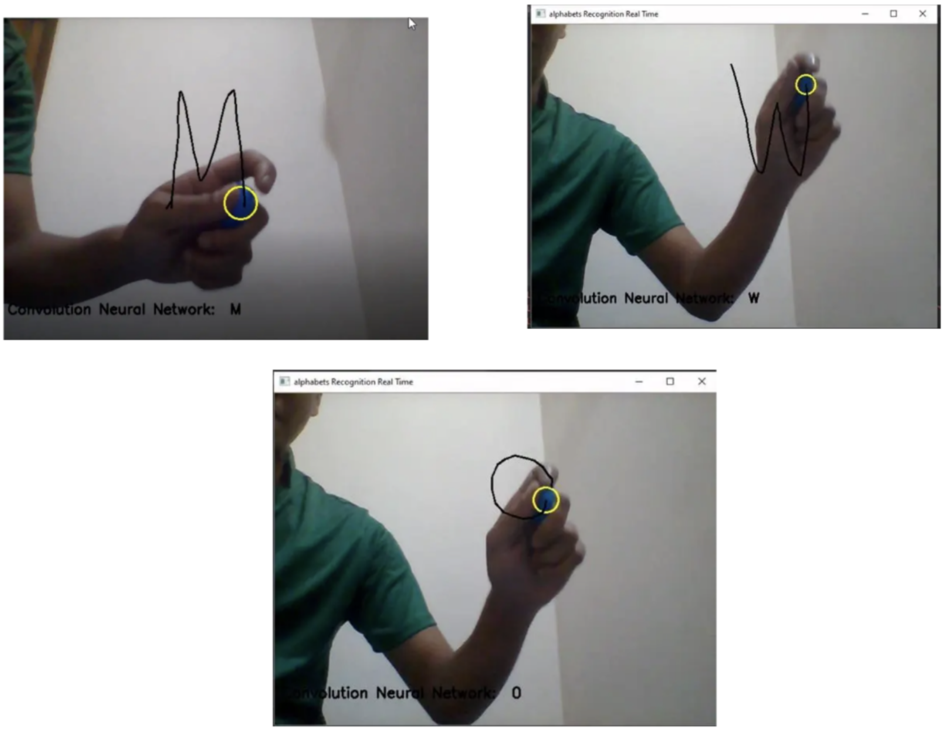

Figure 5: English alphabet (M, N, 0, W) output

5. Conclusion and Implication for Further Study

Despite advances in object recognition techniques, the challenge of Realtime English Alphabet Recognition through Hand Gestures (REARG) remains unsolved due to a number of issues. In reality, many of the most advanced existing techniques fail to deliver satisfactory outcomes. This work offers a CNN paradigm for handwritten digit recognition, written on air gathered using a camera, which functions well in real time in recognizing the majority of the input letters. The results show that the test accuracy rate for detecting 26 English alphabets is 93.08 percent. This clearly demonstrates that basic CNNs are capable of tackling even the most difficult categorization problems. Furthermore, this research endeavor will serve as a model for future projects that will shed light on this topic for future machine learning researchers. I intend to expand our research into Real Time English Character Recognition using hand gestures in the next stage of our work (RECRG). Recognizing handwriting in real time utilizing OpenCV is one of the remaining hurdles, even though I can retrain our system for the character dataset sections of the BanglaLekhaIsolated dataset. Our goal is to develop a technology that allows for greater human machine connection.

References

[1] P. R. a. S. S, "Online and off-line handwriting recognition: a comprehensivsurvey," IEEE Transactions on Pattern Analysis and Machine Intelligence22, pp. 63-84. https://doi.org/10.1109/34.824821

[2] S. A. J. T. a. A. V. T. Cohen G, "EMNIST: "Extending MNIST to handwritten letters," IEEE International Joint Conference on Neural Networks IJCNN, pp. 2921-26, 2017.

[3] H. B. a. B. H. Claus B, "Online handwriting recognition with support vector machines-a kernel approach," IEEE Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, pp. 49-54, 2002

[4] English at Ethnologue (22nd ed., 2019).

[5] Ethnologue 2010. https://doi.org/10.1080/17449057.2010.502305

[6] C. L. a. C. Y. Suen, "A new benchmark on the recognition of handwritten Bangla and Farsi numeral characters," Pattern Recognition, vol. vol. 42, no. no. 12, p. pp. 3287–3295, 2009. https://doi.org/10.1016/j.patcog.2008.10.007

[7] R. S. S. B. M. K. M. N. a. D. K. B. N. Das, ”A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application,” Applied Soft Computing, vol. 12, no. 5, p. 2012, 1592–1606. https://doi.org/10.1016/j.asoc.2011.11.030

[8] J. X. a. Y. L. J.W. Xu, ”Handwritten Bangla digit recognition using hierarchical Bayesian network,” in IEEE, November 2008., Xiamen, China, in Proceedings of 3rd International Conference on Intelligent System. https://doi.org/10.1109/ISKE.2008.4731093

[9] A. A. H. a. K. I. A. H. A. Khan, ”Handwritten Bangla digit recognition using sparse representation classifier,” in IEEE, Dhaka, Bangladesh, May 2014.

[10] B. D. R. S. S. B. M. K. a. M. N. N. Das, ”Handwritten Bangla basic and compound character recognition using MLP and SVM classifier,” 2010 https://doi.org/10.48550/arXiv.1002.4040

[11] T. L. C. J. R. B. B. O. a. N. M. Y. J. A. J. Alsayaydeh, ”Handwritten text recognition system using Raspberry Pi with OpenCV TensorFlow,” International Journal of Power Electronics and Drive Systems/International Journal of Electrical and Computer Engineering, vol. 15, no. 2, p. 2291, Jan. 2025. https://doi.org/10.11591/ijece.v15i2.pp2291-2303

[12] S. E. H. S. P. a. M. A. M. Ali, ”Real-Time recognition of NZ sign language alphabets by optimal use of machine learning,” Bioengineering, vol. 12, no. 10, p. 1068, Sep. 2025. https://doi.org/10.3390/bioengineering12101068

[13] A. A. A. A.-S. A. S. a. A. A. A. Qedear, ”Real-Time Air-Writing Recognition for Arabic letters using deep learning,” Sensors, vol. 24, no. 18, p. 6098, Sep. 2024. https://doi.org/10.3390/s24186098

[14] Y. R. S. N. M. a. S. M. V. Shravya, ”Air writing with Effective Communication Enhancement for Dyslexic Learners,” Procedia Computer Science, vol. 235, p. 2056–2068, Jan. 2024. https://doi.org/10.1016/j.procs.2024.04.195

[15] K. M. O. N. e. al, ”Recognition of Arabic Air-Written Letters: machine learning, convolutional neural networks, and optical character recognition (OCR) techniques,” Sensors, vol. 23, no. 23, p. 9475, Nov. 2023 https://doi.org/10.3390/s23239475

Authors

Fahmida Islam is the Chairman and Assistant Professor of the Department of Computer Science and Engineering at The People’s University of Bangladesh. Fahmida Islam is the Chair and an Assistant Professor in CSE, which is part of the Faculty of Applied Science. Her work in school is mostly about teaching, doing research, and leading the CSE department. **Email:** fahmida.islam@pub.ac.bd

In 2023, **Prome Saha Resha** got her Bachelor’s degree in Information and Communication Technology from Comilla University in Bangladesh. In April 2025, she got her Master’s degree in the same field. Her research interests include wireless communication, routing protocols, and methods for machine learning and deep learning. She is now a lecturer in the Department of Computer Science and Engineering at The People’s University of Bangladesh in Dhaka, Bangladesh. **Email:** promesaharesha5678@gmail.com