

**“Digit Recognition”**

**PROJECT REPORT**

**Submitted for the course-‘Neural Network and Fuzzy Control’**

**COURSE CODE- “EEE1007”**

**SLOT-“G1”**

**BY**

|  |  |
| --- | --- |
| **Naman Agrawal**  **Utsav Dasgupta**  **Varun Singh Inda**  **Satyam Jindal** | **16BEE0095**  **16BEE0222**  **16BEE0023**  **16BCE0357** |

**NAME OF FACULTY- Associate Professor Vivekanandan S**

**(SCHOOL OF ELECTRICAL ENGINEERING)**

**CERTIFICATE**

This is to certify, that the project entitled “Digit Recognition” submitted by the following students, B. Tech, VIT University, Vellore for the partial fulfilment of the B. Tech- Neural Network and Fuzzy Control- CAL course, is a record of bonafide carried out by the students under my supervision, as per VIT regulation of academic.

The project carried by the following students have not submitted in anywhere in the University.

**Name of the students**

Naman Agrawal 16BEE0095

Utsav Dasgupta 16BEE0222

Varun Singh Inda 16BEE0023

Satyam Jindal 16BCE0357

**Place: Vellore Signature of FACULTY**

**Date: 31st October 2017 Prof. Vivekanandan S**

**ACKNOWLEDGEMENT**

With all our gratitude we would like to thank VIT UNIVERSITY for giving us the opportunity to work under this project as a team. We would like to express our sincere thanks to **Prof. Vivekanandan S** for guiding us and in completing this project successfully, your guidance helped us a lot.

At last we would thank the university staff for letting us access the resources and giving their expertise suggestions.

**ABSTRACT**

Handwriting number recognition is a challenging problem researchers had been research into this area for so long especially in the recent years. In our study there are many fields concern with numbers, for example, checks in banks or recognizing numbers in car plates, the subject of digit recognition appears. A system for recognizing isolated digits may be as an approach for dealing with such application. In other words, to let the computer understand the numbers that is written manually by users and views them according to the computer process. Scientists and engineers with interests in image processing and pattern recognition have developed various approaches to deal with handwriting number recognition problems such as, minimum distance, decision tree and statistics. The main objective for our system was to recognize isolated digits exist in different applications. For example, different users had their own handwriting styles where here the main challenge falls to let computer system understand these different handwriting styles and recognize them as standard writing. We presented a system for dealing with such problem. The system started by acquiring an image containing digits, this image was digitized using some optical devices and after applying some enhancements and modifications to the digits within the image it can be recognized using feed forward back propagation algorithm.The digits were then taken through to a dataset known as MNIST which contributes around 70,000 patterns of all the digits.At last the digits are processed and being displayed on the output unit. An overall accuracy meet using this system was 92% on the test data set used. We developed a system for handwritten recognition. And we efficiently choose a segmentation method to fit our demands. Our system successfully designs and implement a neural network which efficiently go without demands, after that the system are able to understand the numbers that was written manually by users.

**INTRODUCTION**

Recently, a lot of works was done by depending on the computer; In order to let the processing time to be reduced and to provide more results that are accurate, for example, depending on different types of data, such as characters and digits and the numbers are used frequently in normal life operation. In order to automate systems that deal with numbers such as postal code, banking account numbers and numbers on car plates. And an automatic recognition number system is proposed in this study. Digit recognition has been extremely found and studied. Various approaches in image processing and pattern recognition have been developed by scientists and engineers to solve this problem. That is because it has an importance in several fields and it may probably be used in checks in banks or for recognizing numbers in cars plates, or many other application. In this study, system for recognized of digits is built, which may benefit various fields, the system concerning on isolated digits, the input is considered to be an image of specific size and format, the image is processed and then recognized to result of an edited digits. The proposed system recognizes isolated digits as the system acquire an image consisting digits, then, the image will be processed into several phases such as image enhancement, thinning, skeletonaization and segmentation before recognizing the digit. A multilayer neural network will be used for the recognition phase; a feed forward back propagation algorithm will be applied for training the network and finally change them into numeral text .

**Methodology Involved**

This project was carried out in python using various dependency which involves Open CV2, Sk-Learn, Ski-Image, Numpy, Collections. The algorithm used in this project is multilayer perceptron. The dataset is taken from MNIST (Modified National Institute of Standards and Technology database).

There are four steps to build the isolated digits recognition system :

**Image acquisition :** We will acquire an image to our system as an input .this image should have a specific format, for example, bmp format and with a determined size such as 30×20 pixels. This image can be acquired through the scanner or, digital camera or other digital input devices.

**Preprocessing :** After acquiring the image, it will be processed through sequence of preprocessing steps to be ready for the next step.

**Noise removal :** Reducing noise in an image. For on-line there is no noise to eliminate so no need for the noise removal. In off-line mode, the noise may come from the writing style or from the optical device captures the image.

**Normalization-scaling** : standardize the font size within the image. This problem appears clearly in handwritten text, because the font size is not restricted when using handwriting.

**Thinning and skeletonization** : Representing the shape of the object in a relatively smaller number of pixels. Thinning algorithms can be parallel or sequential. Parallel is applied on all pixels simultaneously. Sequential examine pixels and transform them depending on the preceding processed results.

**Segmentation** : Since the data are isolated, no need for segmentation. With regards to the isolated digits, to apply vertical segmentation on the image containing more than digit will isolate each digit alone.

**Normalization scaling and translation** : Handwriting produces variability in size of written digits. This leads to the need of scaling the digits size within the image to a standard size, as this may lead to better recognition accuracy. We tried to normalize the size of digit within the image and also translate it to a specific position by the following.

**Feature extraction :** Feature extraction is not part of this project. Feature types are categorized as follows:

• Structural features: It describes geometrical and topological characteristics of a pattern by representing its global and local properties

• Statistical features: Statistical features are derived from the statistical distribution of pixels and describe the characteristic measurements of the pattern

• Global transformation: Global transformation technique transforms the pixel representation to a more compact form. This reduces the dimensionality of the feature vector and provides feature invariants to global deformation like translation, dilation and rotation

**Classification and recognition** : Neural Network is a network of non-linear system that may be characterized according to a particular network topology. Where, this topology is determined by the characteristics of the neurons and the learning methodology. The most popular architecture Of Neural Networks used in Arabic digits recognition takes a network with three layers. These are: Input layer, hidden layer and output layer. The number of nodes in the input layer differs according to the feature vector’s dimensionality of the segment image size.

**Code Used in the Project**

The Code is divided into two parts: -

Training a Classifier – The HOG features for each sample are calculated and a multi-class linear SVM (Support Machine Vector) is trained along with the label for each feature.

Testing the classifier – After the training is done the classifier is tested.

**CODE – (Training Classifier)**

# Import the modules

from sklearn.externals import joblib

from sklearn import datasets

from skimage.feature import hog

from sklearn.svm import LinearSVC

from sklearn import preprocessing

import numpy as np

from collections import Counter

# Load the dataset

dataset = datasets.fetch\_mldata("MNIST Original")

# Extract the features and labels

features = np.array(dataset.data, 'int16')

labels = np.array(dataset.target, 'int')

# Extract the hog features

list\_hog\_fd = []

for feature in features:

fd = hog(feature.reshape((28, 28)), orientations=9, pixels\_per\_cell=(14, 14), cells\_per\_block=(1, 1), visualise=False)

list\_hog\_fd.append(fd)

hog\_features = np.array(list\_hog\_fd, 'float64')

# Normalize the features

pp = preprocessing.StandardScaler().fit(hog\_features)

hog\_features = pp.transform(hog\_features)

print "Count of digits in dataset", Counter(labels)

# Create an linear SVM object

clf = LinearSVC()

# Perform the training

clf.fit(hog\_features, labels)

# Save the classifier

joblib.dump((clf, pp), "digits\_cls.pkl", compress=3)

**Code-(Testing Classifier)**

# Import the modules

import cv2

from sklearn.externals import joblib

from skimage.feature import hog

import numpy as np

import argparse as ap

# Get the path of the training set

parser = ap.ArgumentParser()

parser.add\_argument("-c", "--classiferPath", help="Path to Classifier File", required="True")

parser.add\_argument("-i", "--image", help="Path to Image", required="True")

args = vars(parser.parse\_args())

# Load the classifier

clf, pp = joblib.load(args["classiferPath"])

# Read the input image

im = cv2.imread(args["image"])

# Convert to grayscale and apply Gaussian filtering

im\_gray = cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

im\_gray = cv2.GaussianBlur(im\_gray, (5, 5), 0)

# Threshold the image

ret, im\_th = cv2.threshold(im\_gray, 90, 255, cv2.THRESH\_BINARY\_INV)

# Find contours in the image

ctrs, hier = cv2.findContours(im\_th.copy(), cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

# Get rectangles contains each contour

rects = [cv2.boundingRect(ctr) for ctr in ctrs]

# For each rectangular region, calculate HOG features and predict

# the digit using Linear SVM.

for rect in rects:

# Draw the rectangles

cv2.rectangle(im, (rect[0], rect[1]), (rect[0] + rect[2], rect[1] + rect[3]), (0, 255, 0), 3)

# Make the rectangular region around the digit

leng = int(rect[3] \* 1.6)

pt1 = int(rect[1] + rect[3] // 2 - leng // 2)

pt2 = int(rect[0] + rect[2] // 2 - leng // 2)

roi = im\_th[pt1:pt1+leng, pt2:pt2+leng]

# Resize the image

roi = cv2.resize(roi, (28, 28), interpolation=cv2.INTER\_AREA)

roi = cv2.dilate(roi, (3, 3))

# Calculate the HOG features

roi\_hog\_fd = hog(roi, orientations=9, pixels\_per\_cell=(14, 14), cells\_per\_block=(1, 1), visualise=False)

roi\_hog\_fd = pp.transform(np.array([roi\_hog\_fd], 'float64'))

nbr = clf.predict(roi\_hog\_fd)

cv2.putText(im, str(int(nbr[0])), (rect[0], rect[1]),cv2.FONT\_HERSHEY\_DUPLEX, 2, (0, 255, 255), 3)

cv2.namedWindow("Resulting Image with Rectangular ROIs", cv2.WINDOW\_NORMAL)

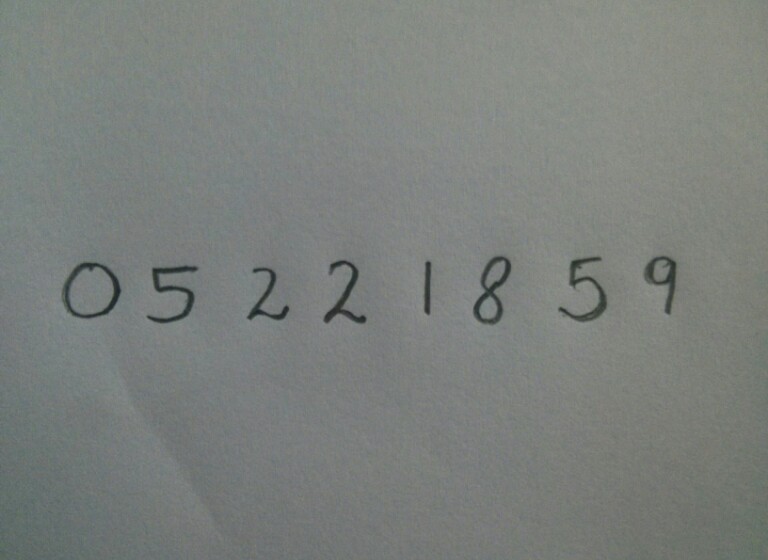
cv2.imshow("Resulting Image with Rectangular ROIs", im)

cv2.waitKey()

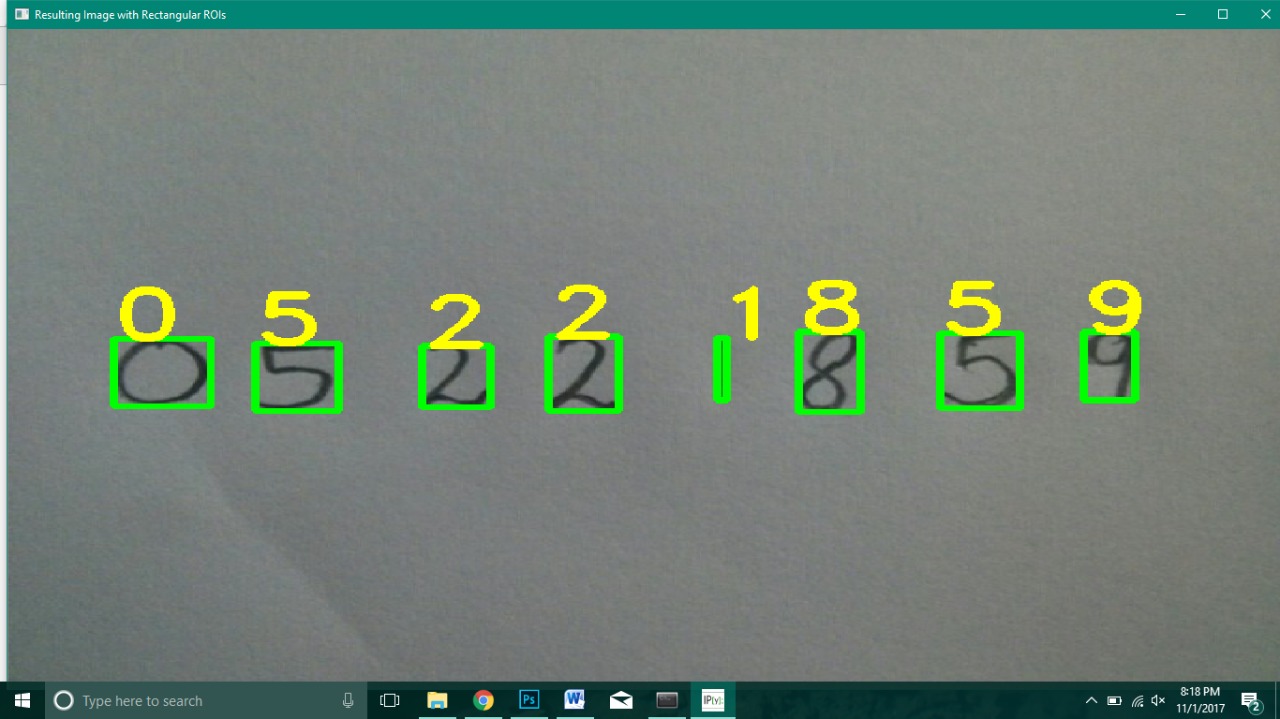
**Results and Observations**

Two sets of digits were taken into consideration to test the algorithm and the outputs were observed. The accuracy seen in the algorithm is around 92%. It can be used for day to day purposes if enhanced on a small scale level. The observed outputs are as :-

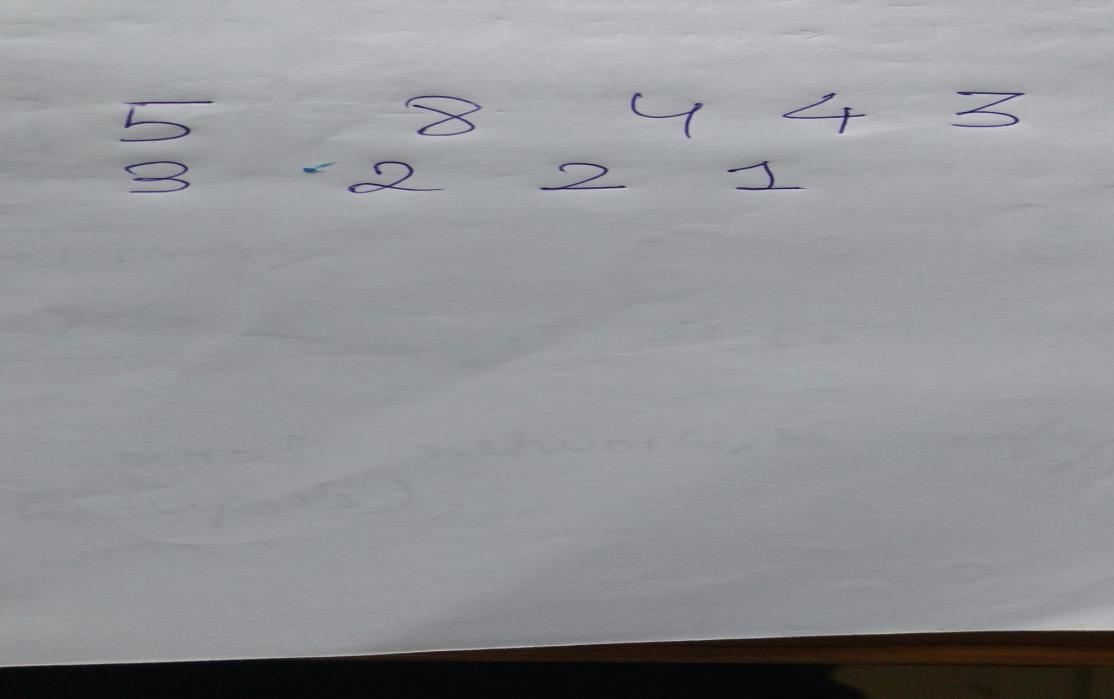
Input 1 :-



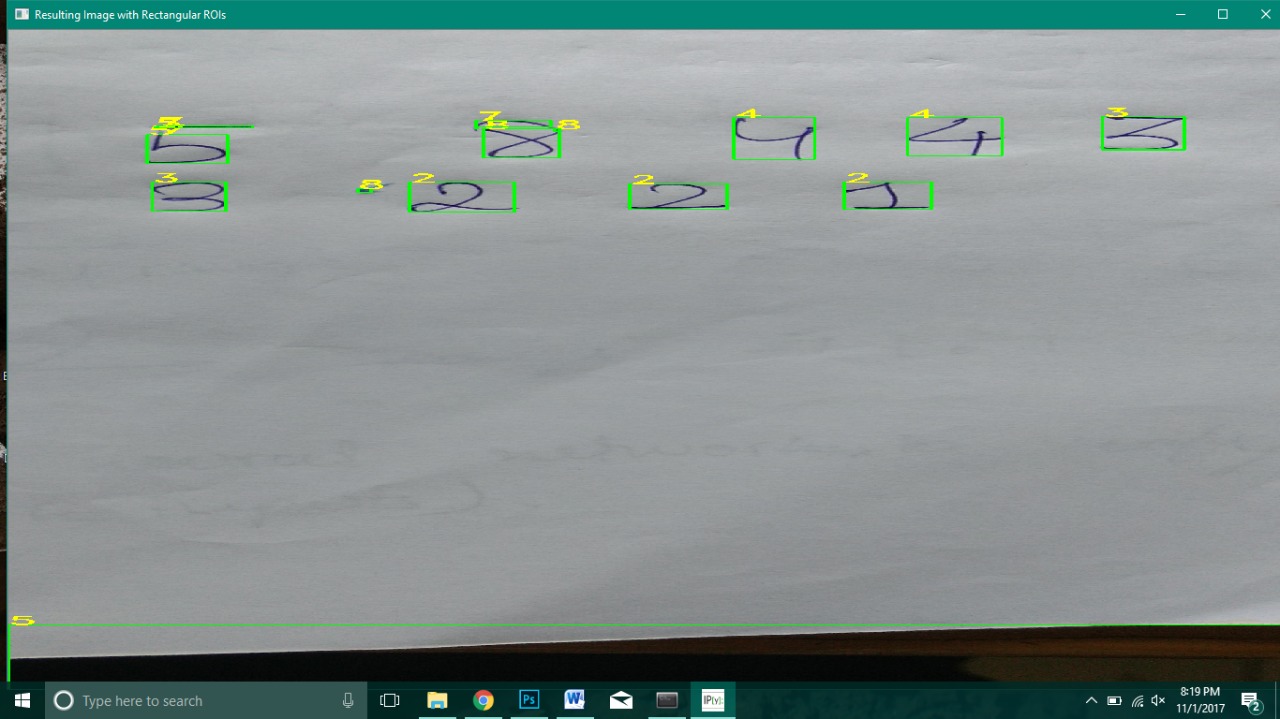
Output 1 :-



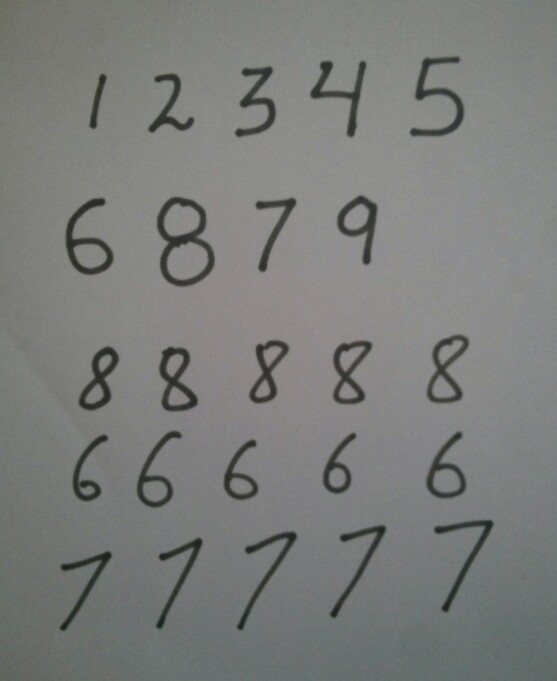
Input 2 :-



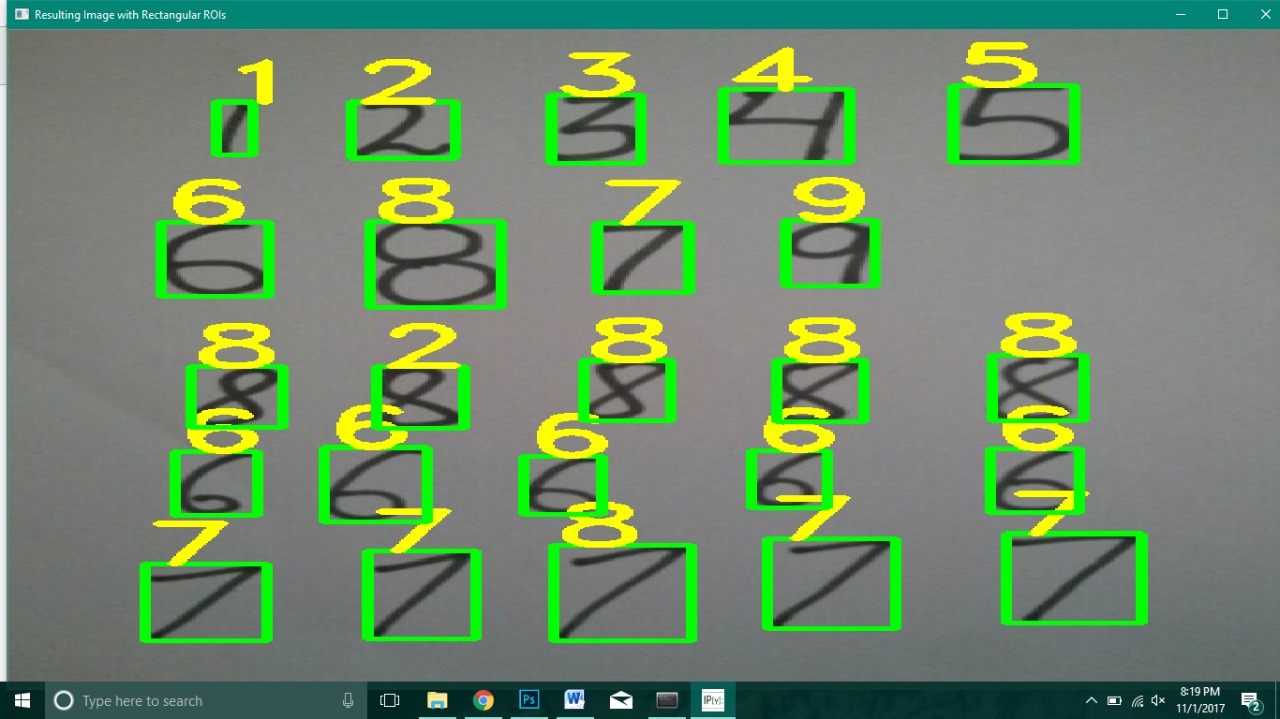
Output 2 :-



Input 3 :-



Output 3 :-



**Need of Digit Recognition**

In this fast moving, evolving world there is a need of systems to fasten our works and in this the processing of handwritten documents plays a major role. We several times observe that while at the banks we see the workers at the bank looking at those number and many times they cannot identify the numbers due to several reasons , here comes the need of this algorithm where it can observe the digits, makes the project faster and even helps in differentiating the digits due to its vast MNIST database. It can also be used for directories and also for observing the number plates of vehicles for several purposes. This change in this world will help the humans and will also show the advancement of our society in this pacing environment.

**CONCLUSION**

We can conclude that we reached the computer to the human’s brain by the importance use of isolated digits recognition for different applications. This recognition starts with acquiring the image to be preprocessed throw a number of steps. As an important point, classification and recognition have to be done to gain a numeral text. In a final conclusion, neural network seems to be better than other techniques used for recognition.