Machine Learning project: Hand written digit data recognition expansion

Raesetje Bonjo Sefala, [redacted]

May 23, 2017

1 Abstract

Handwritten digit recognition is an important task in Computer Vision. Although, accurately recognizing Handwritten digit in a document is not an easy task to achieve because of the variations in handwriting, different points of view, cluttered backgrounds,lighting variations etc.
This project aims to increase the number of samples in the MNIST dataset by increasing the number of labeled images with more variations.

2 Introduction

Apart from human labour, we can also recognise handwritten digits using machine learning algorithms. These algorithms get better by using a bigger number of training samples and more varying samples to mimic real life data.

The MNIST dataset which is a database of hand written digits with a homogeneous background, consists of 70 000 hand written digits of 28×28 pixels. This dataset does not have enough images with more variations of things such as different lighting, different heterogeneous backgrounds, e.t.c. and as a result, models trained using this dataset do not perform as well when tested on more complex test samples.

In this project, we will be adding more images to the MNIST dataset. These images will be labeled and have more variations.

3 Methodology

3.1 Training the Classifier with an ANN model

The MNIST dataset is a labeled dataset of images. In order to correctly label the images which will be added, an Artificial Neural Network(ANN) with 784 inputs of pixels, 3 hidden layers and 10 output nodes was trained using 36000 training images, 12 000 validation images and 12 000 testing images. A 91% accuracy was achieved upon testing. The trained model was then used to label the new images.

3.2 Ceating a classifier using Histogram of Oriented Gaussians (HOG) features

Using the MNIST dataset which consists of images of 28×28 pixels, we start by setting the cells per block equal to one and each cell to size 14×14. Since our images are of size 28×28, there will be four blocks of size 14×14 each adding up to 28×28. We also set the size of the orientation vector equal to 9. So our HOG feature vector for each sample will be of size $4\times9 = 36$. We then save the run the function and save the results. We use these features and labels together with a multi classifier Support vector Machine to train the classifier.
We can either use this classifier or the trained ANN for classification but they both gave almost the same accuracy rate.

3.3 Detecting the digits from documents

From the opencv library, we use the Video Capture function so that we can conveniently save a lot more labeled images in real time by pressing a capture button which was previously set. We convert each frame to grayscale and apply a Gaussian filter to the grayscale frame to remove noisy pixels.We then threshold each frame by a value of 90 because we are assuming dark ink on the digit and a lighter coloured paper. This then sets the background to black and the digit to white. We then calculate the contours and then fit a rectangle around the contours. Figure **??** shows the results of this subsection.

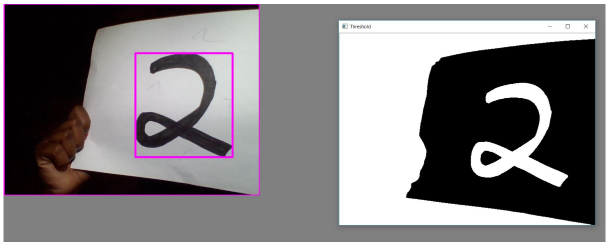

Figure 1: The Image detection

We then take a snapshot of the number inside the rectangle and re-size it to 28×28 pixels and then classify it using the classifier(either ANN or HOG with the SVM).

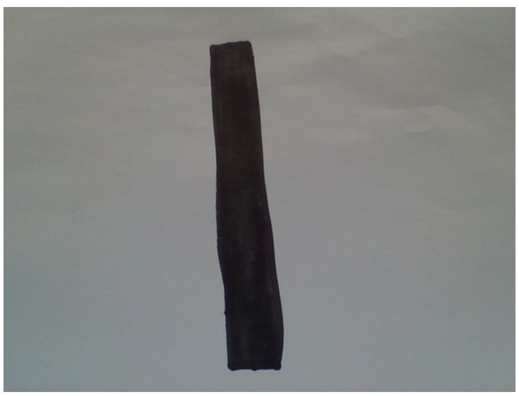

Figure 2: The ideal sample: The digit 1 with a homogeneous background, centered and with good lighting(Like those in the MNIST dataset

4 Observations

After testing the data collection model on 30 samples, it has been observed that:

- A good lighting position is important for more accurate recognition.
- It is also important that the numbers are well spaced from each other in order to avoid confusing the classifier as it was trained on single digits.
- The digits which will be detected should be visible enough in order to obtain better accuracy.

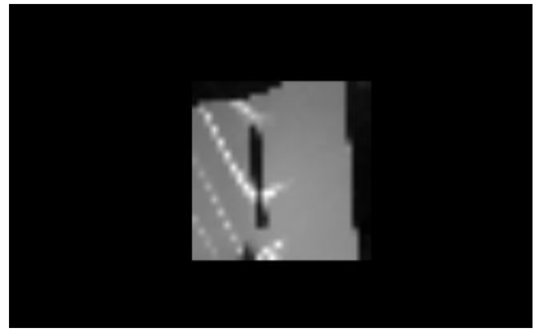

Figure 3: The results of bad lighting in the room. After re-sizing, the digit 1 was misclassified as as the digit 5 due to bad lighting

5 Conclusion

Although the MNIST dataset has been producing good enough results throughout the years, its problem is that models do not generalise well with varying backgrounds found in the real world. This project aimed to expand the MNIST dataset firstly so that we can have more than 70 000 data points to train machine learning models so that they can be better and secondly we wanted to add background variation to the images so that the models can be better adapted to the real world.

Real world data does not always have a homogeneous background and perfect lighting. This project provides a convenient method of expanding handwritten digit dataset using any camera, a trained classifying model and handwritten digits on paper.