

# **American Sign Language Detector Using Convolutional Neural Networks**

Angad Anil Gosain(5490264)

Siddhant Shirodkar(5493740)

December 5, 2022

## **Abstract**

In the following project, we apply computer vision and deep learning concepts to recognize the American sign language(ASL) through a camera. We make use of Convolution neural networks in order to train our dataset. Our model is capable of recognizing the alphabet in the ASL format and present the user with an output in the form of text. In order to support our prediction we also calculate the confidence about our model's prediction, which indicates the percentage of the accuracy of the recognized hand sign. We make use of the MINST dataset, which gave us vivid set of image data. We present our output, which validates the prediction made by our model and gives us a room for improving the model.

## **Introduction**

Sign Language is considered as an important form of communication among a several groups of people. It allows us to remove the barriers of communication between people. Sign language is widely used among unfortunate people with disabilities. Sign language becomes an important factor for dumb, deaf and blind people. It allows them to communicate more easily. However this barrier is still standing when the communication between both communicators is different. The power of computer can allow us to solve this problem. The intricate and novel research in the field of computer vision and deep learning has given a chance to solve this challenge with a greater success. Since it is difficult for normal people to understand sign language, computer vision can allow us to recognize the hand sign and communicate the data into a more readable format for the user which is English text.

## Method

Currently to tackle this problem there are several existing solution such as learning the sign language which is a quite mundane solution as it can take up time and money and cannot produce an efficient solution. Another way could be to have a live assistant. However with this solution it can be very difficult to give a result which will be satisfactory considering the time and availability and capital factors. Thus for our solution we make use of the deep learning concept of CNN which creates the model which trains on a particular train dataset and then tests the model on the test dataset. The idea of using computer vision and CNN is to simplify the process of understanding the sign language.

Our method can be explained by the following steps-

- 1) We use a camera device in order to obtain image input from video.
- 2) Our trained CNN model is using the input data it gets from the video snippet.
- 3) Our model then returns 3 prediction with a confidence percentage for evaluation.
- 4) Our model detects the image data from the snippet based on the training done through the dataset and returns the output.

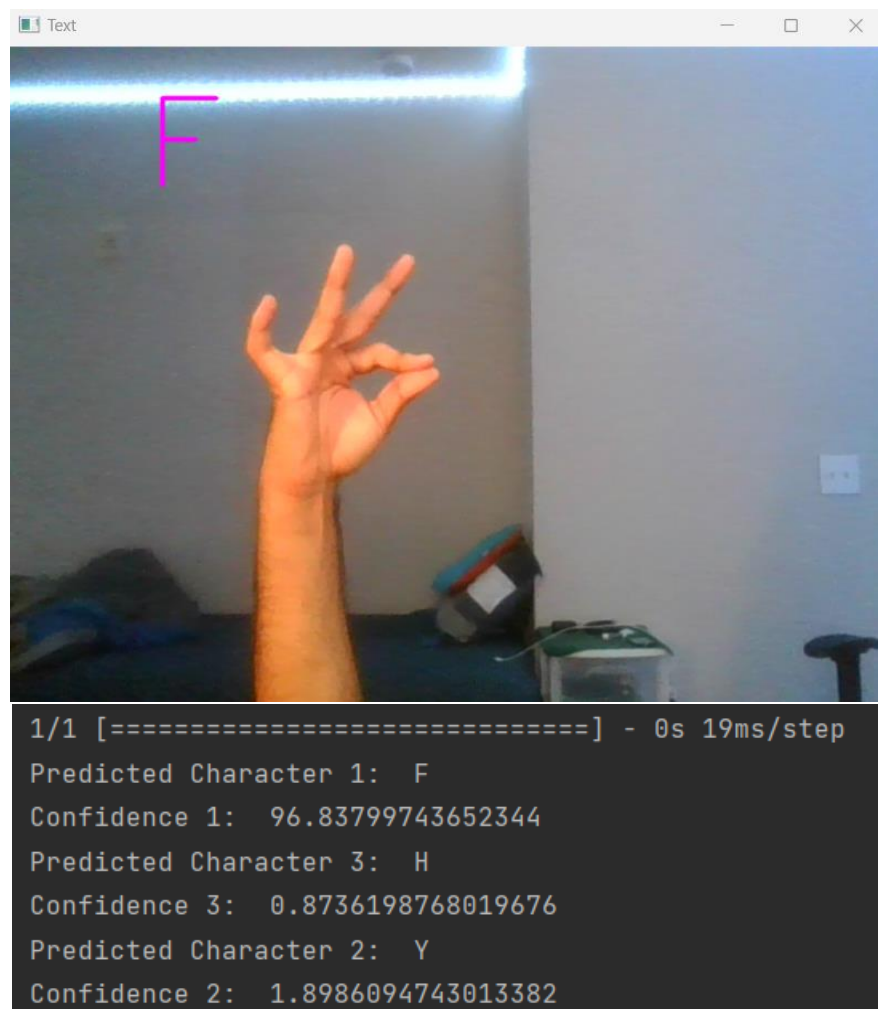
## Dataset



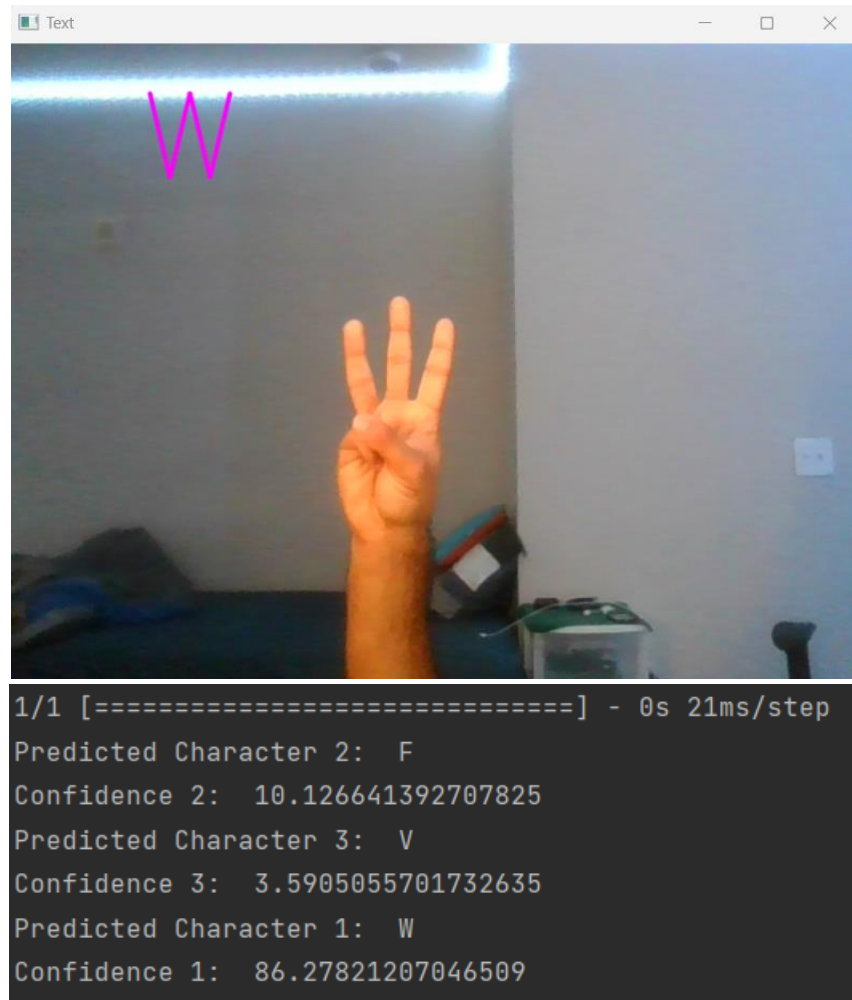
The dataset used in our project is MNIST American Sign Language presented by Kaggle. Each training and test case represents a label (0-25) as a one-to-one map for each alphabetic letter A-Z (and no cases for 9=J or 25=Z because of gesture motions). The training data (27,455 cases) and test data (7172 cases) are approximately half the size of the standard MNIST but otherwise similar with a header row of label, pixel1,pixel2....pixel784 which represent a single 28x28 pixel image with grayscale values between 0-255.

## Results

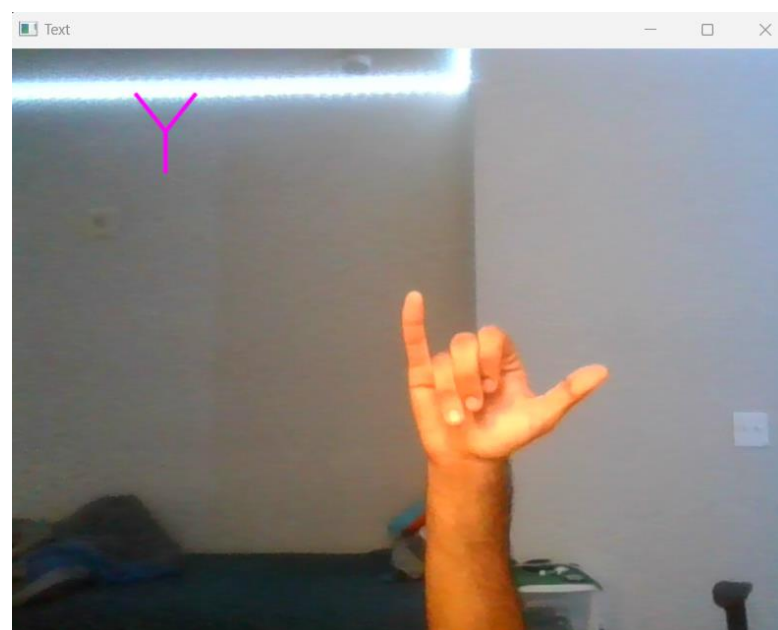
- Results with 1 Convolution Network Layer



Letter F Output



Letter W Output




```

1/1 [=====] - 0s 17ms/step
Predicted Character 3:  A
Confidence 3:  3.0131943162814423e-06
Predicted Character 2:  I
Confidence 2:  0.0013090566426399164
Predicted Character 1:  Y
Confidence 1:  99.99868869781494

```

Letter Y Output

- Results with 2 Convolution Network Layers



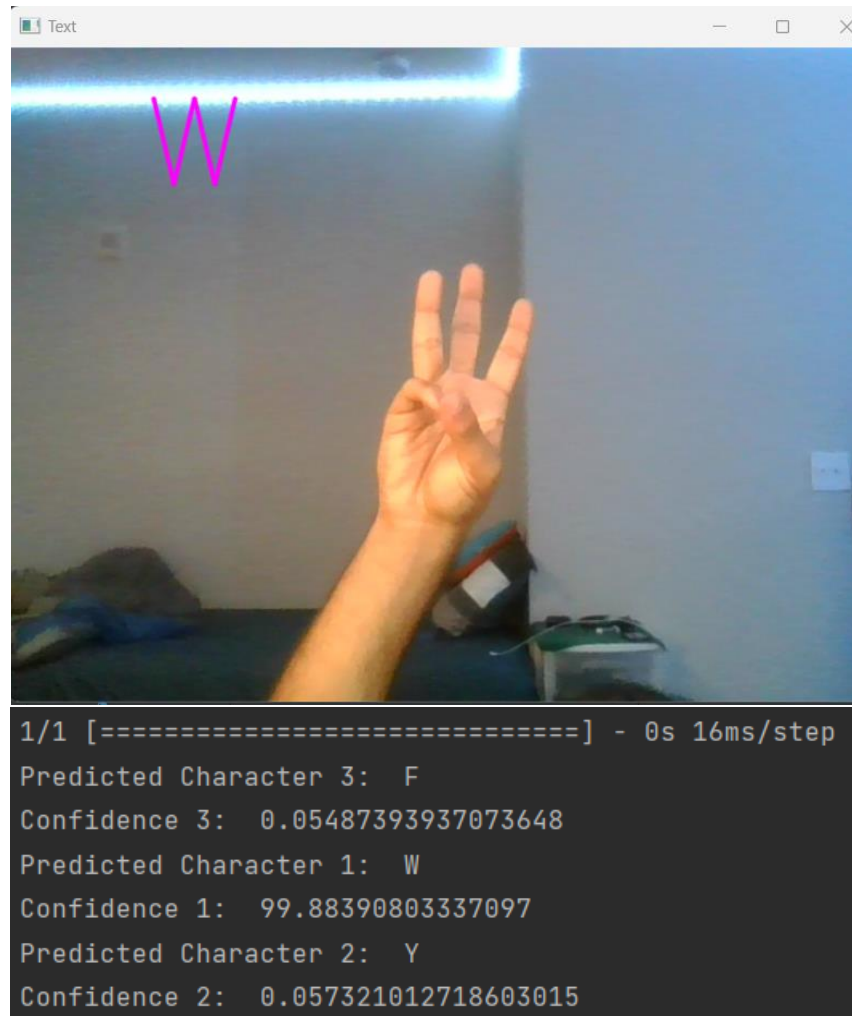
The image shows a video frame from a window titled 'Text'. It depicts a person's hand in a gesture that resembles the letter 'F'. A bright pink 'F' is overlaid on the top left of the frame. The background is a dimly lit room with a blue light source at the top.

```

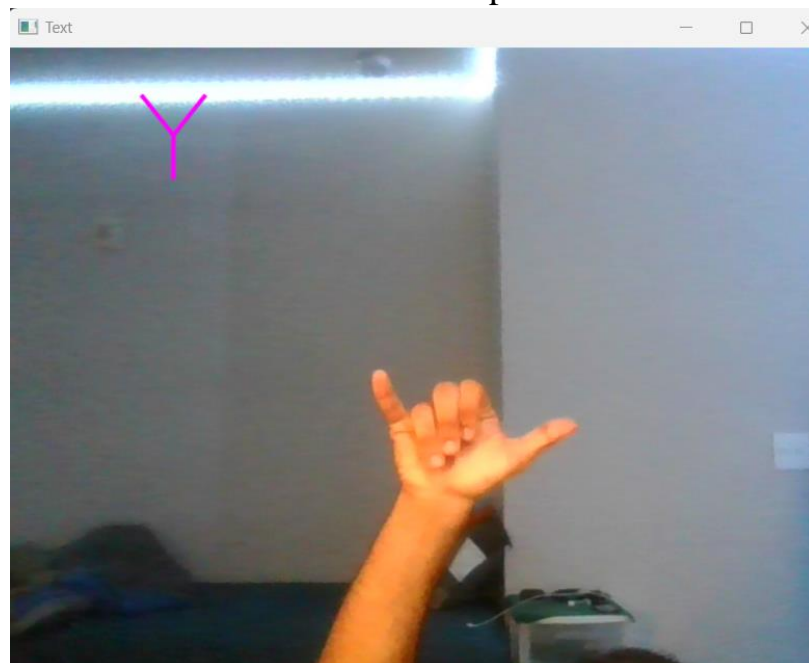
1/1 [=====] - 0s 19ms/step
Predicted Character 1:  F
Confidence 1:  92.07552671432495
Predicted Character 2:  H
Confidence 2:  5.485602468252182
Predicted Character 3:  Y
Confidence 3:  2.3306384682655334

```

Letter F output



Letter W Output






```
1/1 [=====] - 0s 16ms/step
Predicted Character 2: A
Confidence 2: 0.0008044904461712576
Predicted Character 3: L
Confidence 3: 0.0006475837381003657
Predicted Character 1: Y
Confidence 1: 99.99836683273315
```

Letter Y Output

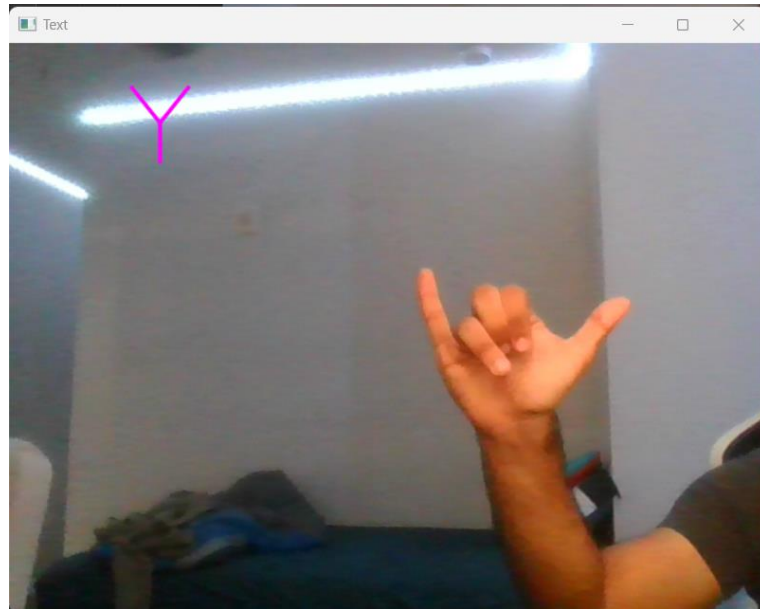
- Results with 3 Convolution Network Layers



The image shows a video frame from a window titled 'Text'. It depicts a person's hand with three fingers raised in a 'V' shape. Above the hand, on a light-colored wall, a pink letter 'W' is drawn. The scene is dimly lit, with a bright light source visible at the top left.

```
1/1 [=====] - 0s 16ms/step
Predicted Character 2: L
Confidence 2: 6.188981607556343
Predicted Character 3: V
Confidence 3: 3.8242310285568237
Predicted Character 1: W
Confidence 1: 86.46872639656067
```

Letter W Output



```
1/1 [=====] - 0s 14ms/step
Predicted Character 2:  A
Confidence 2:  0.54616779088974
Predicted Character 3:  L
Confidence 3:  0.1392587088048458
Predicted Character 1:  Y
Confidence 1:  99.01514649391174
```

Letter Y Output

## Analysis and Discussion

There are several factors which can help us to determine the accuracy of our model. However it is important to select the factor which help us understand the depth of our application. The evaluation metrics should be accurate lucid and viable. The confidence metric allows us to understand how accurately the model is predicting a particular alphabet. Confidence metric returns a percentage of its accuracy on the prediction.

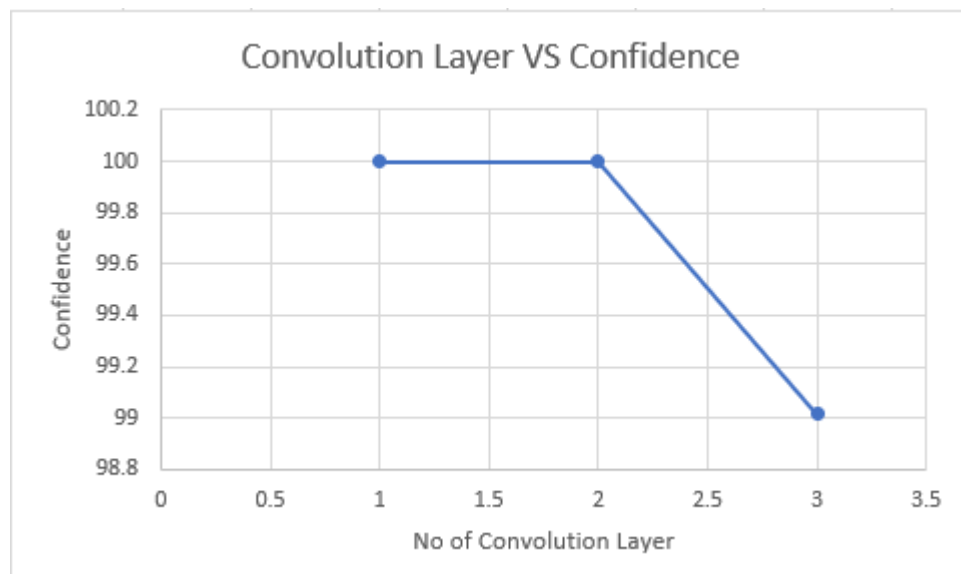
We also tested our model by changing its network structure. We trained our model using 1 convolution layer, 2 convolution layers and 3 convolution layers.



## Observations

- 1) With one convolution layer, we found that all the predictions made by the model were very accurate and gave a good confidence percentage for most of the sign language gesture.
- 2) With two convolution layer, we found that the confidence level were dropped by some percentage although prediction were correct for most of the sign language alphabetical gestures.
- 3) With three convolution layer, our models started to give poor performance, where few letters were getting detected and the confidence rate was also decreasing.

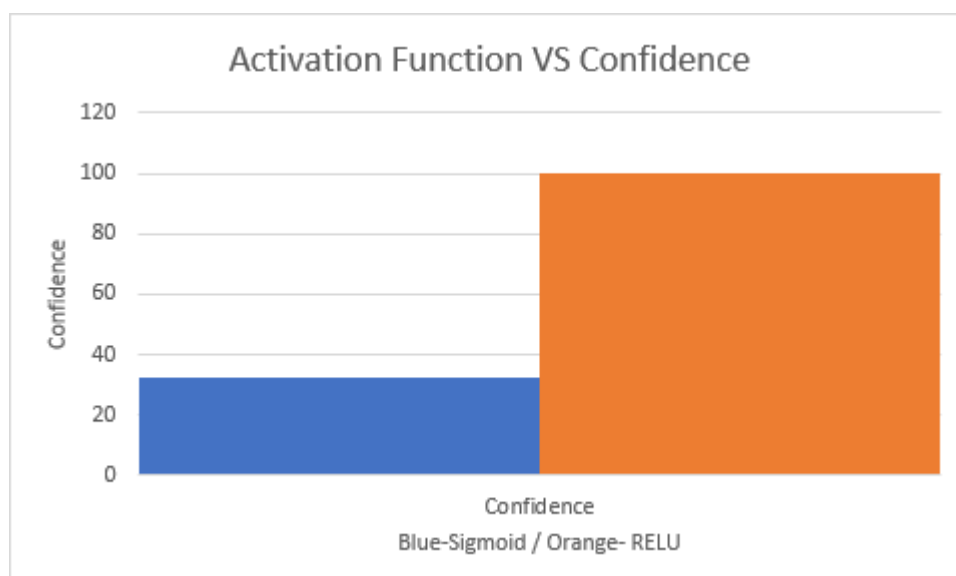
We created a graph plot for confidence vs the number of convolution layers which depicts the changes and the trends we observed.



We also tested out network by changing the activation function from **RELU** activation function to **SIGMOID** activation function and tested it on Letter Y.

### Observations

We observe that the change in activation function drastically reduces the confidence rate and hence the performance of the model. The graph plots accurately determine the change in the observation when activation function is changed.



### Conclusion

Thus, our project includes a convolution neural network which is trained using an image dataset. This model gives a higher confidence rate based on different way the model is constructed and trained. The evaluation metrics such as confidence provide a more concrete base for our model. The project can be used efficiently to remove the barriers between sign language and normal language as it present the output in a more readable format. The implementation of the following system could definitely improve capital spent on the current resources such as time and money. This system could give us a more efficient way to understand sign language and help people all around the world.

## **Contribution**

I created the script for hand detection in real time using the video camera of our systems. I also loaded the neural network model created earlier in this script and further predicted the letter based on the hand gesture that was being captured in the video input. I have also tested the model and provided the required outputs and confidence values for plotting graphs which were further used in the final report.