HANDWRITTEN DIGIT RECOGNITION

APRIL 12

Machine Learning Project Report
Authored by: Sriashika Addala

LOVELY PROFESSIONAL UNIVERSITY

What is handwritten digit recognition?

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.

More specifically, the handwritten digit recognition is the ability of computers to recognize human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavors. The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.

Handwritten Digit Recognition System

I. Pre-requisites

In order to build a machine learning model, we would require relevant datasets with the help of whom we shall train the model. Hence we have gathered the MNIST digits dataset from Keras. Samples provided from MNIST (Modified National Institute of Standards and Technology) dataset includes handwritten digits total of 70,000 images consisting of 60,000 examples in training set and 10,000 examples in testing set, both with labeled images from 10 digits (0 to 9). This is a small segment form the wide set from NIST where size was normalized to fit a 20x20 pixel box and not altering the aspect ratio. Handwritten digits are images in the form of 28x28 gray scale intensities of images representing an image along with the first column to be a label (0 to 9) for every image. The same has opted for the case of the testing set as 10,000 images with a label of 0 to 9.

II. Proposed classifiers to train the model

We have selected four different models to train and test for comparison in order to extract the best model for evaluating this project. The following models have been used:

1. Neural network model
2. Convolutional neural networks
3. K-NN model
4. Random forest classifier

Neural network model: Artificial neural networks or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. We have used the python.scikit-learn's in-built digits dataset and have achieved a 92% training and testing accuracy.

Convolutional neural network model: In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks, based on their shared-weights architecture and translation invariance characteristics. Excellent results achieve a prediction error of less than 30%. State-of-the-art prediction error of approximately 0.2% can be achieved with large Convolutional Neural Networks. Input & Output: When a computer or system takes an image, it just sees an array of pixel values. Suppose 480 * 480 * 3 where (480*480) is size, 3 refers to RGB values. Each of these numbers is assigned with a value of 0 to 255 as pixel intensities at that point. The key point is that based on taking the image as an input, computer system predicts and assumes as output for describing the probability of the image being a said or certain class (say 0.90 for class 1, 0.96 for class 2, 0.4 for class 3). Using CNN model, we have achieved a 98% training accuracy and 93% testing accuracy with a maximum testing loss of 27%.

k-NN model: KNN is the non-parametric method or classifier used for classification as well as regression problems. KNN explains categorical value using majority votes of K nearest neighbors where the value for K can differ, so on changing the value of K, the value of votes can also vary. Different distance functions used in KNN are:

1. Euclidean function
2. Manhattan function
3. Minkowski
4. Hamming distance
5. Mahalanobis distance

Using KNN classifier, we have achieved an 86% training accuracy and 77% testing accuracy.

Random forest classifier: Random forests are an ensemble learning method that can be used for classification. It works by using a multitude of decision trees and it selects the class that is the most often predicted by the trees. Each tree of the forest is created using a random sample of the original training set, and by considering only a subset of the features (typically the square root of the number of features). The number of trees is controlled by cross-validation. Using RandomForestClassifier, we have achieved 95.7% training accuracy and 87.3% testing accuracy.

III. Language/Platform Used

The language used for building this project is Python 3.7. I've also used Jupyter notebooks by Anaconda Inc. This project runs equally swift on both Windows and Linux systems.

IV. Libraries Used

Python library is a collection of functions and methods that allows you to perform many actions without writing your code. For example, the Python imaging library (PIL).is one of the core libraries for image manipulation in Python. Pillow is an actively developed fork of PIL. Open-CV Python is a library of Python (also binds to C++, C# etc.) aimed at real-time computer vision and image processing. It makes use of NumPy, another library for numerical operations. Each library in Python contains a huge number of useful modules that you can import for your everyday programming. I've used the following libraries:

1. NumPy
2. Matplotlib
3. Pandas
4. Scikit-learn (sklearn)

5. TensorFlow
6. Keras
7. Warnings

V. Visualization of the inputs/outputs

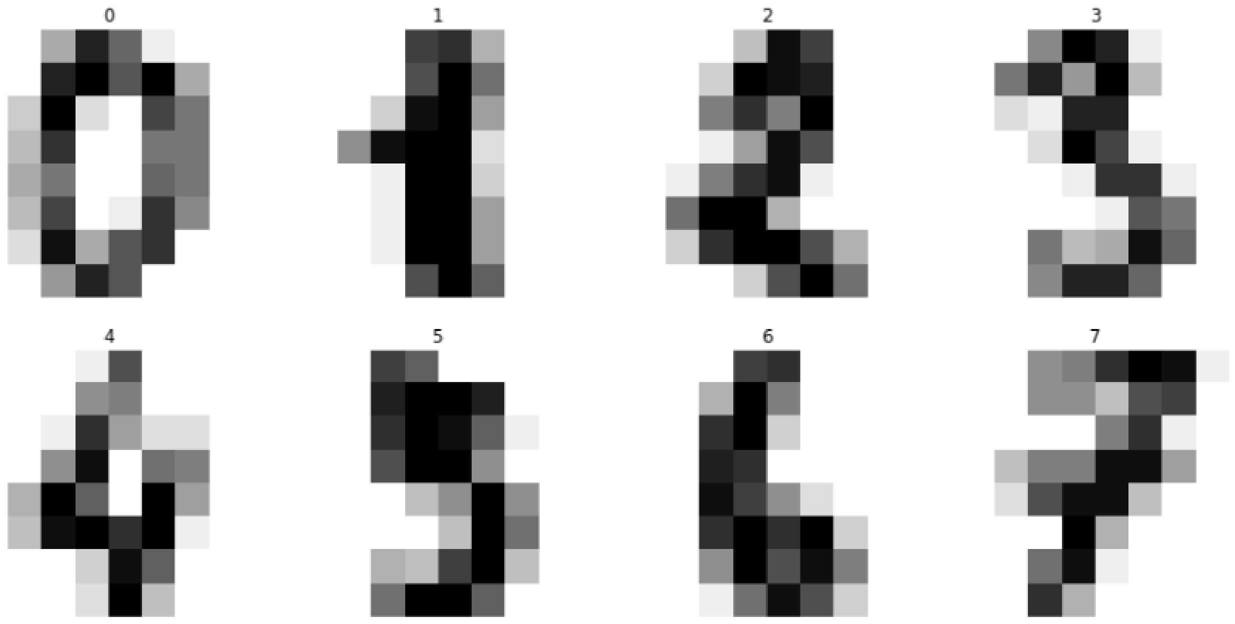

Fig.1. Training outputs where model is trained to recognize unclear images containing numbers

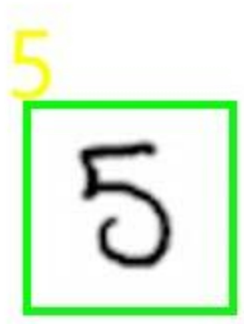

Fig.2. Output image is boxed, and then recognized

VI. Important links to project

Project URL: https://github.com/Lovely-Professional-University-CSE/int247-machine-learning-project-2020-kem031-rollno-27.git

VII. Results of all models

S.No.	Category of comparison	kNN	NN	CNN	Random Forest
1	Training Accuracy	86%	92%	98.2%	95.7%
2	Testing Accuracy	77%	92%	93.3%	87.3%
3	Classification Accuracy	81.5%	92%	95.75%	91.5%

Stats: The model with highest classification accuracy is the CNN model and the one with least classification accuracy is the kNN model.

VIII. Conclusion

As per the results, it is certain that the convolutional neural networks model is the perfect model for handwritten digit recognition, among the others.

THANK YOU
