The Brand Recognition Using Machine Learning from Google Street View Images

Alper Frat KAYA
Hacettepe Univesity
Ankara, Turkey
alperfiratkaya@gmail.com

Hulya Sermin KARAKAS
Hacettepe Univesity
Ankara, Turkey
hulyasermin.karakas@gmail.com

Abstract

Brand recognition in natural images like Google street view images is a challenging problem that contains both computer vision and machine learning inside. Most of the projects in this area are focused on the optical character recognition which is not applicable for the natural images.

1. Introduction

The main problem that we focused on is that the banks or small stores do not have tags in the Google Maps and when a person try to find that place, it is really hard to reach that point and he/she could not be hundred percent sure about the correctness of the place in the given address. With our approach, from the Google Street View images, we could tag the places and tag from map.

In this project, we are planning to detecting brands by recognizing characters. We will train the program by alphabet dataset which has lots of different fonts, then operate it with google street view images. The project has three main steps which are the localization, classification and recognition parts. In general, localization finds the borders in the image by the computer vision techniques, in the classification part, the program classify whether the given image contains a letter or not and finally, the recognition part detects which character in the image. Thus, a system that can locate and recognize text could be used for many other purposes like street name detection, a virtual eye application for the blind people or

The recognition of the natural images is a much harder task than we planned from the beginning of the project. The traditional approaches have been focused on the optical character recognition. In addition, there are very few researches about the natural image recognition and those researches were made at least 8 - 9 years ago.

2. Related Work

Text recognition, also could be called scene text recognition, is one of the old and main problems of the machine learning. Researches are using text recognition for lots of areas from the cars plate detection to street name detection. It provides us to extend our perspective about the usage area of machine learning.

There are many ways to train the data but the other researches [4] [5] show that, the best approach for character recognition for natural images is the some kind of the neural network. In addition, there are very limited number of dataset for text recognition on natural images. Thus, most of the researches related scene text recognition are used same datasets [4] [5]. Because of that, researches are get very similar accuracy values for the problem.

Other than differences, we could say about the inspired parts of the related works are definitely the approaches to the problem and the dataset usages.

3. The Approach

The approach that we implement is called “Multilayer Perceptron” which could also named “Multilayer Neural Network”. Multilayer perceptron is a class of feedforward neural network which means it feedforwards and backpropagate itself. There are at least three nodes in a MLP which are input layer, hidden layer(s) and output layer. There are weights for each element in the nodes and those elements initializing randomly at the beginning of the algorithm. After that, by forward and back propagation with the help of perceptron, we are trying to find the best weight values. In addition, we are using some activation and loss functions to train the program. Some example of the activation functions could be sigmoid, ReLu, tanh etc.

$$Wi = Wi + d(n)xi(n)$$

There are two hidden layers used in the network that we construct. The reason behind this choice is that, it is easy

Localization
Classification
Recognition

Figure 1. Main Steps Followed on The Project

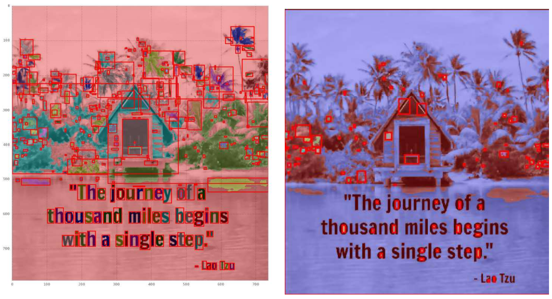
"The journey of a thousand miles begins with a single step."
- Lao Tzu
"The journey of a thousand miles begins with a single step."
- Lao Tzu

Figure 2. Outputs of The Localization Part

and accurate to recognize characters for our research topic. In addition, related works that close to our project was also prefer to use neural networks.

Before starting to implement our method to dataset, it is needed to localize and classify the characters. Thus, we could say that the project contains three parts which are localization, classification and recognition.

3.1. Localization

First step to start project is localization. Localization mainly finds the character like parts of the images and borders them. To do that, it uses the OpenCV which is a computer vision library for the processing image. In this part, because the all team members have no experience with the computer vision or image processing, we used the part of an open source project.[2] On left part of the Figure 2, you could see the outputs that code owner get and on right side of the Figure 2, you could see the outputs we get from the dataset. The huge difference between those outputs are occured because of the resolution of the images of dataset. Our computers do not have a powerful video card to process that image, so we had to decrease the resolution of the image. Another reason of the difference is the noise on the dataset that we trained.

3.2. Classification

Second step of the project is the classification of the images. To send the images to recognition algorithm, it is needed to determine whether the image contains a letter or not. The classification algorithm gets the position of the character like parts of the image and

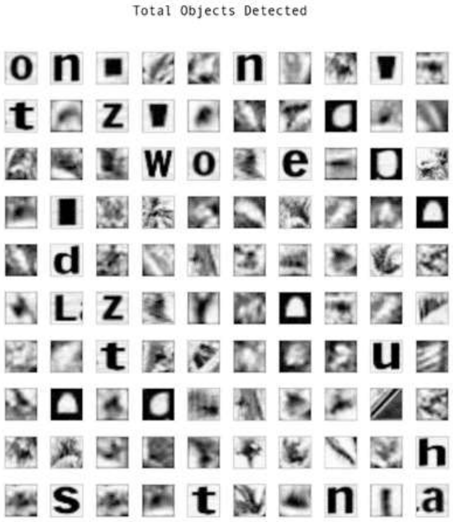

Figure 3. The Images to Classify

3.3. Recognition

Last and foremost step of the project is the recognition part. For the recognition part of the project, we prefered to use neural networks with two hidden layers. We used Char74k dataset to train the algorithm.[3] The dataset contains 74k images with handwritten, natural images and computer generated images. However, we used 25k of the images which are collected from natural images. We resize the images in the dataset to 28 * 28 and make the images greyscale because our computers are not enough powerful for process those images. Unfortunately, it decreased the accuracy of the project.On Figure[4] you can see the details of the neural network. In addition, we used 75 percent of the images to train the data and 25 percent of the images to validate the data.

In the neural network we build,we gave an image as in figure[5] and get probabilities of the possible characters it can be. then select the highest one. We used Keras framework with tensorflow backend.

4. Experimental Results

In this section, it is described of the results and comments of the accuracy values that we get from the experiment. In addition, you could see the visualization results of the project on figure 2 and 9.

Before discussing about the results, We would give some information about the test dataset. As we mentioned before on recognition section, the dataset for training has 25k images. The purpose of the training was the detection and identification of the words. In addition, for the testing, we used manually captured and saved images from all around the world.

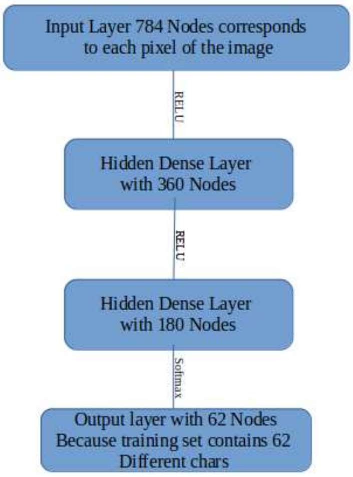
Input Layer 784 Nodes corresponds to each pixel of the image
RELU
Hidden Dense Layer with 360 Nodes
RELU
Hidden Dense Layer with 180 Nodes
Softmax
Output layer with 62 Nodes Because training set contains 62 Different chars

Figure 4. MLP Structrure of The Project

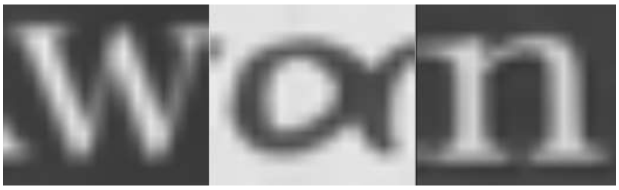

Figure 5. Possible Characters

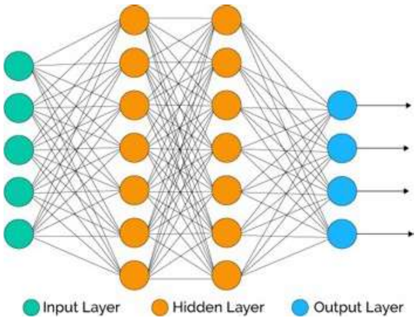

Figure 6. Multilayer Perceptron Visualization

According to our observations, we got 55.62 percent accuracy from the word based approach when we are testing and 83.16 accuracy from the letter based approach when we are validating the dataset. On Figure 7 and 8, it could be seen the loss and accuracy value changing according to time while training the data.In addition, on Figure 9, it could be seen the outputs of the localization with an image from Kizilay Square.

The percentage of the accuracies could be improve with not decreasing the resolution of the images used in training dataset and working on a less noisy and pithy dataset.

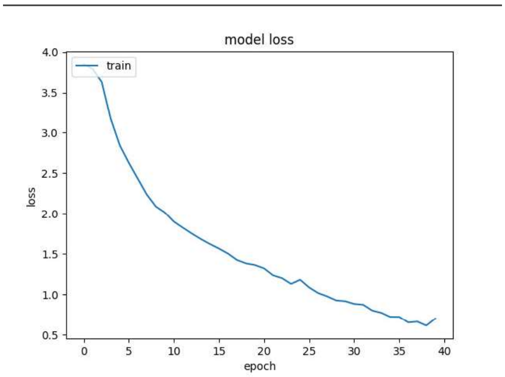

Figure 7. The Loss Function Graphic According to Train Process

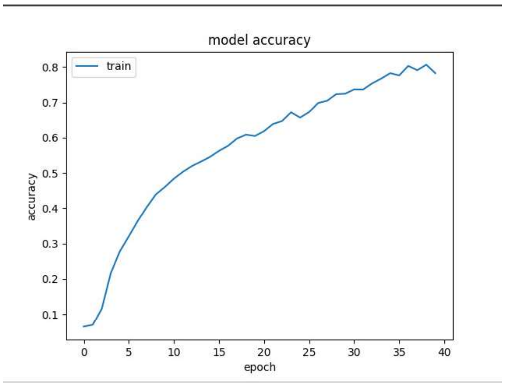

Figure 8. The Accuracy Graphic According to Train Process

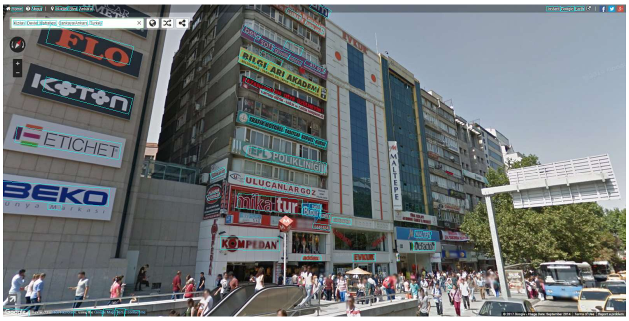

Figure 9. The Output of The Localization

5. Conclusions

As a conclusion, it was a great opportunity to work on machine learning project. The main function of our project was the detecting brand names from the natural images such as Google Street View images. The project successfully completed and it could be used on many different areas other than its purpose which is tagging the places on Google Maps. In the future, it could be used for detecting the street names according to Google Street View images and a road map fpr the companies who want to open a new place according to contribution of the other brands in the nearby places.

6. References

1.http://www.wikizero.net/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvTXVsdGlsYXllcl9wZXJjZXB0cm9u

2.https://github.com/argman/EAST

3.http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

4.http://cs231n.stanford.edu/reports/2015/pdfs/jingrui_final.pdf

5.https://cseweb.ucsd.edu/classes/wi10/cse190-a/reports/jlintern.pdf