Handwritten Digit Recognition

Submitted

By

Sushil Kumar (CSE/15038/93)

Aditya Agarwal (CSE/15142/97)

Project Report submitted to

Indian Institute of Information Technology Kalyani

for the partial fulfilment of the degree of

Bachelor of Technology

in

Computer Science and Engineering

December, 2018

Certificate

This is to certify that the project entitled *"Handwritten digit Recognition"* being submitted by Sushil Kumar and Aditya Agarwal, undergraduate student, Reg No. 000093 and 000097, Roll No. CSE/15038/93 and CSE/15142/97, respectively, in the Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India, for the award of Bachelors of Technology in Computer Science and Engineering is an original research work carried by him under my supervision and guidance. The project has fulfilled all the requirements as per the regulations of Indian Institute of Information Technology Kalyani and in my opinion, has reached the standards needed for submission. The work, techniques and the results presented have not been submitted to any other University or Institute for the award of any other degree or diploma.

(Dr. Tamal Ghose)
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Information Technology Kalyani,
Webel IT Park, West Bengal 741235, India

Declaration

I hereby declare that the work being presented in this project report entitled, "**Handwritten digit Recognition**", submitted to Indian Institute of Information Technology Kalyani in partial fulfilment for the award of the degree of Bachelor of Technology in Computer Science and Engineering during the period from July, 2017 to May, 2018 under the supervision of Dr. Tamal Ghosh, Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India, does not contain any classified information.

Sushil Kumar (CSE/15038/93)

Computer Science and Engineering,
Indian Institute of Information Technology Kalyani,
Webel IT Park, West Bengal 741235, India

Aditya Agarwal (CSE/15142/97)

Computer Science and Engineering,
Indian Institute of Information Technology Kalyani,
Webel IT Park, West Bengal 741235, India

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

.....
(Dr. Tamal Ghose)
Assistant Professor
Departmental of Computer Science and Engineering
Indian Institute of Information Technology Kalyani,
Webel IT Park, West Bengal 741235, India

Acknowledgements

Firstly, we would like to thank our supervisor **Dr. Tamal Ghosh** for his support and guidance to complete this project work without whom we would not be able to make out this far. We would also like to thank our friends who supported us greatly and were always willing to help us. We are very grateful to Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India, for providing us this wonderful opportunity.

Lastly, we would like to thank our Parents a God for their never ending grace.

Sushil Kumar (CSE/15038/93)
Computer Science and Engineering,
Indian Institute of Information Technology Kalyani,
Webel IT Park, West Bengal 741235, India

Aditya Agrawal (CSE/15142/97)
Computer Science and Engineering,
Indian Institute of Information Technology Kalyani,
Webel IT Park, West Bengal 741235, India

Table of Contents

List of Figures

Figures Page

ABSTRACT

The handwritten digit recognition problem becomes one of the most famous problems in machine learning and computer vision applications. Many machine learning techniques have been employed to solve the handwritten digit recognition problem. In this paper ,we focuses on Neural Network (NN) approaches. The most three famous NN approaches are deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN). We tried to implement this problem, using CNN. The data set which we used contains 42,000 training and 28,000 testing cases.

Chapter 1

INTRODUCTION

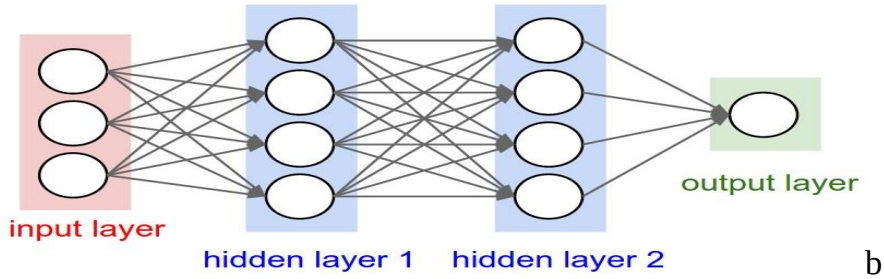
HANDWRITTEN digit recognition is the ability of a computer system to recognize the handwritten inputs like digits, characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades. Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc. A lot of classification techniques using Machine Learning have been developed and used for this like K-Nearest Neighbors, SVM Classifier, Random Forest Classifier etc. but these methods although having the accuracy of 97% are not enough for the real world applications.

One example of this is, if we send a letter with address name as “Anuj” and the system detects and recognizes it as “Tanuj” then it will not be delivered to “Anuj” but “Tanuj”. Although eventually it may come to the right address but if the mail is important, this delay can cost a lot. In short, the accuracy in these applications is very critical but these techniques do not provide the required accuracy due to very little knowledge about the topology of a task.

1.1 BACKGROUND

Deep learning has become the hot tool for Image Processing, object detection, handwritten digit and character recognition etc. A lot of machine learning tools have been developed like scikitlearn, scipy-image etc. and pybrains, Keras, Theano, Tensorflow by Google, TFLearn etc. for Deep Learning. These tools make the applications robust and therefore more accurate.

Also , the Artificial Neural Networks can almost mimic the human brain and are a key ingredient in image processing field. For example, Convolutional Neural Networks with Back Propagation for Image Processing, Deep Mind by Google for creating Art by learning from existing artist styles.



b

Fig 1: 3-layer Neural Network

1.2 PURPOSE

Given a training data on “Handwritten digit recognition” we have to train our model according to it and try to increase the accuracy of the model.

1.3 CONVOLUTIONAL NEURAL NETWORK (CNN):

CNNs works preety well on images. Convolutional Neural Networks [1] are very similar to ordinary Neural Network, they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. Architecture-wise, they are layers consisting of one or more sets “convolution filter” parameters followed by “max pooling”, and finally, layer(s) of FCNs plus output layer.

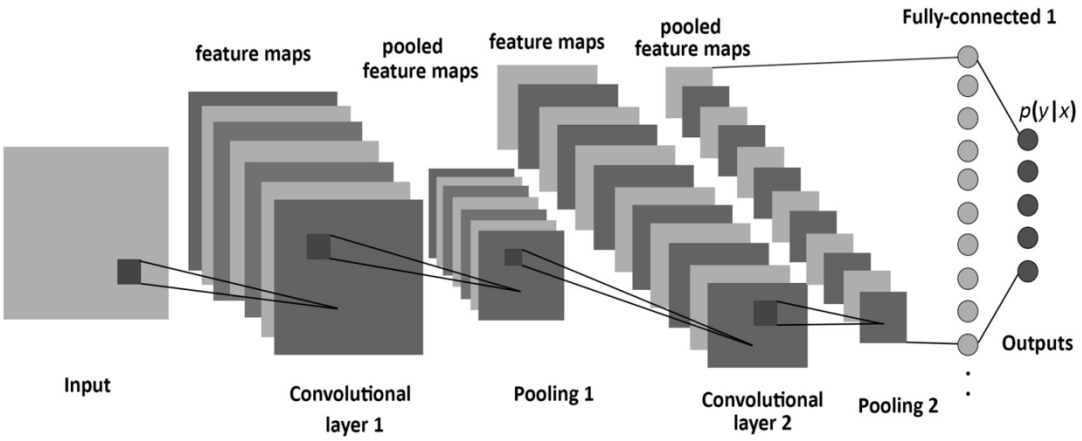


Fig 2: Structure of CNN

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered.

1.4 LAYERS OF CONVOLUTIONAL NEURAL NETWORK:

A CNN consists of a lot of layers. These layers when used repeatedly, lead to a formation of a Deep Neural Network.

The main types of layers used to build a CNN are:

i. **Input:** This layer holds the raw pixel values of image.

ii. **Convolutional Layer:** This layer gets the results of the neuron layer that is connected to the input regions. We define the number of filters to be used in this layer. Each filter may be a 5x5 window that slider over the input data and gets the pixel with the maximum intensity as the output.

iii. **Rectified Linear Unit [ReLU] Layer**: This layer applies an element wise activation function on the image data . We know that a CNN uses back propagation. So in order to retain the same values of the pixels and not being changed by the back propagation, we apply the ReLU function.

iv. **Pooling Layer:** This layer perform a down-sampling operation along the spatial dimensions (width, height), resulting in volume.

v. **Fully Connected Layer:** This layers is used to compute the score classes i.e which class has the maximum score corresponding to the input digits.

Chapter 2

Literature Review

We have applied deep learning to the real-word handwritten Digit recognition, and obtained good performance for image recognition. The dataset which we used is the mnist dataset for handwritten digits.
For the image recognition problem such as handwritten classification, it is very important to make out how data are represented in images. The data here is not the row pixels, but should be the features of images which has high level representation [2, 4]. For the problem of handwritten digit recognition, the digit's structure features should be first extracted from the strokes. Then the extracted features can be used to recognize the handwritten digit. The high performance of large-scale data processing ability is the core technology in the era of big data. Most current classification and regression machine learning methods are shallow learning algorithms [4]. It is difficult to represent complex function effectively, and its generalization ability is limited for complex classification problems[5, 6].
Deep Learning algorithms are highly efficient in image recognition tasks such as MNIST digit recognition[7].

Related Works:
1. A Comparative Study on Handwriting Digit Recognition Using Neural Networks by Mahmoud M. Abu Ghosh and Ashraf Y. Maghari.
In this paper, they compared three Neural Network based recognition algorithms to determine the best algorithm in terms of many factors such as accuracy and performance. Other criteria such as execution time have been also taken in consideration. CNN algorithm and DNN are of almost equal in terms of accuracy.
2. In [8] the author proposed a decision tree learning to classify different writing styles of the identical digits. Several direction features were employed to implement the classification. That is, when the stroke direction of some digits is similar, the decision tree learning can classify them properly. However, it is difficult to manage sets of possibilities as more features.
3. Moreover, the comparison of classification Handwritten Digit Recognition have been published continuously. For example in [9] the authors have performed the experiments by extracting structural features from the handwritten digits by using SVM and tree classifier. The recognition rate of SVM classifier is more than the Tree classifier.

4. "Kaensar et al.[3] have concluded that different classifier affects the recognition rate for handwritten digit recognition. Accordingly, they applied

three classification techniques by using the open source Weka tool kit for training and testing the dataset which was obtained from the UCI repository. The presented results show that SVM is the best classifier to recognize handwritten digits. However, the main problem of the SVM classifier is the time consuming of the training process. Conversely, other methods like neural networks give insignificantly worse results, but their training is much quicker .

Chapter 3

SYSTEM DESCRIPTION

3.1 Data Preparation

We have used the data files provided by Kaggle namely **dig_train.csv** and **dig_test.csv** which contain gray-scale images of hand-drawn digits, from zero through nine.

Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255, inclusive. Each pixel column in the training set has a name like pixelx, where x is an integer between 0 and 783,inclusive.

The training data set, (dig_train.csv), has 785 columns. The first column, called "label", is the digit that was drawn by the user. The rest of the columns contain the pixel-values of the associated image.

3.1.1:Load data

Reading data using panda's "*read_csv* function" as follow-

```
train=pd.read_csv("/content/drive/My Drive/Colab_Notebooks/dig_train.csv")
test=pd.read_csv("/content/drive/My Drive/Colab_Notebooks/dig_test.csv")
```

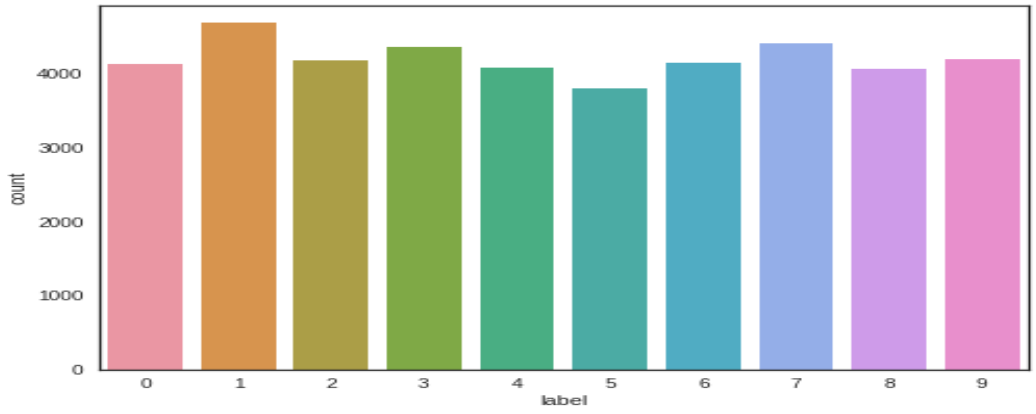


Fig 3: Counting of different digit in training set

3.1.2: Normalization:

As the pixel value ranges from 0 to 255, firstly we conveted it in range [0..1] as CNN converges faster on [0..1] data than on [0..255].

```
# Normalize the data
X_train = X_train / 255.0
test = test / 255.0
```

3.1.3:Reshape

Train and test images (28px x 28px) has been stock into pandas.Dataframe as 1D vectors of 784 values. We reshape all data to 28x28x1 3D matrices.

```
# Reshape image in 3 dimensions (height = 28px, width = 28px , channal
= 1)
X_train = X_train.values.reshape(-1,28,28,1)
test = test.values.reshape(-1,28,28,1)
```

Keras requires an extra dimension in the end which correspond to channels. MNIST images are gray scaled so it use only one channel. For RGB images, there is 3 channels, we would have reshaped 784px vectors to 28x28x3 3D matrices.

3.1.4: Label encoding

Labels are 10 digits numbers from 0 to 9. We need to encode these lables to one hot vectors (ex : 2 -> [0,0,1,0,0,0,0,0,0,0]) .

```
Y_train = to_categorical(Y_train, num_classes = 10)
```

3.1.5: Split training and validation Set:

I perform random split of the train set in two parts : a small fraction (10%) became the validation set which the model is evaluated and the rest (90%) is used to train the model. Validation set basically helps us in the estimation of model skill while tuning model's hyperparameters.

```
# Set the random seed
random_seed = 2
```

```
# Split the train and the validation set for the fitting
X_train, X_val, Y_train, Y_val = train_test_split(X_train, Y_train,
test_size = 0.1, random_state=random_seed)
```

3.1.6: Visualising dataset:

Few handwritten digits present is .

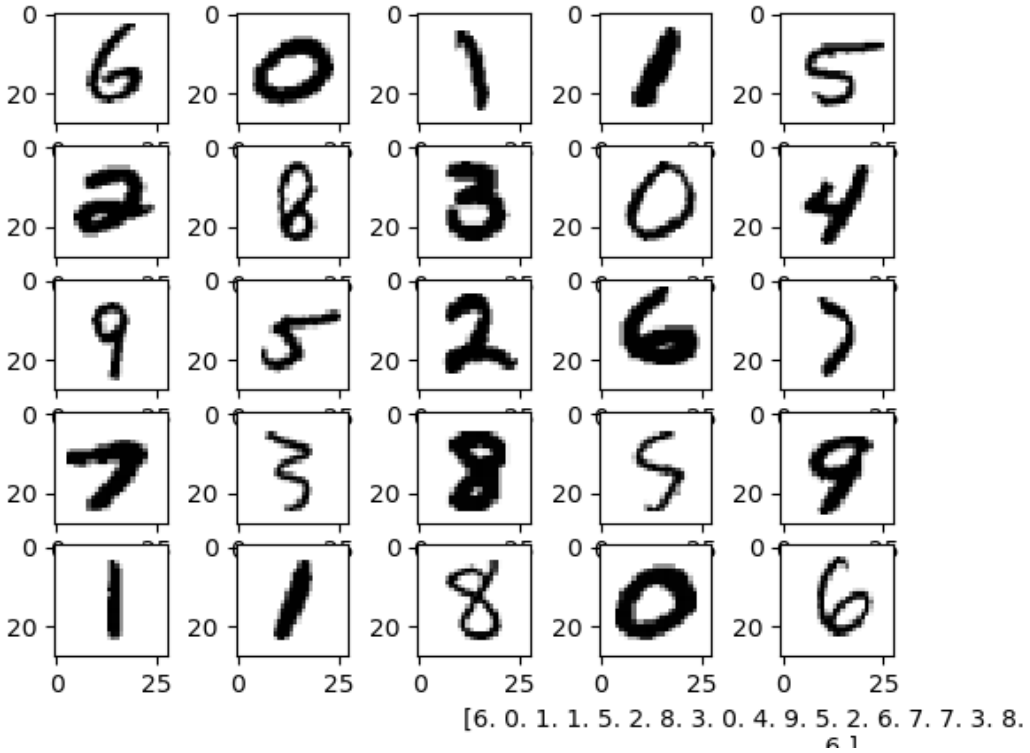


Fig4 :Some of the digits in dataset

3.2:CNN

3.2.1:Defininig model:

We used the Keras Sequential API, where we have just to add one layer at a time, starting from the input. The first is the convolutional (Conv2D) layer. It is like a set of learnable filters. We choose to set 32 filters for the two firsts conv2D layers and 64 filters for the two last ones. Each filter transforms a part of the image (defined by the kernel size) using the kernel filter. The kernel filter matrix is applied on the whole image. Filters can be seen as a transformation of the image.

The CNN can isolate features that are useful everywhere from these transformed images.The second important layer in CNN is the pooling (MaxPool2D) layer. This layer simply acts as a down sampling filter. It looks at the 2 neighbouring pixels and picks the maximal value. These are used to reduce computational cost, and to some extent also reduce overfitting. Combining convolutional and pooling layers, CNN are able to combine local features and learn more global features of the image.
We used "ReLU" as our Activation function for hidden layer and "Softmax " for the output layer. The rectifier activation function is used to add non linearity to the network. We then use Flatten layer to convert the final feature maps into a one single 1D vector, now this is send to the Fully Connected Layer i.e. Output Layer.

3.2.2:Setting Parameters

Once our layers are added to the model, we need to set up a score function, a loss function and an optimisation algorithm.
We define the loss function to measure how poorly our model performs on images with known labels. It is the error rate between the oberved labels and the predicted ones.
The most important function is the optimizer. This function will iteratively improve parameters (filters kernel values, weights and bias of neurons) in order to minimise the loss.

3.3: Evaluate the Model:

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. We can take help of Training and Validation curve and confusion matrix for this purpose.

3.3.1:Training and Validation Curve.

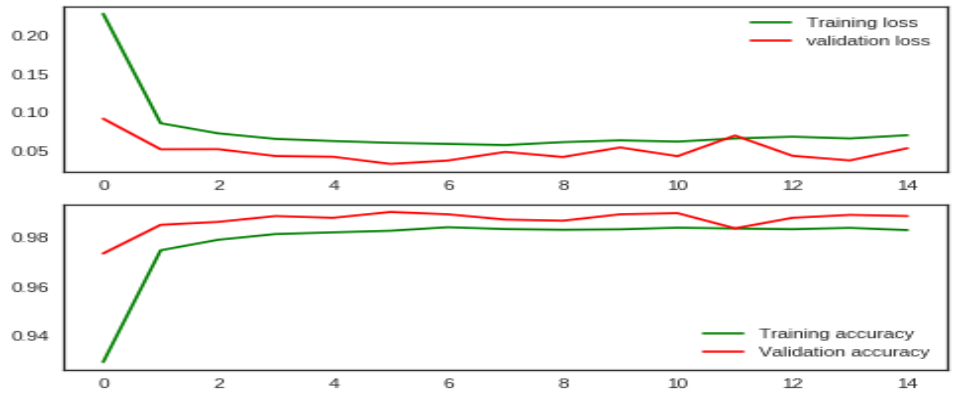


Fig 5: Training and validation Curve

Chapter 4

Algorithm used

We have used CNN, which is basically a neural network for implementing our model. CNN are very much suited for images. Detection using CNN is rugged to distortions such as change in shape due to camera lens, different lighting conditions, different poses, presence of partial occlusions, horizontal and vertical shifts, etc. However, CNNs are shift invariant since the same weight configuration is used across space. In theory, we also can achieve shift invariantness using fully connected layers. But the outcome of training in this case is multiple units with identical weight patterns at different locations of the input. To learn these weight configurations, a large number of training instances would be required to cover the space of possible variations.

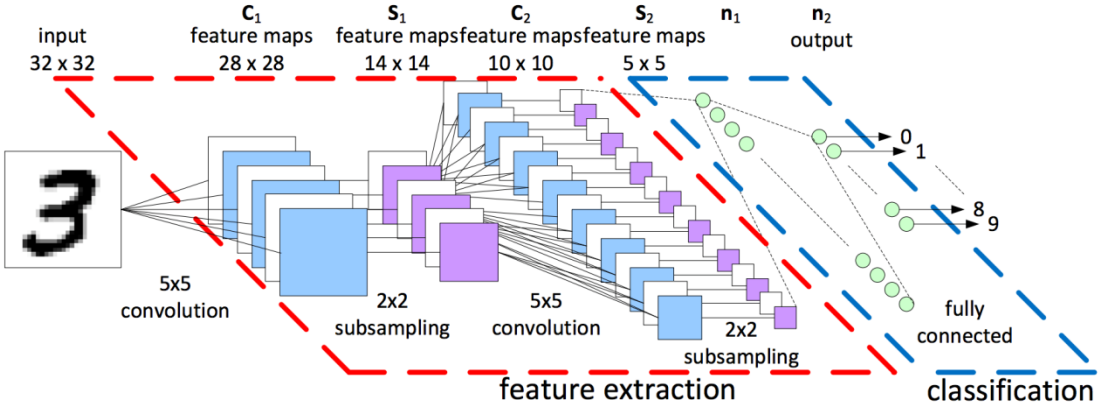


Fig 6: Working of CNN

A CNN consists of one or more convolutional layers, often with a subsampling layer, which are followed by one or more fully connected layers as in a standard neural network.

Chapter 5

Results

We are able to get an accuracy of ~98% by our model. Some of the error which our model predict wrongly are-

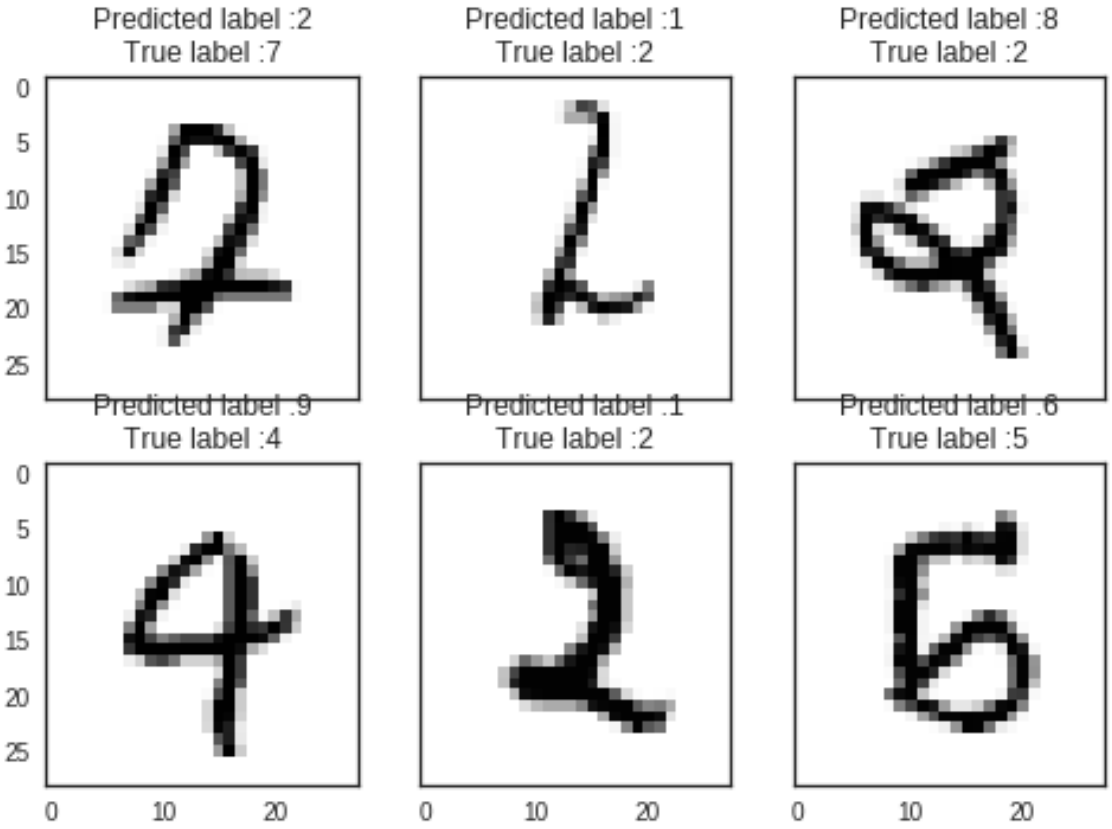


Fig 7:Some of the errors by the model

Some of these errors can also be made by humans, especially for one the 7 that is very close to a 2. The last 5 is also very misleading, it seems for me that is a 6.

Chapter 6

Conclusion and Future Work

We have successfully achieved the accuracy of ~98%, which is much closer to the current achieved accuracy of 99.2%.

We have done the Feature Engineering behind the CNN , and tried to achieve the maximum accuracy , because if we are dealing with Industrial Standards then 1% mistake is a huge. Since , Digit Recognition is used in approximately every Courier Services, so its a demanding Research paper.

We are planning to extend our semester project with the “HAND-WRITTERN ALPHABET RECOGNISTION” , which is also a demanding model for scalability and Industrial use.

Bibliography

[1] http://cs231n.github.io/convolutional-networks/

[2] Walid , R. and A. Lasfar. Handwritten digit recognition using sparse deep architectures. in Intelligent Systems: Theories and Applications (SITA-14), 2014 9th International Conference on. 2014. IEEE.

[3] Kaensar, C. (2013). A Comparative Study on Handwriting Digit Recognition Classifier Using Neural Network, Support Vector Machine and K-Nearest Neighbor. Advances in Intelligent Systems and Computing, 155–163. doi:10.1007/978-3-642-37371-8_19

[4] Schmidhuber, J., Deep learning in neural networks: An overview. Neural Networks , 2015. 61: p. 85-117.

[5]LeCun , Y., Y. Bengio , and G. Hinton, Deep learning. Nature, 2015. 521(7553): p. 436-444.

[6]Hinton, G.E. and R.R. Salakhutdinov , Reducing the dimensionality of data with neural networks. Science, 2006. 313(5786): p. 504-507.

[7]LeCun, Y., C. Cortes, and C.J. Burges, The MNIST database of handwritten digits. 1998

[8] Jiang, W.L., Sun, Z.X., et al.: User-Independent Online Handwritten Digit Recognition. In: Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, pp. 3359–3364. IEEE Press (2006)

[9] Garg, N.K., Jindal, S.: An Efficient Feature Set For Handwritten Digit Recognition. In: The 15th International Conference on Advanced Computing and Communications, pp. 540–544. IEEE Press (2007).