Air writing using Deep Learning

P. Rachana
Department Of Computer Science and Engineering
VVIT, Guntur, Andhra Pradesh, India

K. Prathyusha
Department Of Computer Science and Engineering
VVIT, Guntur, Andhra Pradesh, India

P. Babji
Department Of Computer Science and Engineering
VVIT, Guntur, Andhra Pradesh, India

A.Prashant
Assistant Professor
Department Of Computer Science and Engineering
VVIT, Guntur, Andhra Pradesh, India

ABSTRACT- Writing is a mode of communication that can effectively convey our thoughts. Today, typing and writing are the modes of storing information. Another technique that is rapidly gaining popularity is air writing recognition. It refers to writing characters or words in free space using a finger, which helps to lessen the gap between the physical world and the digital world. Our project mainly focuses on the characters written in the air and converts it to the text that would potentially bridge the gap for smart wearables to use air-writing. It is a combination of computer vision and handwriting recognizer. Air-writing is an optimized approach for deaf and dumb people.

KEYWORDS - Computer vision, convolutional neural networks, objectdetection, image classification.

I. INTRODUCTION

1.1 Introduction to air writing recognition

Handwriting analysis is a well-explored problem in machine learning that encompasses many of the important topics of neural networks. When we had seen the handwriting analysis, we realized that it could be combined with other concepts to traverse for implying more features to build a helpful application. Our project aims to use computer vision and deep learning to create a system that acts as a virtual whiteboard. Our model recognizes the characters written in the air and converts it to text. User would be able to write the characters in the air either by using the webcam or a video.

1.2 Historical Perspective

For the last decade, keyboards and mouse were used for the interaction of humans and computers. As generations change we use a variety of writing methods. Every time we use a different method it consumes less time. As in the last two decades, we used keyboard and mouse and then mobile phones and notepads to write. People who use computers or PCs to save their notes can't carry their computers. Even when people use mobile phones, it makes them lose their concentration to write in the notepad with such a small keyboard in the mobiles. And for the people who carry notes to write the notes, you know there will be problems with carrying the notes everywhere with a pen.

1.3 Latest scenario

Now coming to lecture halls, lecturers will find it difficult to write every time on the board or when making a presentation, etc., the solution for all these problems is air writing. We'll start from lecture halls since we are in college. Lecturers can write in the free space in lecture halls or while giving presentations. People, who use computers, mobile phones or notebooks, and pens, can use this air writing as this will be very efficient to use. Potentially it will be used in smart wearables where it uses camera vision to recognize the data. This project uses computer vision or any attached vision with this program can be acted as a receiver of the image and recognizes the image from its dataset.With the help of the OpenCV and CNN, the tip of the tracking object will be identified first. Then, the trajectory of this fingertip will be recorded and drawn on a black background. The resulting image will be fed to a character recognition model which will identify and output the drawn alphabet. Section 2 refers to the review of the other base papers. Section 3 provides the solution i.e., a method to solve the problem. Section 4 explains how to implement the solution. Section 5 explains the variables used. Section 6 shows the results which we got from implementation and Section 7 determines our work and scope of this project in the future.

II. PROPOSED METHOD

The Proposed solution is Air Writing Using Deep Learning. It helps to write in the free space and shows the output on the computer screen. This proposed solution is a better version than the reviewed research paper projects. Because here we use the technologies that are not there at the time of those projects which we researched. Because of this reason, those projects lack either faster computation or may take more time on the classification of images. Air Writing Using Deep Learning uses technologies that are developed in the current time which reduces the computational expenses and work in real-time. It uses CNN, Keras which offers a high-level neural network API that is consistent and simple. It reduces the number of user actions required for common use cases and provides clear and actionable error messages. Markov Chaining shows the predicted words as we enter every letter.

III. IMPLEMENTATION

The project implementation can start only after the system meets the requirements for the project. Therefore the data related to the project implementation should be available and then implementation can be started. The project implementation period can be defined by the time taken to train the model.

- Understanding the requirements for the project.
- Designing and developing the solution.
- Initiating and planning.
- Training the data model.
- Executing or implementing the developed code.
- Compiling the content parts of the project.
- Controlling and monitoring the input and output data.
- Evaluating the code by quality checking of results.
- Achievement of project objectives and quality control of delivered outputs.

IV. MODELS AND ALGORITHM:

This Project enables air writing in which the user can write in the air and the word written in the air is recognized. It has two main components first one is tracking the object and its motion, second one is processing the motion of the object.

4.1 Object Tracking

A) Setting the values of the object

We have to set the upper and lower HSV values of a color to detect the object of the particular color. In our case, we try to detect an object of light green color so the HSV values are (29, 86, 6) and (64, 255, 255).

B) Motion capturing

We would try to recognize the motion of an object in a video which could be a webcam or a video source. A video is a collection of many frames moving with certain frames per second speed. We would process each frame in the video by using an OpenCV library. Object detection is done by setting the values. After the object is detected, we will process the motion of the object.

C) Storing the required values

We try to store the required values by masking with the help of the OpenCV library. A mask allows us to focus only on the portions of the image that interests us.

D) Processing the motion

We process the frame by erosion and dilation and then find the number of contours. Dilation process adds pixels to the boundaries of objects in an image, whereas erosion process removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the image. Contours can be explained simply as a curve joining all the continuous points, they are similar to edges. The contours are a used for shape analysis and object detection and recognition. For better accuracy, we used binary images.

E) Segmentation Problem

We identified the types of characters of 10 digits and 26 uppercase letters, and 26 lower cases. When working on gesture tracking, we realized that many of the characters we want to recognize required multiple "pen strokes" or segmented lines to be accurately placed for accurate differentiation for both computer and human perception.

4.2 Dataset

The EMNIST dataset has set of handwritten letters which are both upper and lower case that are being converted into 28x28 pixels from the original images of 128x128 binary images. The dataset provides EMNIST byclass containing 814,255 characters and 62 unbalanced classes.

4.3 Handwriting Classification:
The written character on the blackboard is pre-processed 28x28 sized image (figure 3) and passed to the next step i.e., trained Convolutional Neural Network (CNN). In our CNN model, we will introduce the two layers that build the convolutional networks that can be possible with two operations i.e., convolution and pooling.

- *Convolution Layer:*
  In convolution layer, each neuron of the hidden layer is connected to a small region of 3x3 neurons of the input image and generates a size output (26, 26, 32) a 3D tensor containing the 32 outputs of 26x26 pixel result of computing the 32 filters of the input(figure 1). The activation function used in this layer is ReLU or Rectified Linear Activation(figure 1) which works well in neural networks even though is two linear pieces.

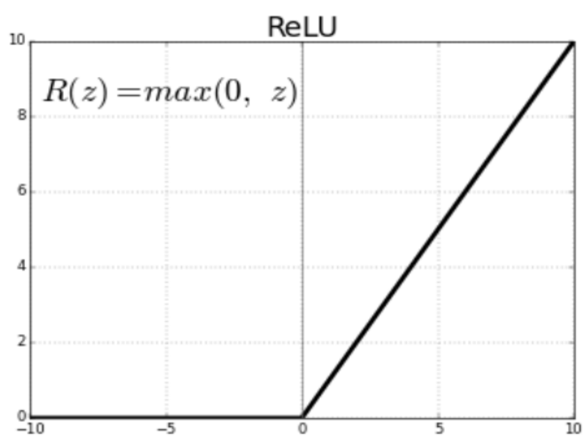

Figure 1: ReLU Activation Function

- *Max pooling*
  The pooling is applied immediately after the Conv2D layers. It condenses the information by choosing the max pooling 2x2 window of the convolutional layer and finds the maximum value in the window. Max pooling works better than other alternative solutions.

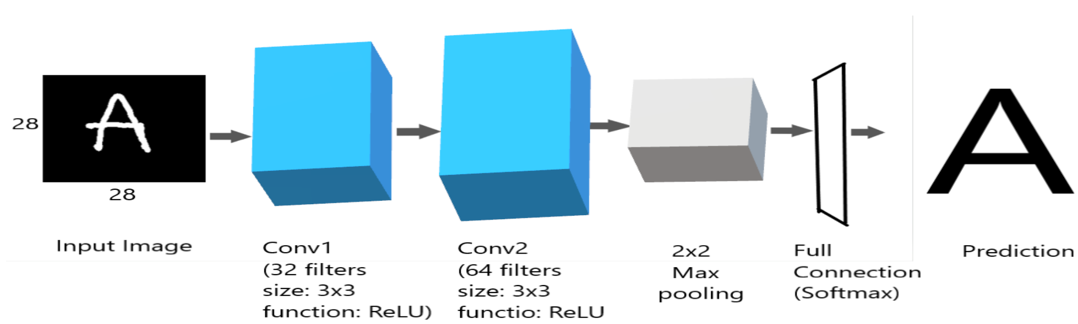

Figure 2: Network Architecture

Our model contains two convolutional layers (Conv2D), one max pooling layer, a dropout layer and a flatten method. Here to adjust the tensors, flatten method is used to convert the 3D tensor to 1D before applying the softmax activation function. The output image from the two convolutional layers and flatten method is added to a densely connected layer, which will serve to feed a final layer of softmax activation function.

$$P(y = j|\theta^{(i)}) = \frac{e^{\theta^{(i)}}}{\sum_{j=0}^{k} e^{\theta_k^{(i)}}}$$

$$\text{where } \theta = W_0X_0+W_1X_1+\cdots+W_kX_k=\sum_{i=0}^{k} W_iX_i=W^TX$$

The above equation represents Softmax Activation function.

Algorithm Adadelta

- Require: Decay rate ρ, constant ϵ
- Require: Initial parameter θ
- Initialize accumulation variables s = 0, r = 0
- **while** Stopping criterion not met **do**
  - Sample a mini batch of *m* examples from the training set $\{x^{(1)},\ldots,x^{(m)}\}$.
  - Set g = 0
  - **for** i = 1 to m **do**
    - Compute gradient:

$$g \leftarrow g + \nabla_\theta L\left(f\left(x^{(i)};\theta\right)\right), y^{(i)};\theta$$

end for

Accumulate gradient: $r \leftarrow \rho r + (1-\rho)g^2$

Compute update: $\Delta\theta \leftarrow \frac{\sqrt{s+\epsilon}}{\sqrt{r+\epsilon}} g$ (operation applied element-wise)

Accumulate update: $\theta \leftarrow \rho\theta + (1-\rho)(\Delta\theta)^2$

Apply update: $\theta \leftarrow \theta + \Delta\theta$

- **end while**

4.4 Pre-processing

In an EMNIST dataset each value of pixel ranges from 0 to 255. An image of pixels of 0 represents a black image and 255 represents a white image. Neural networks increase their working speed if the range is [0,1] rather than [0,255]. So, in pre-processing we change the numbers in range.

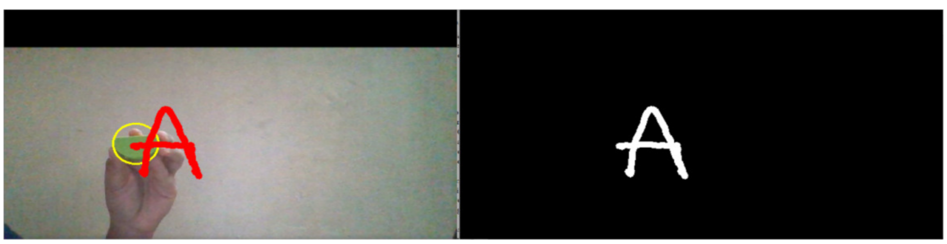

Figure 3: Input Generation

4.5 Hyperparameters

CNN's performance depends on many hyper-parameters namely CNN depth, number of filters, and their respective sizes. The number of hidden layers and units are set before the training as in our model the first convolutional layer contains 32 filters and later the parameters are increased in the second convolutional layer. The 128 nodes are used as a hidden layer for the fully connected layer. So, The Hyperparameters value gives less error and consequently the higher classification accuracy.

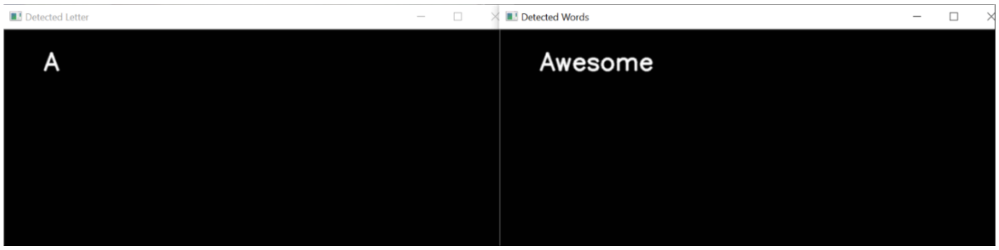

Figure 4: Output Generation

4.6 Obtained result

The target object is detected in the frame and the motion of it is observed and stored. The movement of it is then performed forsome morphologicalimage processing operations, the character is then sent to the neural network for prediction. The letter written in the air is detected by using the eminst dataset and then displayed on the window (figure 4). The corresponding word of the letter is displayed in another window.

V. VARIABLES USED

1. **S -** Ifthe 's' key is pressed, consider this the "start".
2. **Q -** If the 'q' key is pressed, stop the loop.
3. **D -** If the ‘d’ key is pressed, the movable object detection is started.

VI. RESULTS

The target object is detected and the motion of it is observed and the path of the movement of target object is sent for the identification of the character to the model. The model then identifies the corresponding character with the help of EMINST dataset and it displays the corresponding character. If the character is a letter, then a corresponding word is displayed on another frame. After detecting the input, the detected word is displayed in the output frame to the corresponding detected letter and is shown in the below figure

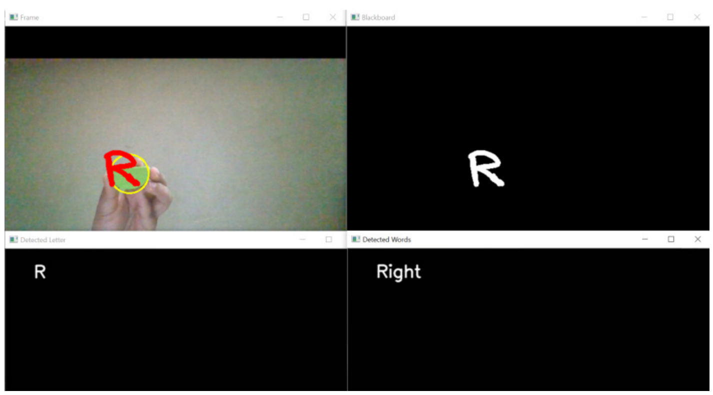

Figure 5: Final Output

When we trained our neural network on 731,668 samples of EMNIST dataset, it showed training accuracy percentage of 86.71% and an evaluation accuracy percentage of 90.43%. And the training loss percentage is observed as 40.93% and an evaluation loss percentage is noted as 35.14%

VII. CONCLUSION & FUTURE SCOPE

In this system, we proposed a very simplified solution for air writing. The air writing enables the users to write in the air and the motion is converted to the text. This air writing has the potential to challenge traditional writing it prevents the need to carry a mobile phone in hand to join the notes. This serves a great purpose in helping especially disabled persons to communicate easily. This technology can be incorporated into IoT devices. This can be used in biometric system verification even in which the user’s authentication can be done in the air itself preventing the user to touch the surface, which could be a huge help in the situations like Covid. This technology can even be improved by providing the gestures symbols used by the deaf and dumb people which helps them to communicate with the people.

There are certain limitations in Air Writing with Deep Learning, which can be improvised in the future. We can use the external webcam for clear pictures which makes the identification of the character mucheasier. In the future, using the YOLO V3 will improve the efficiency of air-writing.

REFERENCES

The references of the paper are provided below and we used EMNIST dataset to train the system.

[1]. P. Ramasamy, G. Prabhu and R. Srinivasan, "An economical air writing system converting finger movements to text using web camera," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, 2016, pp. 1-6, doi: 10.1109/ICRTIT.2016.7569563.

[2]. P. Anjaneyulu, Y. Jampaiah, R. Karthik and T. Vijetha, "Air writing recognition modeling and recognition of characters, words and connecting motions," 2017 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, 2017, pp. 1112-1115, doi: 10.1109/ISS1.2017.8389354.

[3]. T. Tsai and J. Hsieh, "Air-writing recognition using reverse time ordered stroke context," 2017 IEEE International Conference on Image Processing (ICIP), Beijing, 2017, pp. 4137-4141, doi: 10.1109/ICIP.2017.8297061.

[4]. P. Puranik, T. Sankhe, A. Singh, V. Vishwakarma and P. Rane, "AirNote – Pen it Down!," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-7, doi: 10.1109/ICCCNT45670.2019.8944690

[5].Xiaoping Yun, M. Lizarraga, E. R. Bachmann and R. B. McGhee, "An improved quaternion-based Kalman filter for real-time tracking of rigid body orientation," Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), 2003, pp. 1074-1079 vol.2, doi: 10.1109/IROS.2003.1248787.

[6] Welch, G.; Foxlin, E. Motion tracking: No silver bullet, but a respectable arsenal. IEEE Comput. Graph. Appl. **2002**, 22, 24–38.