HANDWRITTEN DIGIT RECOGNITION

Student ID: 19IT130, 19IT151

Abstract:

Digit Recognition is a noteworthy and important issue. Manually written digits have invariant position, shape, style etc. so it becomes tedious as well as time consuming process for computer. The uniqueness and assortment of each digit has still been a major issue in recognizing to minimize certain efforts ML model is formed which recognize the digits through several neural network techniques.

It has a great significance on recognizing zip codes, processing bank cheque amounts, tax forms etc.

Aim:

Recognize and classify the digits of MNIST dataset by different classifiers and choose the appropriate model

Literature Survey:

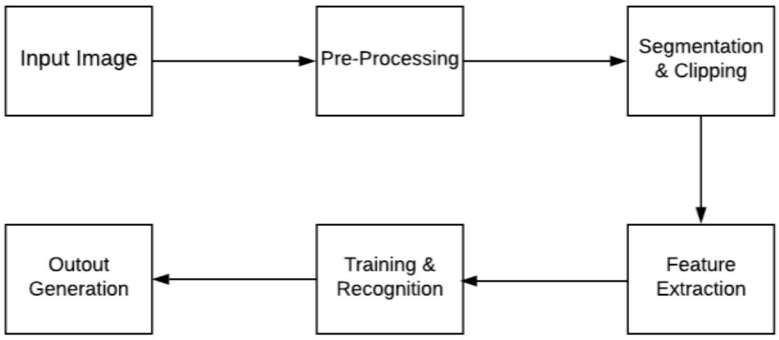
Anuj Dutt demonstrated in his article that by using the deep learning system he has the ability to achieve extremely high precision measurement. Using the Comfort Neural Network as a backend with Carcass and Theano gave them an accuracy of 98.72%. Additionally, CNN's performance using Tensorflow gives a surprisingly better result of 99.70%. Despite the fact that the complexity of the process and the code seems to be greater compared to specific machine learning algorithms, the precision they achieve is increasingly evident.

In an article published by Saeed al-Mansuri, the Multilayer Perceptron Neural Network (MLP) was applied to identify and predict handwritten digits from 0 to 9. The proposed neural system was trained and tested on a derived data set by MNIST

Architecture:

SGP SEMESTER 4

List of Modules in your Project:



1. Pre-processing

   Normalize the image from RGB to gray scale to get more accurate resultsAs well as there is no effect of colour change on the image

2. Segmentation

   Reshapes the images by adding one extra dimension for the kernel.
   It has segments it into sub images of individual digits which are assigned to each digit

3. Feature Extraction

   The pre-processed images are represented in matrix form. The image undergoes convolution for feature extraction. To decrease time efficiency ,data is passed maxpolling layer where its size is reduced and then it is flattened to 1-D by flatten layer

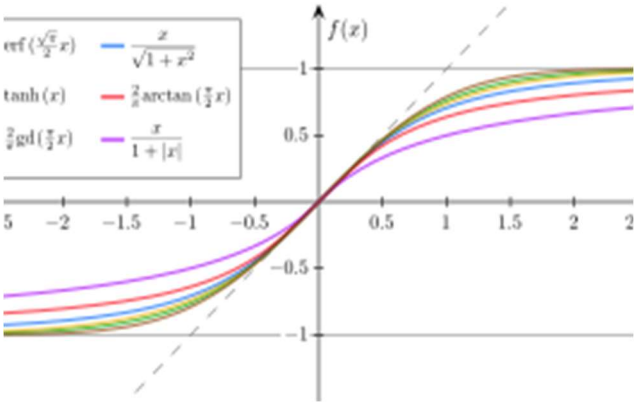
4. Classification and Recognition

It is passed through classifiers like KNN, CNN, Logistic Regression, Random Forest, Decision Tree etc.

Classifiers:

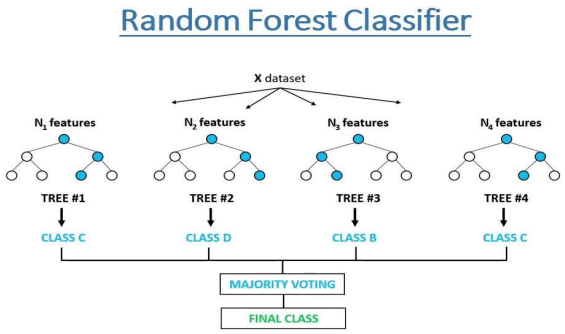
LOGISITC REGRESSION

- Used when dependent variables are generally binary in nature
- It uses logistic (sigmoid) function to predict



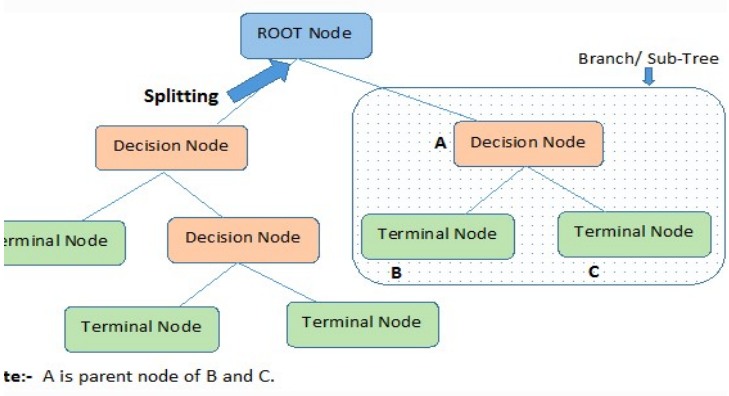
RANDOM FOREST CLASSIFIER

- This classifier can be used for regression as well as classification.
- Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.
- It has nearly the same hyperparameters as a decision tree or a bagging classifier.
- Random forest adds additional randomness to the model, while growing the trees



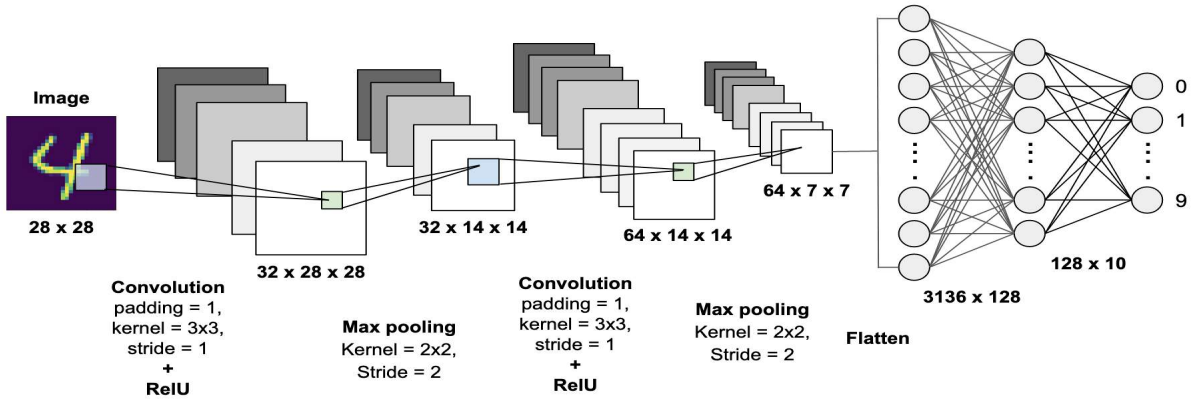
DECISION TREE CLASSIFIER

- Creates a training model in which data is split continuously according to certain parameter
- It can be explained by two entities decision nodes and leaves



CNN

- Convolutional Neural Network is a special architecture of artificial neural network .
- Instead of image computer sees an array of pixel.
- It automatically detects the features and give more accuracy compared to others
- Does not get affected by the location of an object in an image

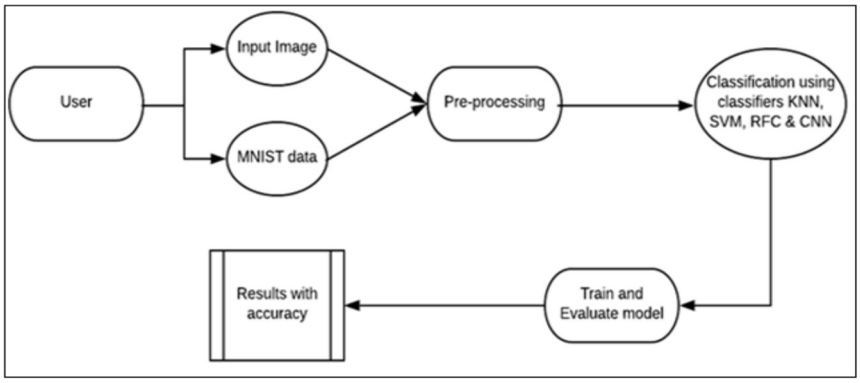


Methodology:

We have loaded MNIST dataset, which is present in the Keras library, it is known for digit recognition. The dataset is assigned to train and test. Here train dataset contains 60000 images whereas test dataset has 10000 images.

Images are 2D matrix where each pixel is represented between [0,255] it is range for RGB colors here 0 means black and 255 means white

To get higher accuracy, it normalizes in the scale of [0,1].Then one dimension is added for the kernel. Afterwards model is prepared using sequence of layers. To get more accurate results we have added 2 convolution and 2 maxpolling layer and to decrease time usage we have also flatten the dataset.
At the end it is passed to 10 neurons through dense layer here 10 neurons are 0 to 9 classes. When the user draws the digit on the canvas it passed through the model and displays which digits it is as well as the percentage of surety to be that digit.



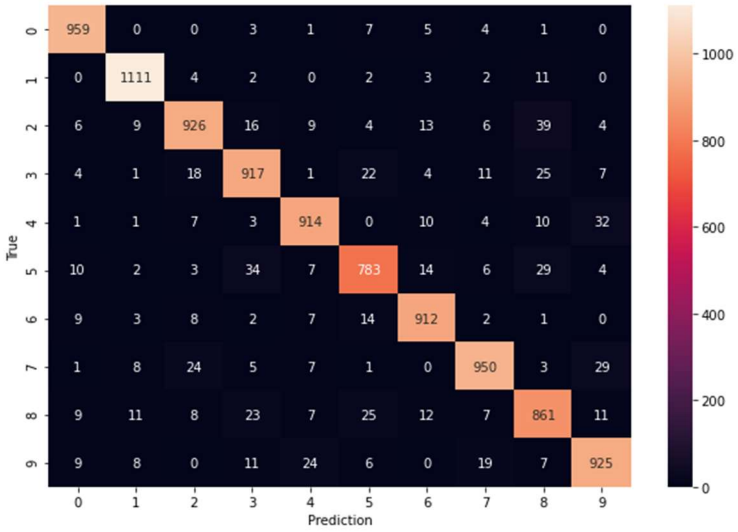
The accuracy and confusion matrix of the different classification model is given below:

1. Logistic Regression

   Accuracy Score:

```
model.score(x_test,y_test)
0.9258
```

   Confusion Matrix:

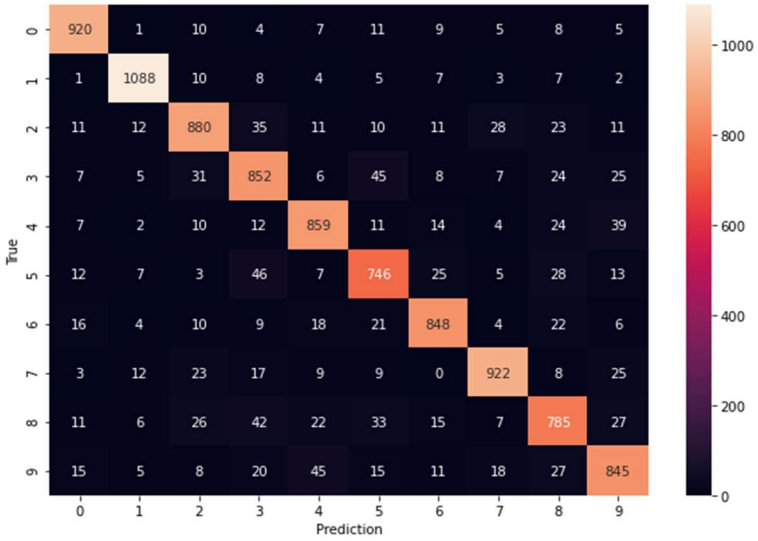


2. Decision Tree Classifier

Accuracy:

```
model.score(x_test,y_test)
0.8745
```

Confusion Matrix:

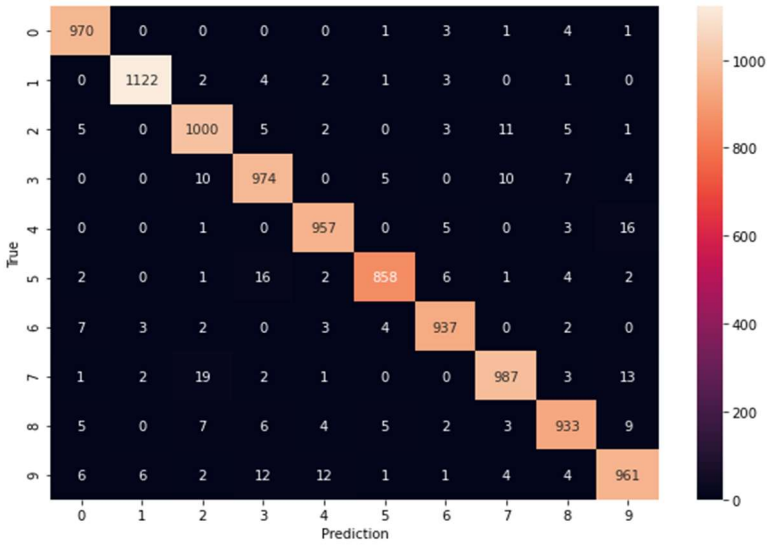


3. Random Forest

Accuracy:

```
model.score(x_test,y_test)
0.9699
```

Confusion Matrix:

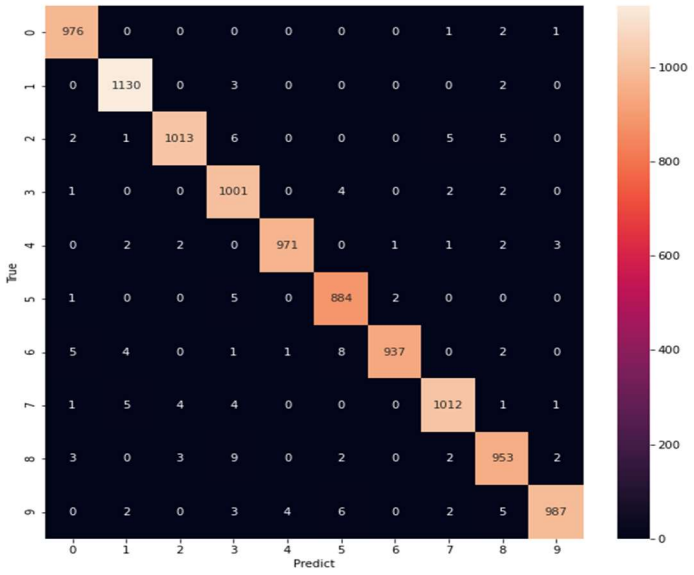


4. CNN

Accuracy:

```
2021-04-15 14:11:43.299091: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized
with oneAPI Deep Neural Network Library (oneDNN)to use the following CPU instructions in performance-critical
operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Epoch 1/5
1875/1875 [==============================] - 71s 38ms/step - loss: 0.1590 - accuracy: 0.9507
Epoch 2/5
1875/1875 [==============================] - 67s 36ms/step - loss: 0.0574 - accuracy: 0.9822
Epoch 3/5
1875/1875 [==============================] - 67s 36ms/step - loss: 0.0414 - accuracy: 0.9869
Epoch 4/5
1875/1875 [==============================] - 67s 36ms/step - loss: 0.0328 - accuracy: 0.9898
Epoch 5/5
1875/1875 [==============================] - 75s 40ms/step - loss: 0.0295 - accuracy: 0.9907
313/313 [==============================] - 3s 10ms/step - loss: 0.0481 - accuracy: 0.9864
0.04805300757288933
0.9864000082015991
```

Confusion Matrix:



Supporting tools/ technology/ framework:

TENSORFLOW: Run train and deep neural network for handwritten digit classification, image recognition, word embedding, repetitive neural network, sequence-to-sequence models for machine translation, natural language processing, and simulations based on PDEs (partial differential equations).

KERAS: Powerful and easy to use open source Python library for the development and evaluation of deep nada learning models. It encompasses efficient theno and tenserflow numerical computation libraries and allows you to define and train neural network models in a few lines of code.

RESULTS


```
In [6]: runfile('C:/Users/sjhan/OneDrive/Desktop/SGP Sem-4/gui.py', wdir='C:/Users/sjhan/
OneDrive/Desktop/SGP Sem-4')
Reloaded modules: tensorflowTesting
argmax 0
 1.0
 0
```

Conclusion:

CNN has more accuracy compared to other classification model as well as it is invariant of position which makes it more effective compared to other neural networks

References:

[1] https://youtu.be/_u-PaJCpwiU

[2] https://youtu.be/Mubj_fqiAv8

[3] https://data-flair.training/blogs/python-deep-learning-project-handwritten-digit-recognition/

[4] https://www.slideshare.net/RishabhTyagi48/handwritten-digit-recognitionusing-cnn-project-ppt-by

[5] M. Wu and Z. Zhang, *Handwritten Digit Classification using the MNIST Dataset*, 2010.

Dutta and A. Dutta, *Handwritten digit recognition using deep learning*, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 6, no. 7, July 2017.
[6] Al Meaded, Somaiya, and Abdelavi Hossain, *Automatic prediction of age, gender, and nationality in offline handwriting*. EURASIP Journal on Image and Video Processing, no. 1 2014.