2022 Intel Cup Undergraduate Electronic Design Contest

- Embedded System Design Invitational Contest

Final Report

Intel Cup Embedded System Design Contest

Project Name：American Sign Language Recognizer

Students: Shubham Kumar, Rajan Patel, Dheeraj Kumar

Faculty: Mrs. Jhuma Saha

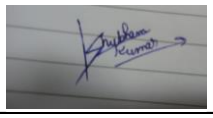
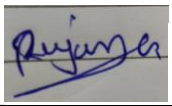
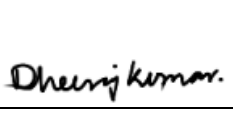
University: Indian Institute of Technology, Gandhinagar

2022 Intel Cup Undergraduate Electronic Design Contest

- Embedded System Design Invitational Contest

Declaration of Originality

We hereby declare that this thesis and the work reported herein was composed and originated entirely by ourselves. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given in the references.

Team Members Signature:   

Name (in Block Letters): SHUBHAM KUMAR, RAJAN PATEL, DHEERAJ KUMAR

Date: 26TH July 2022

ABSTRACT

The project is to recognize American sign language using, SEAVO CPU and DE-10 nano-board. The people who are deaf and mute, can communicate with each other with American sign language but people who can speak and hear are not able to understand the American Sign language much. Our objective is to break this barrier of language among peers. Thus, we have developed a system which can recognize the sign language done by hand gestures and display it on the digital medium. The main aim is to enhance the communication for the specially challenged human beings. In an effort to do so, we have developed a model based on neural networks, and have implemented it on DE-10 nano-board. Input given to the network is in the form of images (could be taken through cameras or manually provided), and the output is displayed on a digital platform (Seven-Segment LEDs). The model can recognize all the alphabets excluding letter “Z” (as it is not possible to display on Seven-Segment LED). Both uppercase and lowercase letters have been implemented. The implemented model shows an accuracy of ~ 93%. We believe that this project is a step towards building effective communication medium for the specially challenged human beings, and finds potential application in surveillance systems, education purposes, portable hand held digital device, and many more. This project could be extended for accepting video inputs and to recognize various sign languages.

Key words: American sign language, neural network, seven-segment LEDs, DE-10 board

Content

Chapter 1: Introduction and motivation

1.1 Introduction

Sign language is a way of communication for the specially challenged human beings who are unable to speak. Several sign languages have been developed by several countries such as American Sign Language (ASL), British Sign Language (BSL), French Sign Language (LSF), and so on. These sign languages help the specially challenged persons to communicate effectively. As we are living in a digital era, therefore it makes sense to develop a system which can recognize the sign language done by hand gestures and display it on a digital medium. The sign language recognizer could be installed in various surveillance systems, airports, and for educational purposes. This would not only help the specially challenged persons to effectively use the digital gadgets/systems but would also make various digital platforms to be inclusive in nature.

In this project, we have developed a system which can recognize the sign language done by hand gestures and display it on the digital medium. The main aim is to enhance the communication for the specially challenged beings. In an effort to do so, we have developed a model based on neural networks, and have implemented it on DE-10 nano-board. Input given to the network is in the form of images (could be taken through cameras or manually provided), and the output is displayed on a digital platform (Seven-Segment LEDs). The model can recognize all the alphabets excluding letter “Z” (as it is not possible to display on Seven-Segment LED). Both uppercase and lowercase letters have been implemented.

1.2 Brief history and motivation

ASL is a complete language like other spoken languages with grammar of different type. This language is specially made for deaf and have problem of hard of hearing. This language is expressed by the movement of hand and face expression. There is no universal sign language. The ASL is specially made by north Americans and on the other hand there is BSL which was made by the British people. The sign language was roughly developed around 200 years ago. Because this language is very old and many people are using it, it is evolved very much and with time it got more complex and unique. Over time, it gets mixed with other sign languages such as LSF. However, we have to take care take that Modern ASL and Modern LSF both are different languages. But the ASL is completely different from spoken English language. It has its own rules and fundamentals. Just like different regional people have different accents while speaking, ASL users also have some different hand symbols which changes regionally. NIDCD supports the ASL and also gives funds on research on ASL and other sign languages. It has been observed that there is a barrier created between the people who do not speak ASL and those who do. To break this barrier, we bought the idea of making a system to understand the sign language by everyone.
Here, we have developed a system model based on Convolution Neural Network made in SEAVO and DE-10 nano board, which can recognize the sign language done by hand gestures and display it on the digital medium.

Chapter 2: Theory and Design

2.1 Neural Networks

Neural Networks is the backbone of AI based algorithms. Neural Networks are subsets of Machine learning (ML) and we can say that these are the base of the Deep learning algorithms. So actually, neural networks as per the name it is inspired by the brain neural structures. Each neuron is connected to one another and all the functioning of brain is done by these networks. The AI neural networks work on this principle. The types of neural network contain, Convolution neural network, Artificial neural network, Deep neural network etc. Neural network has multiple layers and these layers has multiple neurons. These layers contain, input layer, output layer and the hidden layer. Figure 1 shows a simple neural network [1]. So, there could be any number of neurons (>=1) in one layer. There could be multiple inputs and corresponding to that there could be multiple outputs. The hidden layers do most of the work. The hidden layer connects input and output and performs multiple functions and predict the right output.[5]

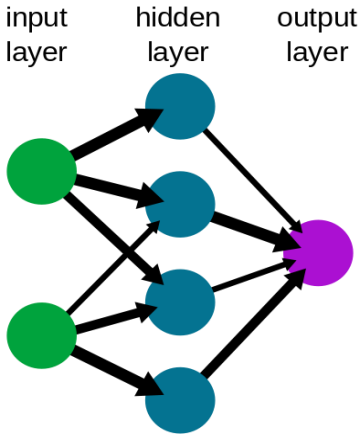


Figure 1: A simple neural network

Neural networks have many different types and it is a very vast topic in itself and we took a small part out of this big domain of neural network, which is convolution neural network, (mostly known as CNN). So, in a neural network there are many neurons and each neuron connected to other through connector this connector is known as weights. These are called weights because when one network is connected to other, we can change the strength of these connection, that is how much strong these connections are made, more the weights more will be strong the connection. They are many other neural networks which one can use accordingly as per their requirements.

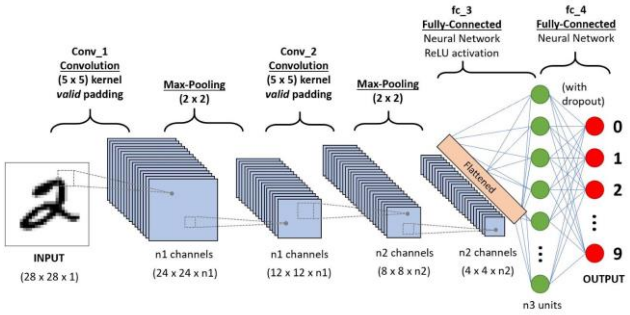


Figure 2: Convolution neural network

Figure 2 shows the structure of Convolution neural network [2]. The values shown in the picture (2), we will discuss them further.

In convolution neural network there are filters which are also known as the kernels. These kernels further convolute with each layer to generate output. Filters, as per the name, they filter out our image/inputs. Here we mean like if we want detect an edge of a particular image, the CNN will pass the image through filter and this filter has a fixed size with some numerical value. The parameters like max pooling, activation function, average pooling, reshape.

2.1.1 Maxpooling Function

In our model, for the first layer we are using 32 filters or kernels which has a size of 3 x 3. In the second layer we are using 64 filters and in third layer we are using 128 filters of same size. We also use Maxpooling2D function of size 2x2. So, in Maxpooling function, we make a matrix of specific size and in that size, we pull out the max number made our layer matrix. Figure 3 shows the functioning of the Maxpooling function.

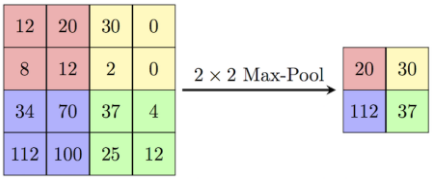


Figure 3: Structure of maxpooling

2.1.2 Activation Function

Activation function is the mathematical equations that applied to model just before the output layer. It helps to normalize the output in the range of -1 to 1 or 0 to 1. it also affects the convergence ability of neural network.

We are using ‘ReLU’ as the activation function. The ‘ReLU’ means rectified linear activation function. It is remains zero for the negative values and it has rectified nature for the positive number. There are also some activation functions like linear activation function, sigmoid function.

2.1.3 Dense Layers

Our model also contains Dense layers. In Dense layers mean a layer which is connected to its next layer deeply. This simply means every neuron of dense layer is connected to its preceding layer. This layer is mostly used in artificial neural network, but for concern we used it in our CNN.

We are also using dropout function that randomly shut down some neurons of the network. The fraction of neuron was shut down was known as dropout rate. The remaining neurons get multiplied so that the overall sum of the neurons will remain same.

2.1.4 Conv2D

So, this inbuilt function makes a 2-dimensional layer(kernel), which then convolve with in the input layer to produce a tensor output. This function takes arguments like filters, kernel size, padding, data format, dilation rate, groups (if any), activation function etc.
Our model was taken from

2.1.5 Adam Optimizer

Optimizers are used to change and get better control on the attributes like weights, learning rate. In our project we are using the 'adam' optimizer. The word Adam is derived from the adaptive moment estimation. It is a replacement of the stochastic gradient optimizer algorithm. It has better accuracy than the stochastic gradient in most of the neural network models.

It involves two gradient descent methodology which are momentum and Root mean square propagation (RMSP). Our neural network model was inspired from github [4].

2.2 Design of the project

We have developed a model which can predict the alphabets accurately. The images of ASL were given as input and the output was the corresponding English letters. First, a convolution neural network was built in python using google collab. We used many libraries: Tensorflow, cv2, keras, matplotlib, numpy and pandas. Among the above mentioned libraries, the most used library was “TensorFlow”. In TensorFlow we used Keras to build our network.
The Neural Network was built and was tested with some test data. The test data was an image, which was further converted into 28 x 28 matrix array. The developed model has an accuracy of 93%. Figure 4 shows the accuracy graph of the Training set and the test sets.

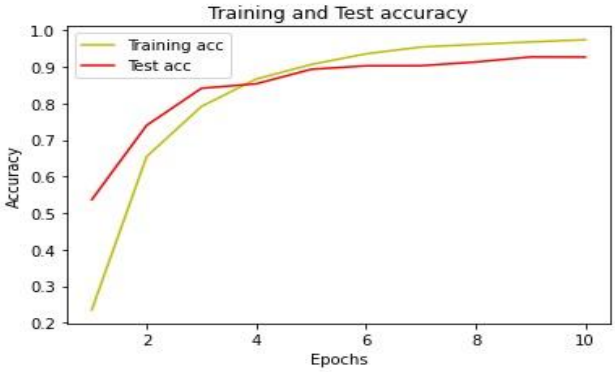


Figure 4: The accuracy graph of the model

Chapter 3: Implementation and Results

3.1 Results and discussions

So now it was time to put some random image or an image/word of our own choice. First, we user has to give the length of the word they want to display and then, we display the word. To display the word, we first take our input image and then convert it into an array format. After converting it into array/pixel format, we first convert it into 784 size matrix and then to 28 x 28 2D matrix. We reshape our image because our model takes input image having dimensions of 28 x 28. So, after that we take many words like ‘is’, ‘am’, ‘Shubham’, ‘intel’ etc. and we get the correct image output.

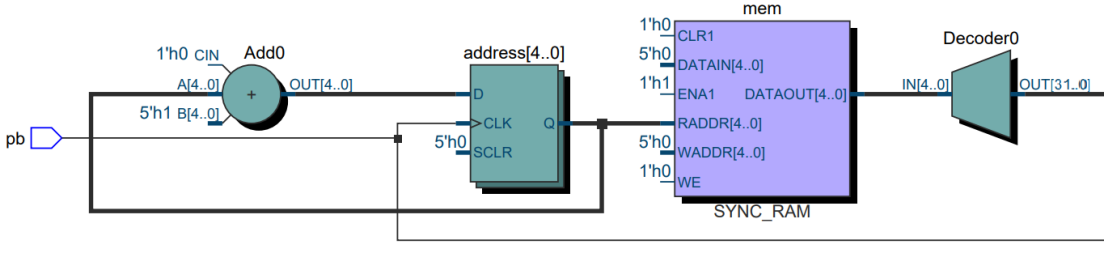
So, this was our result of word INTEL and it was giving us as expected output. We then created an array of words which have binary values of that word, e.g. if the word is predicted is “A” then the value will get store in file will be “00001”, which is binary representation of array. We are appending these values in array and then storing in a particular folder.

3.2. Working on DE-10 nano

As we discussed above, we generate the text file which contains the binary data. Then in our Verilog code we give the push button as the input. On every press of push button, we change our output, i.e. we go to the next state. Then, we read this binary data by the help of the inbuilt function ‘$readmemb’. Then, we map our binary data with the pins of seven segment LEDs. We are using two seven segment LEDs in our project. It is not possible to display all the alphabets in one seven segments LED. The 2nd seven segment LED is oriented at the right angle of the first seven segment LEDs because we are displaying the alphabets like "n", "M ","W " on the second LED.

Then, we give our binary data to GPIO pins. We are using vectors of GPIO pins for each seven segment LED. Length of each vector is 7. This vector array is connected to respected Pins of the seven segment LEDs. We are also using push button in our project. We use it to execute the next set of instructions (reading the next binary data and assign pin values accordingly). After all the letters are executed. The program will go in default mode. Both of the seven segment LEDs are off in default mode.

The developed neural network model was converted into a Verilog code, so as to dump it into the FPGA board. Different modules of the block diagram (RTL diagram) of the FPGA implementation has been shown in Figure 5 for better clarity, and the complete implementation is shown in Figure 6. Figure 7 shows the FPGA board connection with seven segment LEDs. Few resistors of 22 Ω were connected between pins of the LED and GPIO Pins.



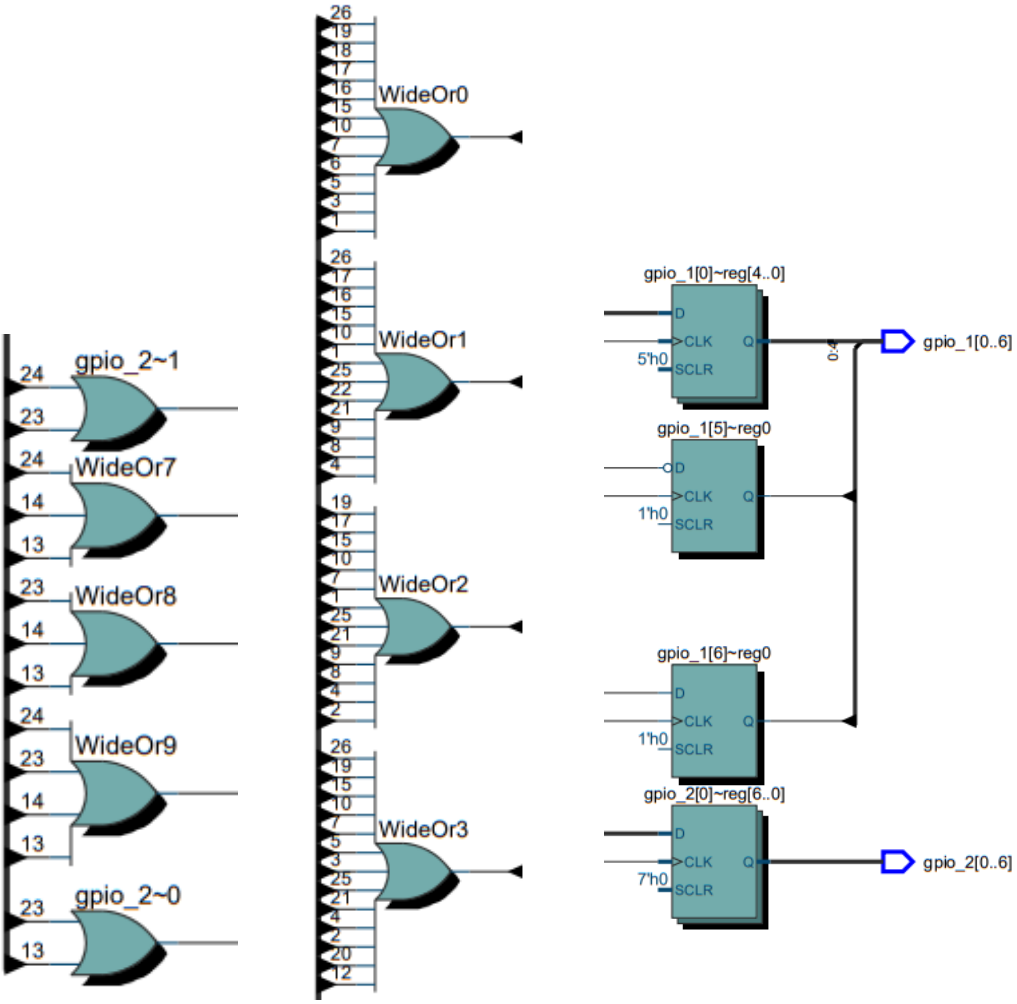


Figure 5: Different modules in the block diagram of FPGA implementation

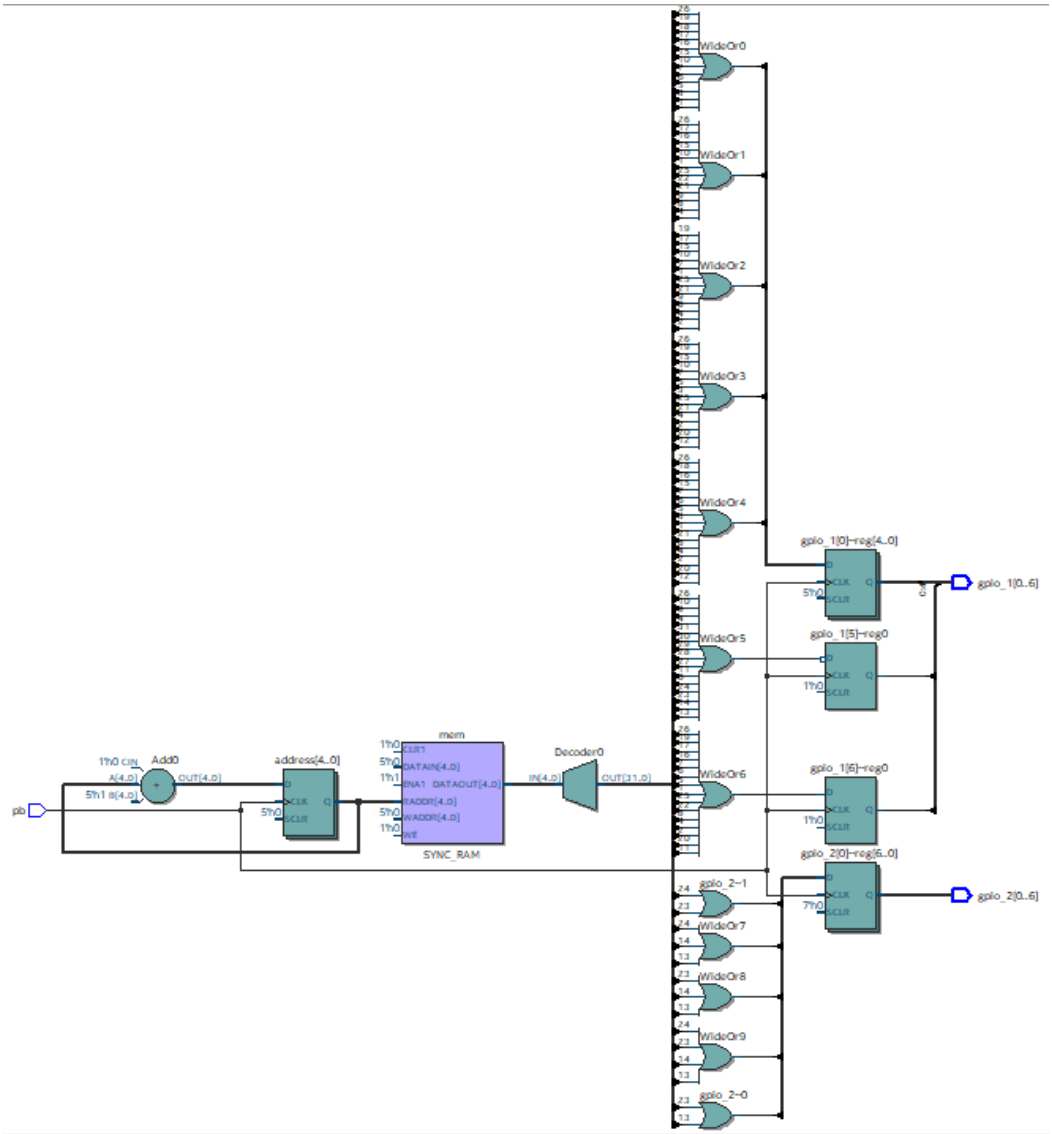


Figure 6: Complete block diagram of FPGA implementation

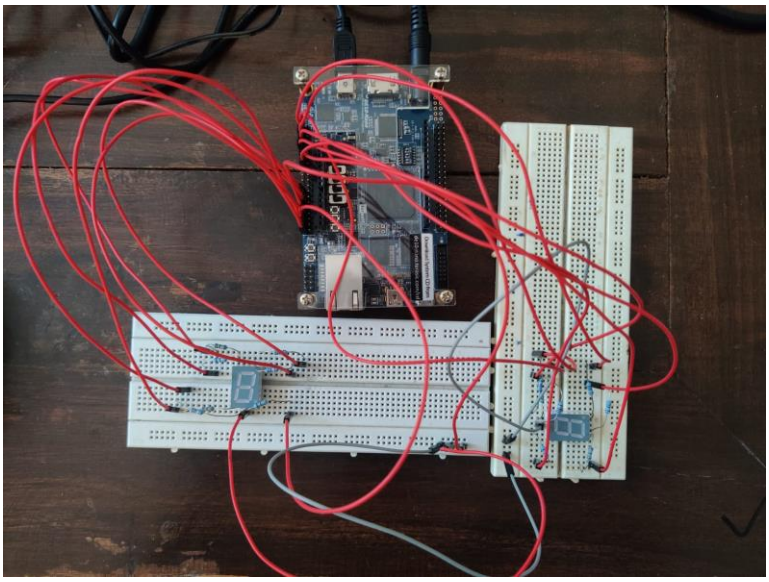


Figure 7: FPGA connection with seven-segment LEDs

The Output of convolution neural network is shown in Figure 8.

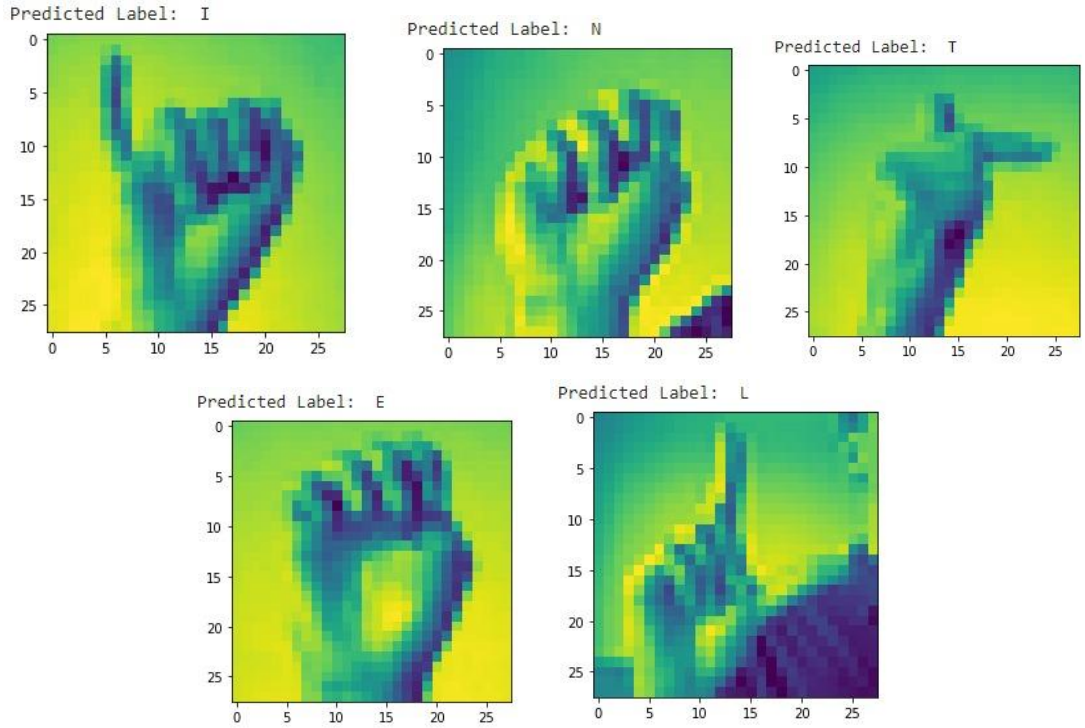
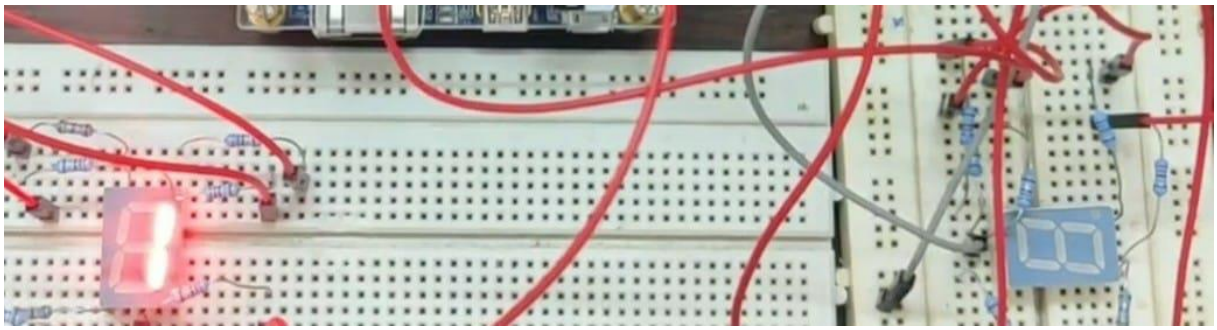
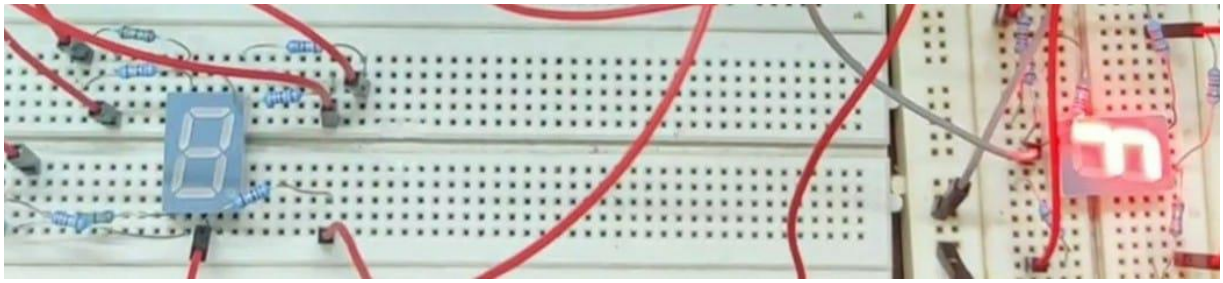


Figure 8: output of the Convolution Neural Network

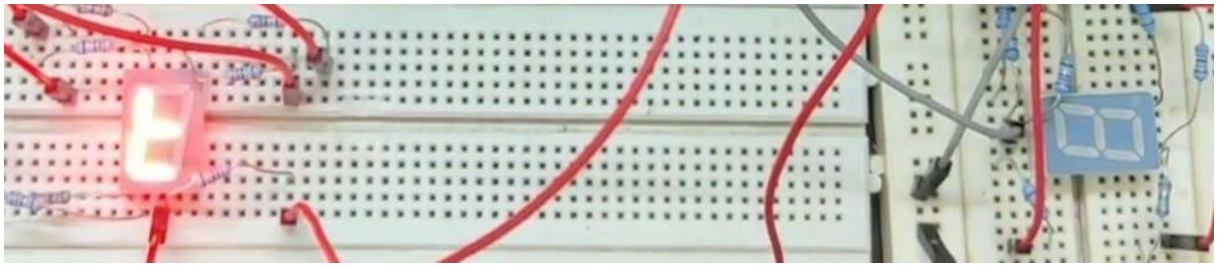
The Output of DE-10 Nano board for the corresponding input images (shown above) are displayed in Figure 9.



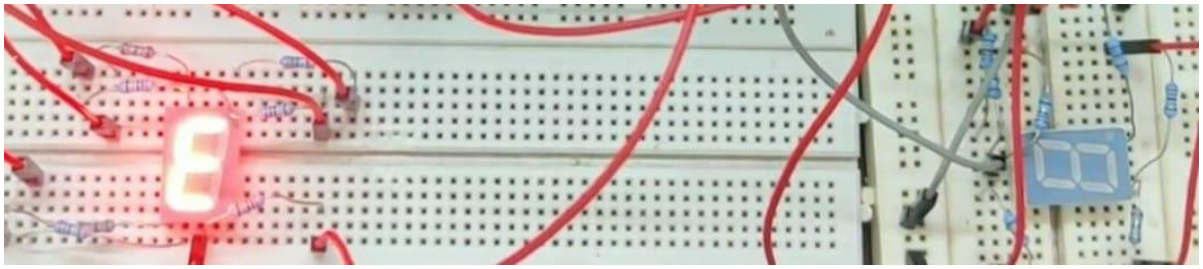
The output of the word “I”. (Capital letter I)



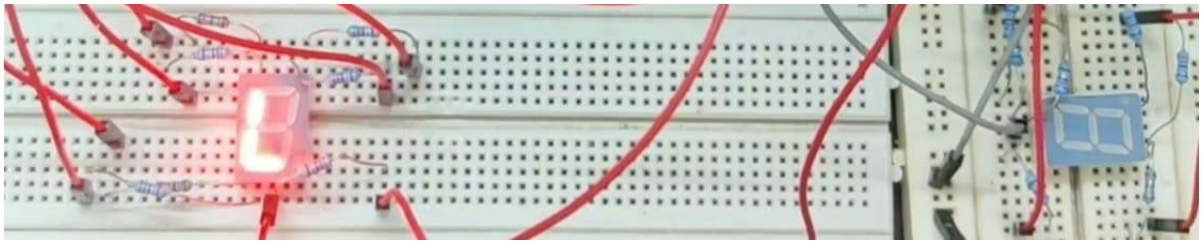
The output of the word “n”. (Small letter n)



The output of the word “t”. (Small letter t)



The output of the word “E”. (Capital letter E)



The output of the word “L”. (Capital letter L)

Figure 9: Output of the letters displayed on seven segment LEDs

Chapter 4: Conclusion

We have successfully developed a neural network based model which could recognize American sign language and have implemented it using DE-10 nano board. The model could be further extended for video inputs and to recognize various other sign languages. This project is a path way towards building effective medium of communication for specially challenged human beings. This work finds potential application in Surveillance systems, education purposes, Airports, etc.

References

[1] https://en.wikipedia.org/wiki/Neural_network

[2] https://link.springer.com/article/10.1007/s00500-020-04860-5

[3] https://paperswithcode.com/method/max-pooling

[4] https://github.com/bnsreenu/python_for_microscopists/blob/master/212-sign_language.py

[5]https://www.ibm.com/cloud/learn/neural-networks#:~:text=Neural%20networks%2C%20also%20known%20as,neurons%20signal%20to%20one%20another.