

HAND DIGIT RECOGNITION USING ARTIFICIAL INTELIGENCE

(INT 404)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

SR. NO.	NAME	ROLL No.	REG. No.
1	VISHNU ARYAN SINGH	K18ZV B61	11802524
2	NABEEL HAMZA	K18ZV B62	11803995
3	ANGUN BOKO	K18ZV B60	11802684
4	NITHIN SANJAY BIRADA	K18ZV B50	11805649



School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

ACKNOWLEDGEMENT: -

I would like to thank my mentor – Madam Kunj Marken for her advice and inputs on this project. Many thanks to my friends and seniors as well, who spent countless hours to listen and provide feedbacks.

BONAFIDE

Certified that this project is the bonafide work of Nabeel,

Angun, Nithin and Vishnu under supervision of

Madam Kunj Marken

25669

Intelligent system

TABLE OF CONTENTS

CONTENTS	PAGE NO.
CHAPTER 1: INTRODUCTION.....	
1.1 PURPOSE	
1.2 BENEFIT OF THIS PROJECT	
1.3ADVANTAGES	
1.4 APPLICATION	
CHAPTER 2: LIBRARIES USED.....	
CHAPTER 3: CODING.....	
CHAPTER 4: REFERENCES	
.....	

CHAPTER 1: INTRODUCTION

Handwriting recognition (HWR), also known as **Handwritten Text Recognition (HTR)**, is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. The image of the written text may be sensed "off line" from a piece of paper by optical scanning (optical character recognition) or intelligent word recognition. Alternatively, the movements of the pen tip may be sensed "on line", for example by a pen-based computer screen surface, a generally easier task as there are more clues available. A handwriting recognition system handles formatting, performs correct segmentation into characters, and finds the most plausible words.

- This is built using artificial algorithms that analyzes user's digit and understand user's message.
- This System is a web application which provides exact number he/she has written by very effectively.
- Students just have to upload their digit.

The System uses built in artificial intelligence to answer the problem.

1.1 PURPOSE

The purpose of this project is correctly identify digits from a dataset of tens of thousands of handwritten images. with curated a set of tutorial-style kernels which cover everything from regression to neural networks

1.2 BENEFITS OF CHABOT

Accessible anytime: Since this code are basically virtual robots they never get tired and continue to obey your command.

Handling capacity: Unlike humans who can only communicate with one human at a time, This can simultaneously gave answers to thousands of people.

Customer Satisfaction: This system runs on a data with percentage importance so making sure that the data given by the customer is same as the result.

Work Automation: This can now automate the tasks which are to be done frequently and at the right time

1.3 ADVANTAGES

- Through the training phase, the system can adapt to the writing style of the use
- it is used in PDA and tablet PC
- Handwriting recognition plays a big role in the technology world now. It also plays an important role in the storage and in the recovery of critical handwriting information.
- This handwriting recognition ensures an accurate medical care and it also reduces storage costs

1.4 APPLICATIONS

Applications of offline handwriting recognition are numerous: reading postal addresses, bank check amounts, and forms. Furthermore, OCR plays an important role for digital libraries, allowing the entry of image textual information into computers by digitization, image restoration, and recognition methods

CHAPTER 2: LIBRARIES USED

1. PANDAS

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

2. SKLEARN

Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbors, and it also supports Python numerical and scientific libraries like NumPy and SciPy

3. NUMPY

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

4. TFLEARN

TFlearn is a modular and transparent deep learning library built on top of Tensorflow. It was designed to provide a higher-level API to TensorFlow in order to facilitate and speed-up experimentations, while remaining fully transparent and compatible with it. Easy-to-use and understand high-level API for implementing deep neural networks, with tutorial and examples.

TFLearn features include:

- Fast prototyping through highly modular built-in neural network layers, regularizes, optimizers, metrics...
- Full transparency over Tensorflow. All functions are built over tensors and can be used independently of TFLearn.
- Powerful helper functions to train any TensorFlow graph, with support of multiple inputs, outputs and optimizers.
- Easy and beautiful graph visualization, with details about weights, gradients, activations and more...
- Effortless device placement for using multiple CPU/GPU.

5. TENSORFLOW

TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.

6. PICKLE

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it "serializes" the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.)

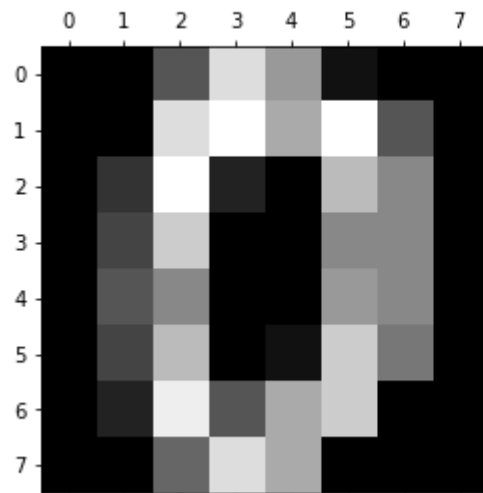
CHAPTER 3: CODING

In this part, let's look at the coding. The following script will only run in python 3.6. Because some of the libraries will not work in the other version or it will show some error.

```
1 import matplotlib.pyplot as plt
2 %matplotlib inline
3 from sklearn.datasets import load_digits
4 digits = load_digits()
5
```

- Import all the libraries. (to install the library, use pip install library name)
- Some of the libraries will only work in specific python version. Make your environment suitable for the script

<matplotlib.figure.Figure at 0x1c7d6ef5d68>



- The above is the digits recognition data, which we will feed as question.

```
6 import pylab as pl
7 pl.gray()
8 pl.matshow(digits.images[0])
9 pl.show()
```

digits.images[0]

```
Out[3]: array([[ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.],
 [ 0.,  0., 13., 15., 10., 15.,  5.,  0.],
 [ 0.,  3., 15.,  2.,  0., 11.,  8.,  0.],
 [ 0.,  4., 12.,  0.,  0.,  8.,  8.,  0.],
 [ 0.,  5.,  8.,  0.,  0.,  9.,  8.,  0.],
 [ 0.,  4., 11.,  0.,  1., 12.,  7.,  0.],
 [ 0.,  2., 14.,  5., 10., 12.,  0.,  0.],
 [ 0.,  0.,  6., 13., 10.,  0.,  0.,  0.]])
```

```

images_and_labels = list(zip(digits.images,digits.target))
plt.figure(figsize=(5,5))
for index, (image, label) in enumerate(images_and_labels[:15]):
    plt.subplot(3, 5, index + 1)
    plt.axis('off')
    plt.imshow(image, cmap=plt.cm.gray_r, interpolation='nearest')
    plt.title('%i' % label)

```



```

import random
from sklearn import ensemble

#Define variables
n_samples = len(digits.images)
x = digits.images.reshape((n_samples, -1))
y = digits.target

#Create random indices
sample_index=random.sample(range(len(x)),len(x)/5) #20-80
valid_index=[i for i in range(len(x)) if i not in sample_index]

#Sample and validation images
sample_images=[x[i] for i in sample_index]
valid_images=[x[i] for i in valid_index]

#Sample and validation targets

```

REFERENCE

1. <https://www.kaggle.com/c/digit-recognizer/data>
2. YOUTUBE AND OTHER SOCIAL MEDIA PLATFORMS