Characters and Digits Recognition Using Neural Networks

Yiyue Zhang

February 27, 2017

1 Abstract

In this project, a Neural Network will be constructed on DSP board to perform recognition on handwritten digits and characters. To achieve this, users are required to write some characters or digits (by users' choices) on a white paper with gridlines. Their writings will be captured by camera and will be displayed in printed forms by a monitor. This project has three major parts – pre-training the Neural Network with a dataset on handwritten characters (The Chars74K Dataset), testing the performance with camera inputs from users' handwritten characters and digits, and optimizing the computing speed of the Network. This project is aimed to work for applications, like conversions from handwritten documents to their printed forms.

2 Description of the system

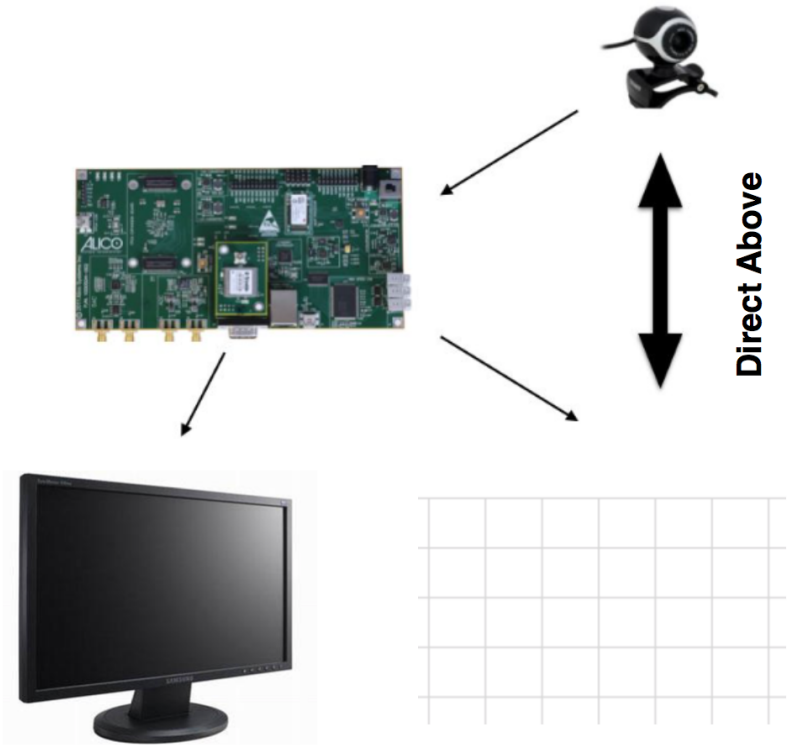

Figure 1: System Setup.

In this project, four equipment parts are required–a camera, white papers with gridlines, a monitor, and a DSP board. Users are required to write characters and digits inside the gridlines on the while paper. And they are required to signal the system the end of their writing. The camera will be place on top of the white paper and be used to capture users' handwritings. Images from the camera will be used as inputs for the Neural Network. After processing by the network, each character should be outputted on the monitor in same physical positions as they are on the white paper. The DSP board will be used to process the Neural Network with a relative fast speed and a high accuracy.

3 Description of Possible Algorithms

There are three possible algorithms for recognizing characters and digits. They are K-nearest-neighbors, Linear classifier, and Convolutional net LeNet-5.

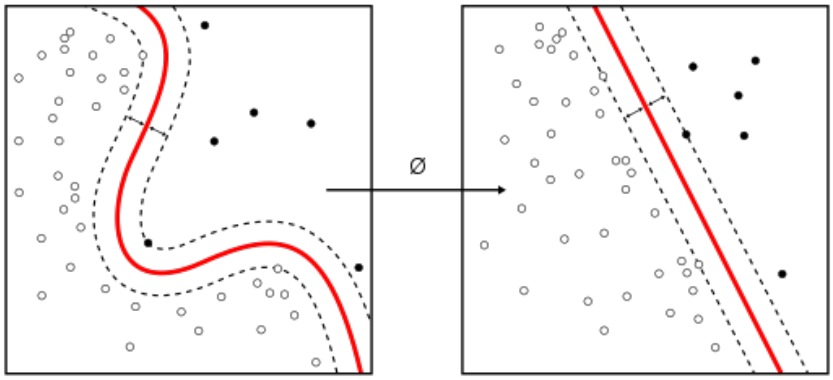

Figure 2: K-nearest-neighbors.

For K-nearest-neighbors, it is used mainly for classification and regression. In K-nearest-neighbor, there are k different classes. And classes are represented by centroids. Each input will be correlated to a feature in feature space. The aim is to classify each input (their features) to one of the classes basing on the majority vote (closest distance to centroid), like in Figure 2. Using this, characters and digits can be separated to different classes. This algorithm can help in recognition process.

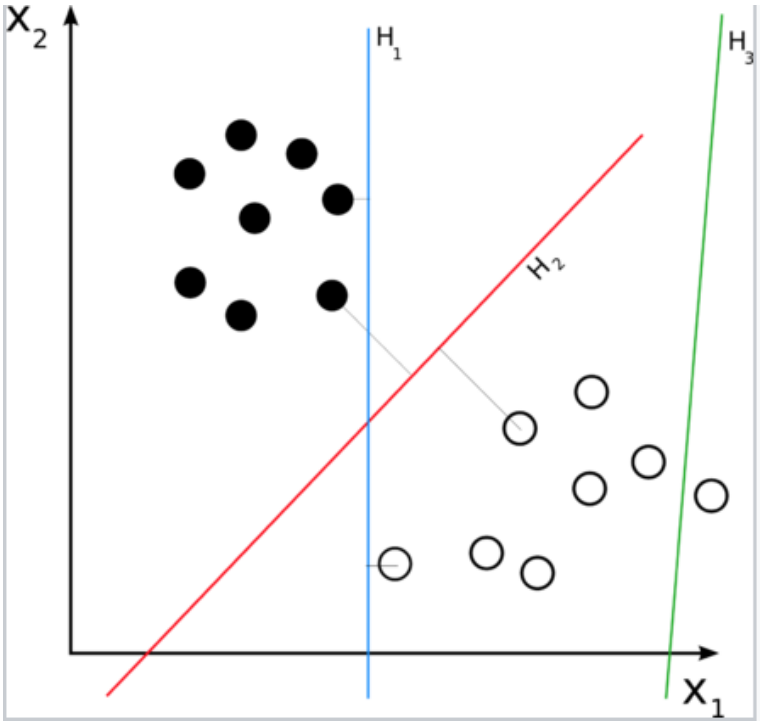

Figure 3: Linear Classifier.

$$y = f(\vec{w} \cdot \vec{x}) = f\left(\sum_j w_j x_j\right)$$

Figure 4: Linear Classifier Algorithm.

The second algorithm is Linear classifier. Linear classifier is used widely in field of machine learning. It is mainly used

for classification basing on the characteristics of inputs. In this algorithm, each input will be mapped to a feature vector. And, a weight vector will be obtained from the training process. It performs classification for each input basing on the dot product of the weighted vector and the feature vector of each input, shown in Figure 3 and 4.

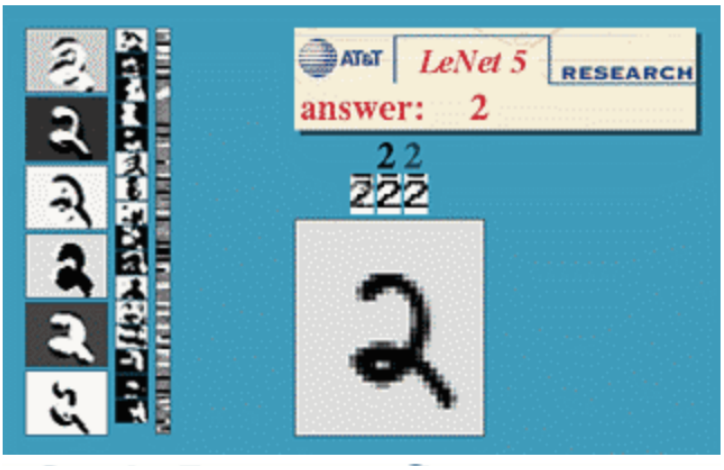

Figure 5: LeNet-5.

The third algorithm is Convolutional net LeNet-5. This is a work from Professor Yann LeCun, who was working mainly on digit recognition. LeNet-5 is his latest convolutional neural network, which was designed for handwritten and machine-printed character recognition, shown in Figure 5. LeNet-5 has a very high accuracy in digits recognition. To apply in this project, modifications of the network are required in order to perform both digits and characters recognition.

4 Complexity Analysis

In this project, the complexities come from five different aspects. The first aspect is the memory size of the Neural Network. The size of the Neural Network should fit to the memory space on DSP board. Less layers inside the Neural Network can reduce the memory requirement from DSP board, but can also lead to a lower accuracy. The second aspect is from the training of the Neutral Network using TensorFlow. This project is based on applying a pre-trained Neural Network on DSP board. Choosing right algorithms for the Neural Network are essential. Achieving a high accuracy from the training is another complexity. The fourth complexity is from the testing speed. In this project, a relative high speed in recognition is required. The results will be outputted on the monitor for users to see. The wait time by users should be low considering on real life applications. The last complexity is from the display of characters and digits. In this project, there are a total of 62 classes to be displayed (26 UpperCase + 26 LowerCase + 10 Digits).

5 Major Challenges

Basing on Complexity Analysis, the challenges are coming from similar aspects. First, this project requires training networks using TensorFlow. Thus, learning to use TensorFlow is one of the challenges. The second challenge is from the sizing requirement. A deep understanding on Neural Networks are required for simplification of the network in order to fit well on DSP board, meanwhile, keep a high enough accuracy. The third challenge is from camera input. In Machine Learning, the training set and testing set are both provided by the dataset. The testing set is for evaluating the performance of Neural Networks. In this project, testing set will be used for evaluation in early stage for optimizing the accuracy of the Neural Network. Later, camera inputs will feed to the network for recognition. Therefore, the camera's resolution and position, and characters' font sizes and qualities are essential to good performance of the project. Last challenge is from the testing speed, the c code needs to be optimized to make functions to run faster, such as using shifts instead of multiplications, integers instead doubles.

6 Modeling Tasks

Dataset Chars74k, which is mainly used for on-line handwritten character recognition methods, will be used in this project to train and test the Neural Network. This dataset consists of 62 classes (26 UpperCase + 26 LowerCase + 10 Digits) with 55 samples per class. In addition, the pen stroke trajectories are provided. Samples from Dataset Chars74k are shown in Figure 6.

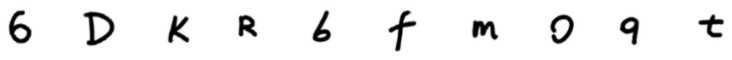

Figure 6: Rough Schedule.

7 Human Factors

This project requires users to provide their handwriting characters and digits on a given white paper with gridlines. Therefore, the size of characters that user writes is important. If the handwriting is too small, it will make the recognition process harder due to the effect from the camera's resolution. Similarly, for a good detection on handwriting characters, writing inside gridlines or outside gridlines is also important. Most importantly, the quality of handwriting can influence the performance as well, for example, care writing can lead to a high accuracy, and poor writing will result more confusions.

8 Training

This project requires training on a small neural network. The training and testing samples are from The Chars74 dataset.

9 Rough schedule

	Best Case	Worst Case
Constructing and Training Neural Network	Feb 27-03	Feb 27-03
Display of 62 different characters on screen	March 06-10	Feb 27-03
Applying NN to DSP	Spring Break	Mar 20-24
Finishing Up and Testing	Mar 20-24	Mar 27-31
Optimizing computational Speed	Mar 27-31	Before April 5th

Figure 7: Rough Schedule.

During the remaining time of this semester, there are five milestones–constructing and training the Neural Network, display 62 characters and digits on a monitor, apply Neural Network to DSP board, finishing up and testing, optimizing the performance and computing speed. Major time breakdowns corresponding to each milestone of this project are shown in Figure 7.

10 Final Test Set-Up

During testing, user should write any characters (Uppercase or Lowercase) and digits on a paper with gridlines. When user finishes writing all characters/digits, user needs to hit a button to signal the finish of their writing. After a relative short amount of time, the handwritten characters/digits should be displayed on the monitor in their printed forms. The accuracy of this transformation should be high. Mistakes, like distinguishing 0-O, 1-l, can be forgiven.

11 Board

In this project, Texas Instruments DaVinci TMS320FM6437 board will be used and with connection to a video camera and a monitor. Additional software, like MATLAB and xCode, might be used to perform training and testing out network performance.

12 References

[1] [4] [2] [6] [3] [5]

References

[1] T. deCampos. The chars74k image dataset - character recognition in natural images. 2012.

[2] J. S. Gray H. Almohri and H. Alnajjar. A real-time dsp-based optical character recognition system for isolated arabic characters using the ti tms320c6416t. 2008.

[3] P. of the IEEE. Mnist handwritten digit database, yann lecun, corinna cortes and chris burges. 2012.

[4] Quinnliu. Machinelearning/unsupervisedlearning/neuralnetworks/digitrecognition at master quinnliu/machinelearning. 2014.

[5] LeCun Yan. Mnist demos on yann lecun?s website.

[6] LeCun Yan. Handwritten digit recognition with a back prorogation network. 1994.