
Name of the Student	**NITIN SHARMA**
Project Title	Automate Identification and Recognition of Handwritten Text from an Image.
Name of the Company	TCS iON
Name of the Industry Mentor	ANAMIKA CHATTERJEE
Name of the Institute	AMITY UNIVERSITY NOIDA, UTTAR PRADESH.

Start Date	End Date	Total Effort (hrs.)	Project Environment	Tools used
31-June-2020	14-July-2020	210	PyCharm IDE	OpenCV, Numpy, Keras, Scikit-learn, Tensorflow, Matplotlib etc.

Project Synopsis:

The primary objective of this industry project is about developing machine learning algorithms in order to enable entity and knowledge extraction from documents with handwritten annotations. The document can be a scanned document, a photo of a document or a live image. This project aims to develop a computer vision system to extract data from an image.

Widely used as a form of data entry from printed paper data records – whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation – it is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

This project will be working on recognizing handwritten digits from a live image using various python packages and functions.

Solution Approach:

A model was trained using a Dataset of about 10,000 images of handwritten digits to recognize the digit and the probability of it when a handwritten digit is shown through a webcam.

To meet this objective, a Convolution Neural Network or CNN approach is used. The agenda for choosing this field is to enable machines to view the world as humans do, perceive it in a similar manner and even use the knowledge for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc. The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular algorithm is Convolutional Neural Network.

Optical Character Recognition or OCR is a field of research in pattern recognition, artificial intelligence and computer vision. OCR is used to extract digital data from a physical entity. Widely used as a form of data entry from printed data records such as passports, invoices, bank statements, receipts, business cards, Email, printouts of static-data, or any suitable documentation. OCR is used to make a digital footprint of printed text so that they can be edited digitally, searched and stored in a more compact directory. These texts can further be used in machine processes such as cognitive computing, machine learning/translation, text-to-speech applications, text mining etc.

Python packages of OpenCV, Numpy, Tensorflow, Matplotlib etc. are also used to attain the objective of the project. Solutions to various problems during the project were also shared during webinars and in the self-learning modules.

Assumptions:

1. The handwritten text should be a digit 0-9.
2. The text across the input image must be clear with no background on the image.
3. The handwritten text should be black with white background.
4. All machine dependencies must be correctly installed.

Project Diagrams:

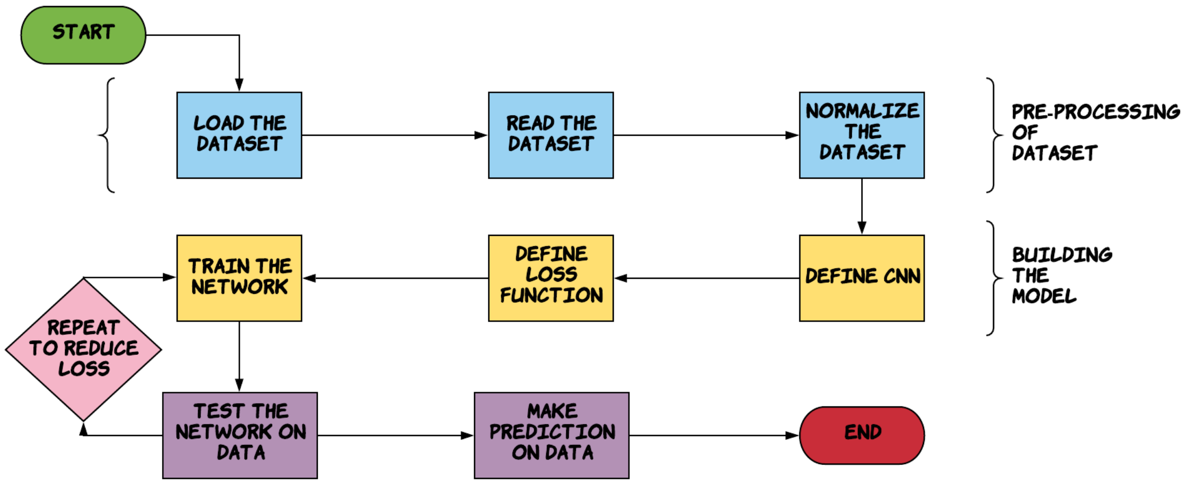


Figure 1. Convolution Neural Network (CNN)

Algorithms:

Optical Character Recognition (OCR) technology recognizes text inside images, such as scanned documents and photos. OCR is used to convert any kind of images containing written text (typed, handwritten or printed) into a digital format.

Text recognition is a two-step task. Firstly, you need to detect the text in the image; second, you identify the characters.

Main approach used in this project is the **Convolutional Neural Network**.

Architecture:

1. **CNN layers**
   a. Convolutional layer with 5x5 filter kernels in the first 2 layers and 3x3 in the last 3 layers.
   b. RELU function.
   c. Pooling Layer.
   d. Output feature map.

2. **Data**
   a. Input: A gray-value image of size 32x32.
   b. Output: Character-probability Matrix.

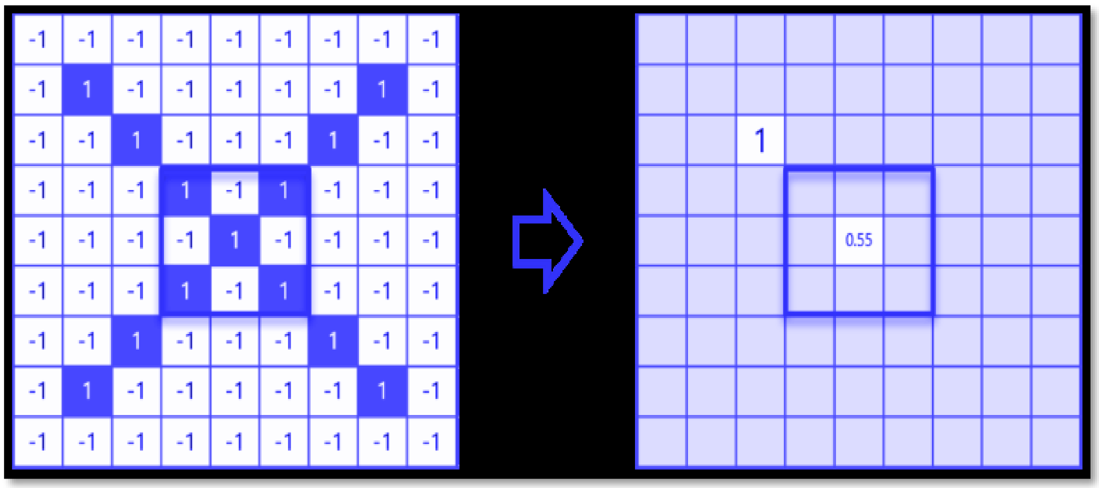
Process:

1. **Create CNN layer and return an output**
   a. For each layer, create a convolution kernel of size k×k.
   b. Feed the convolution result into the RELU operation and to the pooling layer with size px×py and step-size sx×sy.
   c. Repeat the above steps for all layers.

2. **Train the Neural Network**
   a. Use the mean of the loss values of the batch.
   b. Feed the values into an optimizer such as RMSProp.

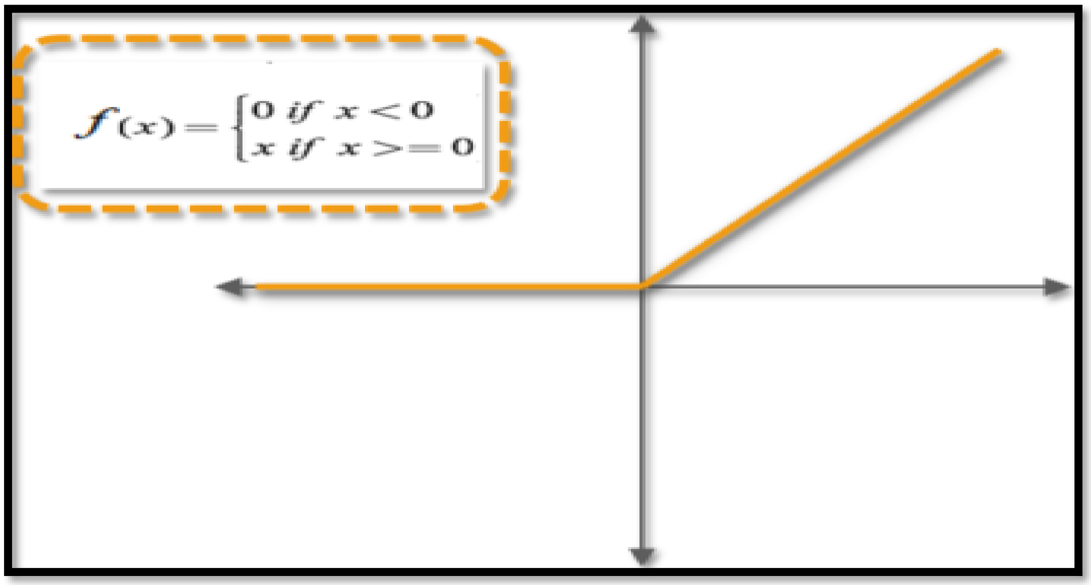
Working:

1. **Convolution:**
   Convolution has the nice property of being translational invariant. Intuitively, this means that each convolution filter represents a feature of interest (i.e. pixels) and the Convolutional Neural Network algorithm learns which features comprise the resulting reference.

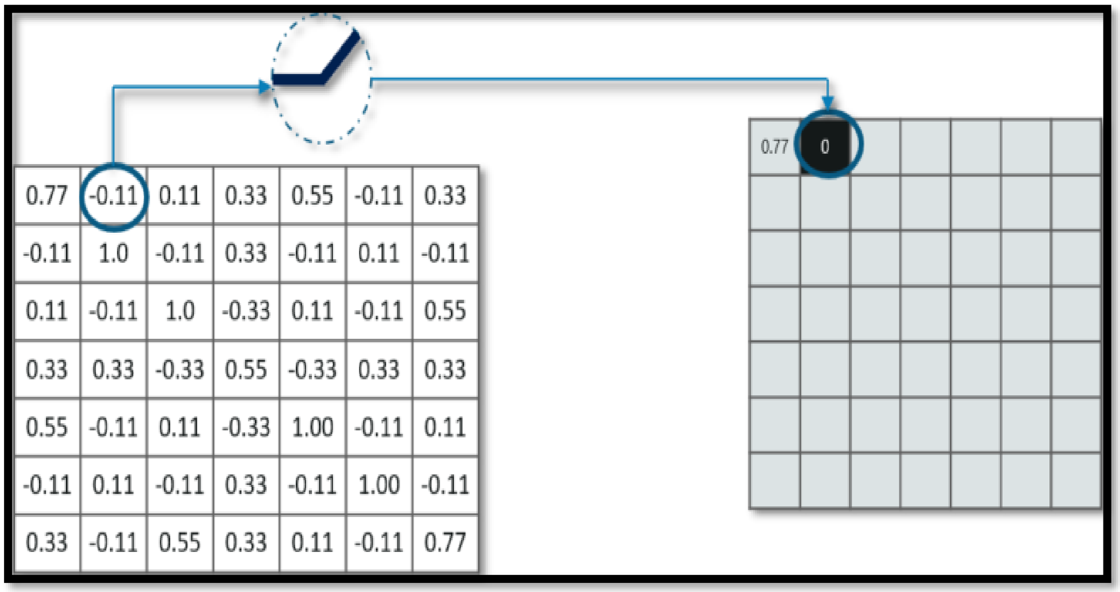


The output signal strength is not dependent on where the features are located, but simply whether the features are present. Hence, an alphabet could be sitting in different positions and the convolutional neural network algorithm would still be able to recognize it.

2. **Rectified Linear Unit (RELU):**
Transform function only activates a node if the input is above a certain quantity, while the input is below zero, the output is zero, but when the input rises above a certain threshold, it has a linear relationship with the dependent variable.

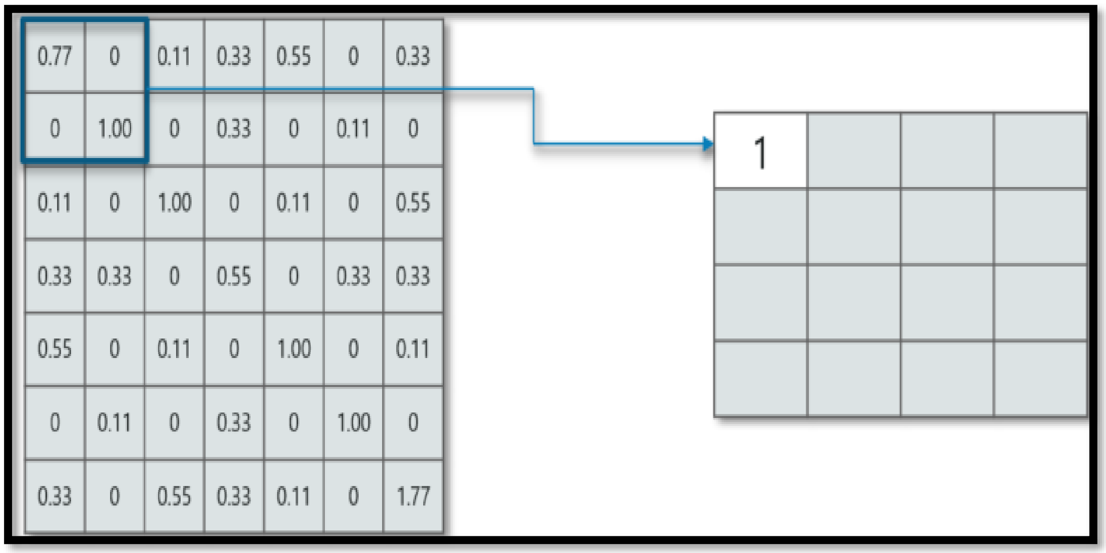


The main aim is to remove all the negative-values from the convolution. All the positive values remain the same but all the negative values get changed to zero as shown below:



0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.0	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.0	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

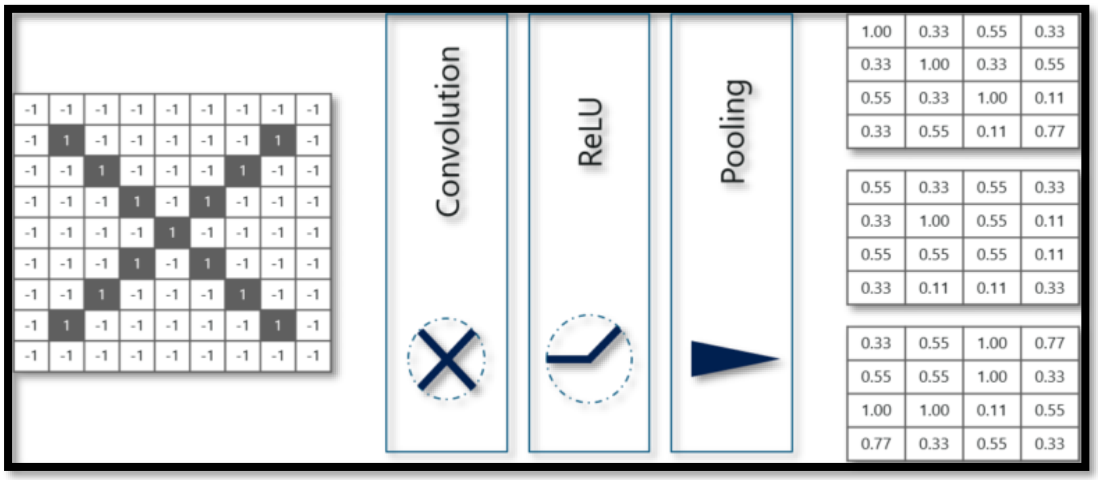
3. **Pooling Layer:**
   In this layer the shrink the image stack into a smaller size. Pooling is done after passing through the activation layer.



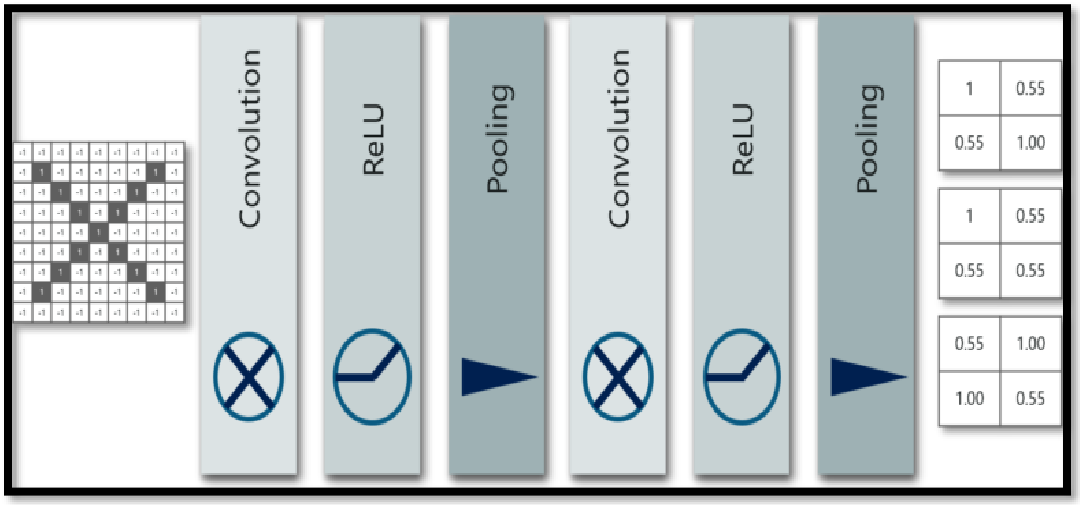
0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	1.77

4. **Stacking Layer:**
   So, to get the time-frame in one picture after passing the input through 3 layers – Convolution, Rectified Linear Unit and Pooling as shown below:

   Say for example we have a 4x4 matrix from a 7x7 matrix after Pooling:



We further reduce the image from 4×4 to 2x2 to achieve this we have to perform the 3 operations in iteration after the first pass. So, after the second pass we arrive at a 2×2 matrix as shown below:



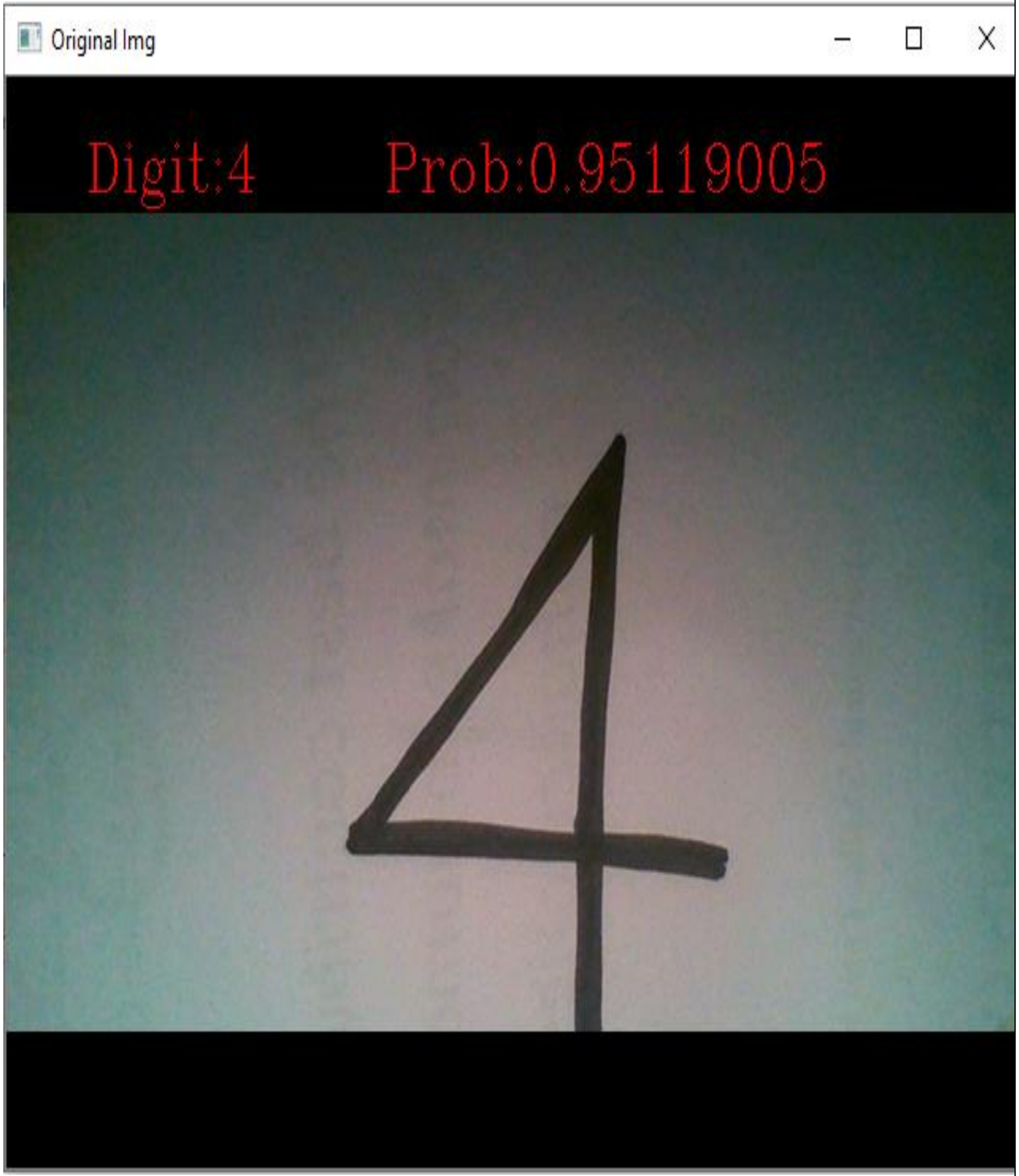
The last layers in the network are fully connected, meaning that neurons of preceding layers are connected to every neuron in subsequent layers.

5. **Prediction:**
At this point in time, we're done training the network and we can begin to predict and check the working of the classifier.

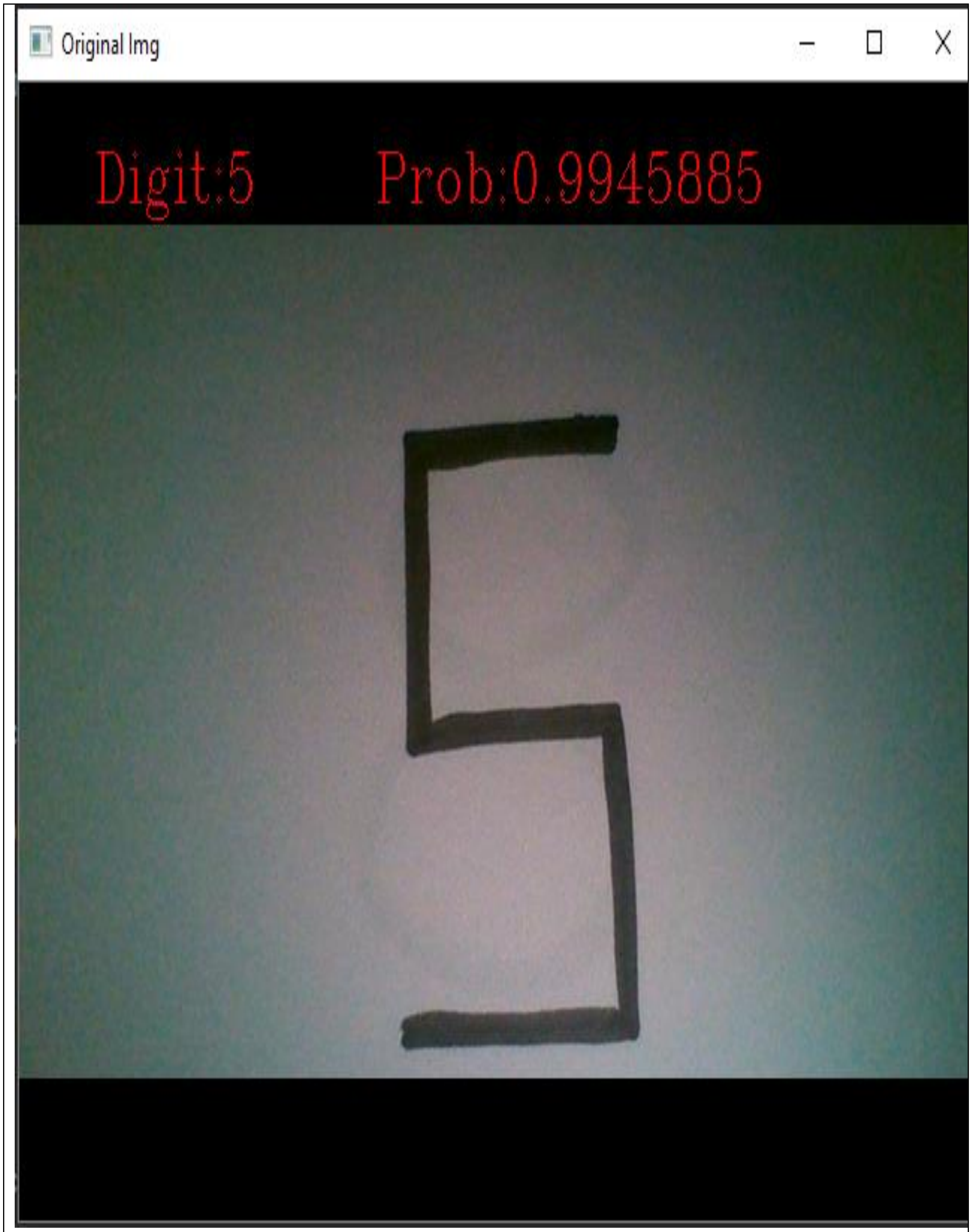
Outcome:

The algorithm is able to detect handwritten text from an image. The model is successfully able to detect digits zero (0) to nine (9) and it is about 98% accurate while implementation and testing.

Testing Handwitten Digits :-

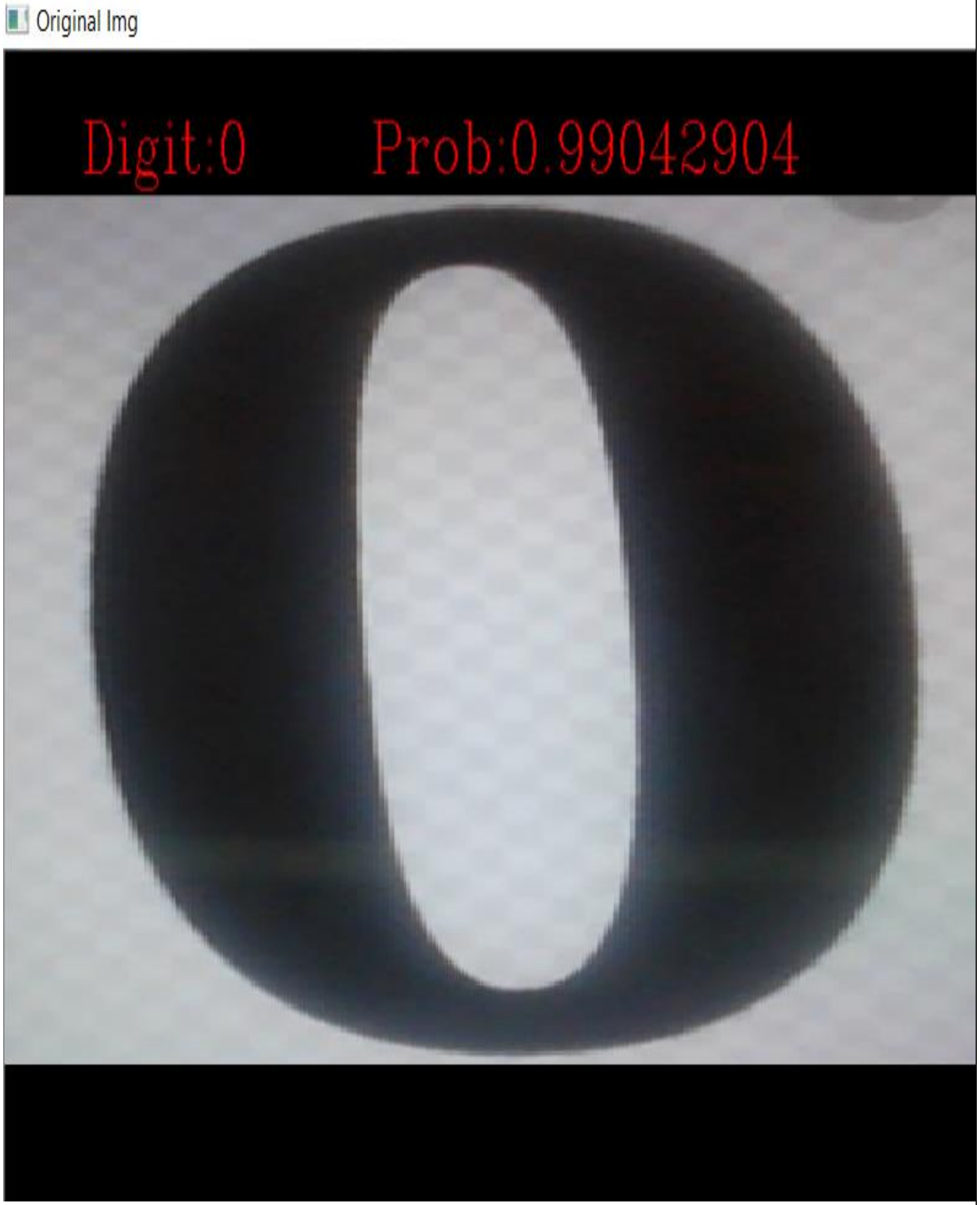


```
Digit:4        Prob:0.95119005
```

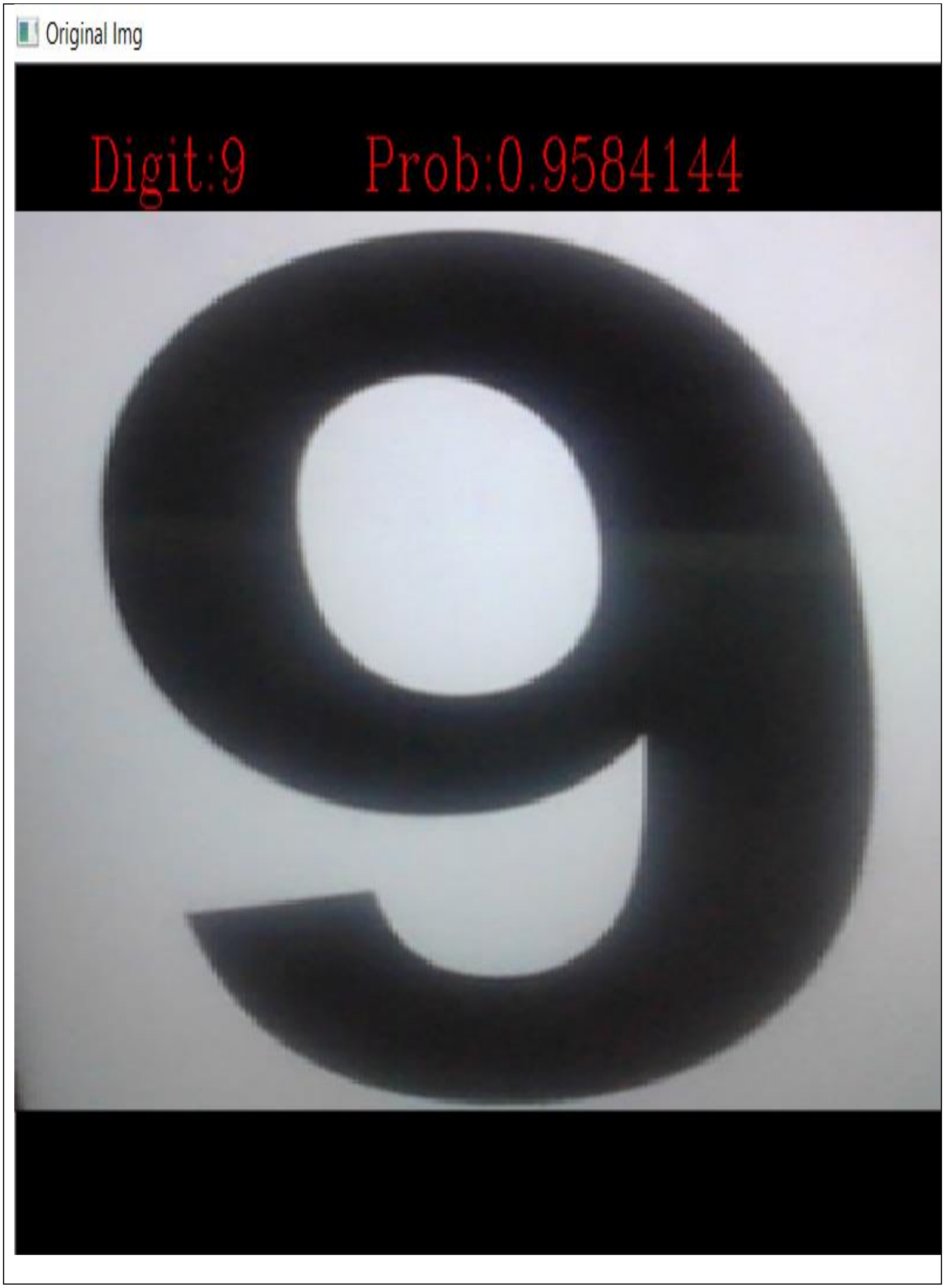


```
Digit:5      Prob:0.9945885
```

Testing Printed Digits :-



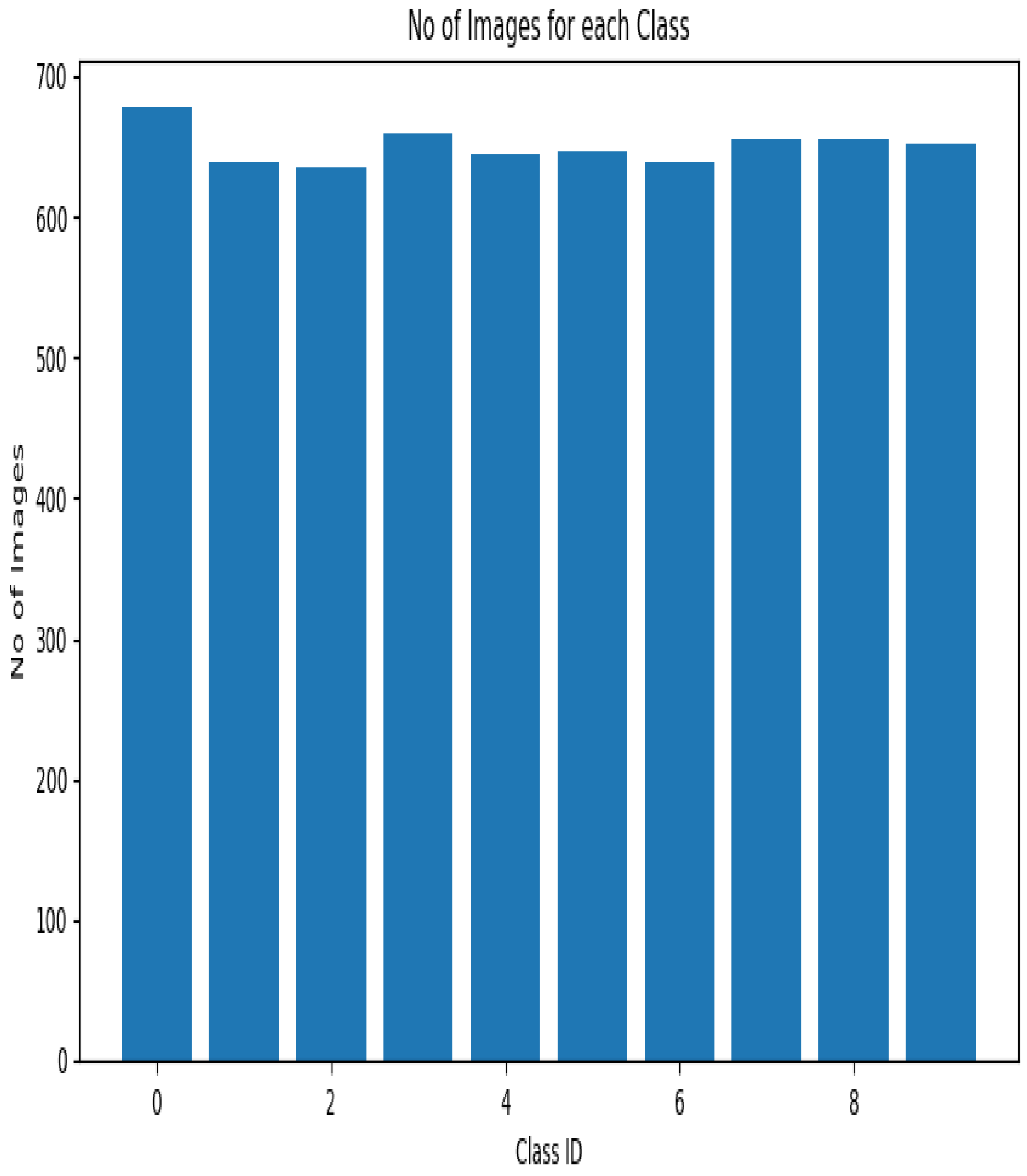
```
Digit:0      Prob:0.99042904
```



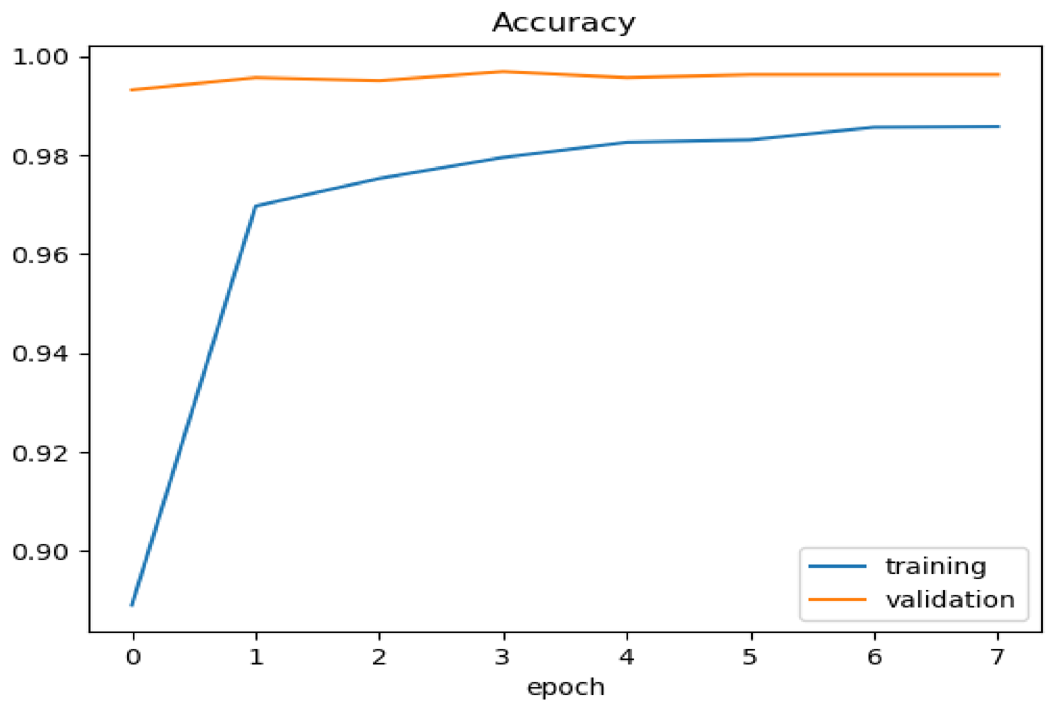
Original Img

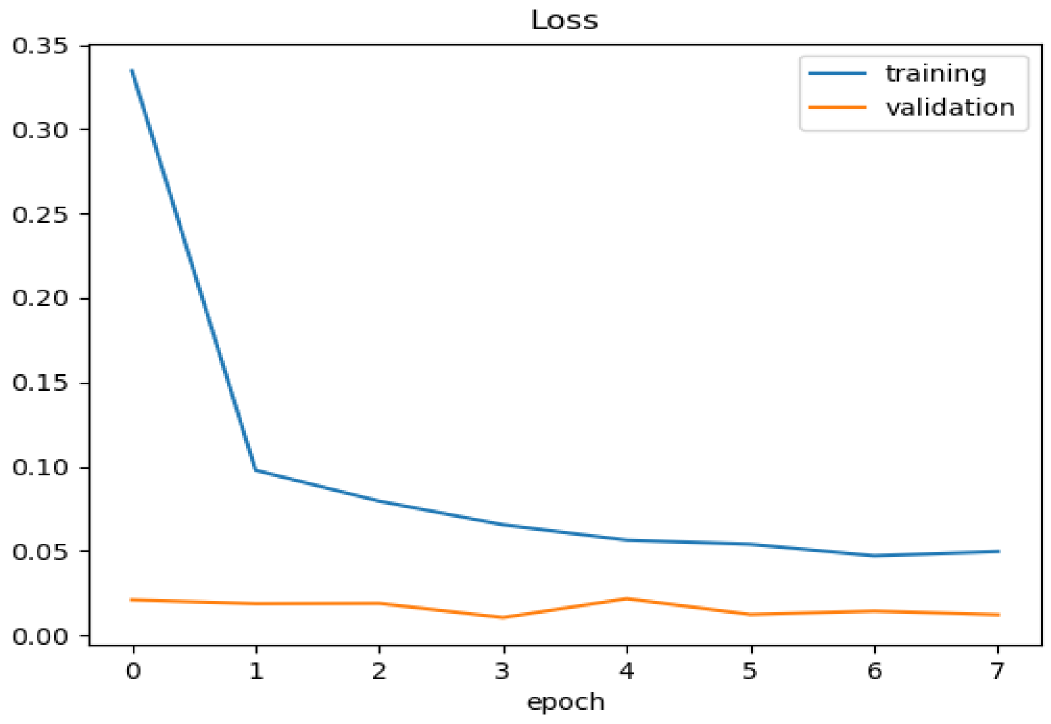
```
Digit:9        Prob:0.9584144
```

As we can see the model is quite accurate and successfully able to extract the handwritten text.



Accuracy and Loss function plots are given below:





Exceptions considered:

1. Text in the image must be Black on white background.
2. The background does not contain aggressive markings on or across the text.

Enhancement Scope:

1. The accuracy of the model can increase with predefined models and powerful machine learning GPU processors can be used to attain a good percentage of accuracy.
2. In future we can use this algorithm with more than one particular language.

References:

https://software.intel.com/content/www/us/en/develop/training/course-artificial-intelligence.html

https://software.intel.com/content/www/us/en/develop/training/course-machine-learning.html

https://www.python-course.eu/machine_learning.php

https://numpy.org/doc/

https://scikit-learn.org/stable/tutorial/index.html

https://keras.io/guides/training_with_built_in_methods/

Link to Code and executable file:

https://github.com/sharmanitin1507/TCS-iON-Remote-Internship-210