Hand-written digits detection based on Caffe

91.423/523 Computer Vision Final Project, UMass Lowell

Shan Cao

Shan_Cao@student.uml.edu

Abstract

This project use image processing technology to design hand-written digits recognition system, to realize this goal, we apply image segmentation, feature extraction to process the image and test the digits image trough caffe, we find that caffe could fit on good quality digits image, but perform poorly on street view house number image. Possible reasons are Noise on the background, Segmentation error, orientation error, training data incompatible.

Introduction

This project mainly focus on the hand-written digits detection by using feature based extraction for finding the interest region , image enhancement for image pre-processing and Caffe deep learning for final detection. In the past, human are trying to learn the world from eyes, ears, the touch feelings of the body. Feeling cold when the temperature is low, feeling uncomfortable when the sound is too loud, and also feeling been attracted while the eye sight catch the light which reelected from the object. Our human has vision part, our eyes, we could detect the color, intensity and also depth from the real world view, the ability of detecting these properties is not enough, luckily, the brains enable the very highly cognitive calculation, all the different stimulating signals that come from different part of the body are been processed in the human brain. Then human could understand the fire which is burning with high temperature, the snow could melt on the skin with a chilling feeling and a lot of other good stuffs. However interesting things are human could not only understand different feeling, but also could get the meaning of different symbol, such as feeling good when hear comfortable music, knowing the meaning of word when words are caught in our eye sights. These are pretty amazing, but here come a question, are human beings born to know music or understand the meaning of the words? The answer is no, when humans are babies, we cannot talk at the beginning, the only sound that human could make is cry, but for most of the people in the world, it seems to be hard to translate these crying sound into meaningful languages, but later on, when babies are grown up, they could acquire tons of new skills, here come another question, how babies acquire these now abilities? As the study of professor B.J. Casey a, Jay N. Giedd b, Kathleen M. Thomas (2000), Despite the significant achievement in the fields of pediatric neuroirnaging and developmental neurobiology, surprisingly our human still know little about the reason why we could acquire these skills. It is the magical power of neurons which we know little about it.

Despite our human beings have not resolve these magical powers but it do no harms for us to enjoy the great gift that given by God, and also nowadays with the increasing popularity of machine learning which has already been capable by the computers with great calculate capabilities, the machine itself could learning to do the "baby works" discussed in previous passage. The machine is able to learn the property of the data such as sound and image, and could try to classify the difference of the data and cluster them into different class.

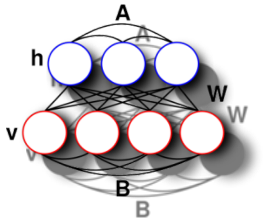


Figure1: this is a simple classifier of neural networks

The image above is an example of machine learning, after training the networks, the machine is capable of predicts new data A into result B. The main purpose of this project is to use machine learning to implement basic hand-written digits detection, hand written digits detection was urgent needed in the past several decades.

As quote from USPS Comprehensive Statement 2001:

"Letter mail recognition rates continued to rise in 2001 as we deployed additional hardware and software upgrades for our existing Multiline Optical Character Reader (MLOCR), Delivery Bar Code Sorter Input Output Subsystems (DIOSS) and Remoter Computer Readers (RCRs). The RCR 2000 program resulted in a 75 percent handwritten address encode rate."

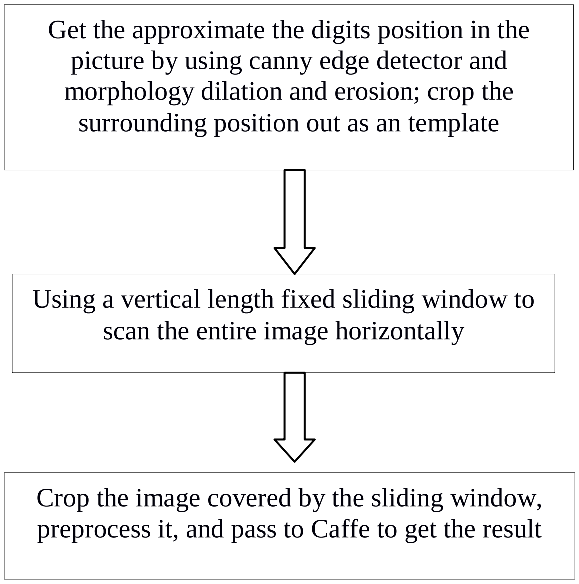
Till the end of 2001, the hand-written digits detection has saved a lot of mailman's work. For now, the hand-written digit might seem to be a simple task, but the concept for detect and hand written digits and detect some other complex image are almost the same, through analysis digits image and the result to find what kind feature could affect the final detection result.

Background

Quite a lot of researchers has done similar hand-written digits recognition, such as Yuchum Lee(1991) did in is paper, He try to compare result from different detection method such as radial basis function (RBF) networks, k nearest-neighbor (kNN) and back propagation network. On a large handwritten digit database, radial basis function (RBF) networks, and k nearest-neighbor (kNN) both provide low error rates. The back propagation network is very good at memory usage and calculation time but can provide "false positive" result even the input is not a digit, this is for the final recognition part, However, for regular use, the data will not be a simple clean one, the data might on a complex background which is hard to detected, and also the digits could be written in connected digits, just as de Santana Pereira, C. and Cavalcanti, G.D.C. (2011) discuss in their paper, they need to fist preprocessing the digits, then let the machine learning algorithm to handle the final step. For the situation of complex background, as Goodfellow, Ian J. , Bulatov, Yaroslav .etc(2013) presented, by preprocess by subtracting the mean of each image, do not use any whitening on the image and local contrast normalization could help increase the efficient of the image. In this project, I am going to use canny edge detector and morphology to locate the digit position on the image, crop the position out as an template and scan the entire template for possible candidates of digits.

Approach

In this project, I trying to detect the hand written digits, I use 3 steps to get the final result.



Get the approximate the digits position in the picture by using canny edge detector and morphology dilation and erosion; crop the surrounding position out as an template

Using a vertical length fixed sliding window to scan the entire image horizontally

Crop the image covered by the sliding window, preprocess it, and pass to Caffe to get the result

I will explain each part here;

1. Locate the digits position in the image

possible segmentation method is using a sliding window to scan the entire image, but sliding window is a vry computational expensive like the pseudocode shown below.

```
define window size=[3 3]
void slding_window
 {
          for (y=1,y< Last_Col_of_image,y=y+2)
                    for (x=1,x< Last_Row_of_image,x=x+2)
                    {
                Crop the image out for processing
                    }
         }
}
```

For a 32by 32 image, the entire number of sliding window is (32-3)X(32-3)=841, this is the number for one scale, cause for reality, we do not what size of digit it is, so to make sure we did not miss any important digit, the program need a lot different scale sliding windows, which could increase the price of computation.

So here what I use is use the canny edge detect to get the edge of the image, then use image morphology to dilate the image, to fill the hole on the image, cause the place which has the digits should have more contrast than other place, so we could get relative more canny edge than the other positions, after the erosion and dilation, the noisy back ground will eliminated and the place which holds the digits will become a full connect area, there might still have some noise in the back ground, but we could choose the connected area which has the biggest area in the image, which is the possible position for the digits .

2. Scan the crop image horizontally

   After get the position for the digit, we crop the image from the original image according to the image position, then use the sliding window which with a fix height which is exactly the height of the cropped image, so for this time, if we still use a 32by 32 image to test, the entire number of sliding window would only be 32-3 which is 29, so we could effectively save calculation time here.

3. Sent each image to Caffe for final detection.

   For this step, Caffe provide already trained Mnist library which could directly accessed through Matcaffe.mexa64, which is a matlab wrapper for Caffe, I wrote a m function named lenet:

   [valide,value,score]=lenet(input_image)

   The output valide means whether this is a valid output, the value means which hand written digits it is in the image, and the score means how many percentage it is which Caffe is sure about the answer.

Table Error! No sequence specified. **Summary of Submitted Code**

filename	**description**	**author**
Seg.m	Segmented the target digits area	Shan Cao
slidinggH.m	Horizontally sliding through the image, and give out the result of this number	Shan Cao
Lenet.m	Predict the what number it is, and give out the 3 outputs, valide,value,score	Shan Cao
Test_32_32_data	Test the slidingH.m on the SVHN gound truth data	Shan Cao
Compare.m	Compare the data style from SVHN, Mnist, and Andrew Ng'data Try to find out the reason why the final out put has a huge error rate	Shan Cao
Test_my_own.m	Test my own hand written digits to see test the accuracy of caffe	Shan Cao
/test_Mnist_data/project.m	Test caffe on its own test data	Shan Cao
Add_noise.m	Try to analysis the original test data which has been added different scale of Gaussian white noise	Shan Cao
/caffe_andrew/project.m	Test caffe on Hand written digits data from Stanford machine learning online course	Shan Cao

Dataset

3 data set was used here:

(1) Mnist

The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

(2) SVHN

SVHN (street view house number)is obtained from house numbers in Google Street View images.

(3) Hand written digits data from Stanford machine learning on-line course. This is the handwritten digits provided by Professor Andrew Ng. The digit is centered in a fixed-size too

Evaluation

(1). At the beginning I test the caffe out put on my own hand written digits, there only one image in the picture, and I hand written 14 digits in a pain paper, the result is very promising, caffe detect them out all (code: test_my_own.m)

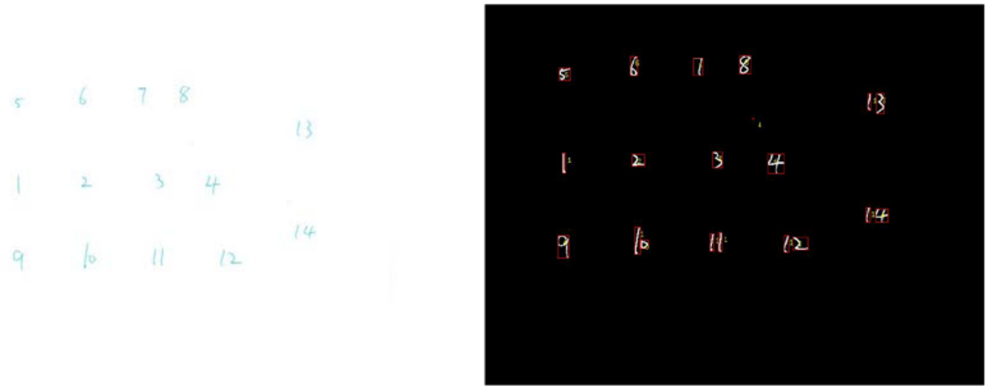


Figure 1: this is the result of my own hand written digit

(2). Then I test the caffe with the original Mnist test data and another hand-written digits data, these two dataset both has ground truth, and also these two datasets are brand new to the caffe network, so this is a very good way to test the caffe network. On the Mnist test set, the result is very promising which is 9890/10000, the accuracy is 98.90 very promising.(code:/test_Mnist_data/project.m)

(3).Next I test caffe on Hand written digits data from Stanford machine learning online course, this data is use in Coursera Machine learning class, an the data is totally new to Caffe which is trained in Mnist. The result is 400/5000. The accuracy is very low, I check the image by myself found that the reason why it turns out to be wrong is the Mnsit and the test data provided by machine learning class is rotated, as the figure 2 shown below. These two different data is in different orientation. So after rotated the data for 90 degree and flip up side down, the data give me 4690/5000, the accuracy is very good now

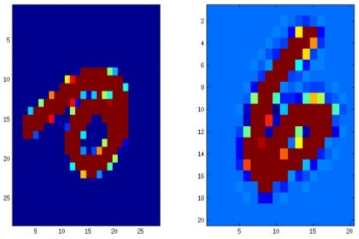


Figure 2: this left the number 6 in Mnist data, the right is number 6 in Andrew Ng' s data

(4). Next I run the test on Google Street view House Number test, to see the result on the real world. There is two different style picture set, as to the figure3, this is the un-cropped image, this image has it own property, it has very huge noise in its background and the numbers are not in the same rotation. So this is a big challenge, so I decide to run my code on second data set, this data set has lower noise in the background, so I try it, first I perform a pre-processing on the image , I use a matlab package on the Internet, it named soft thresholding for image segmentation as to the figure 6 below, this package could help me to get rid of background noise, for this I run test on the all 26032 house number pictures, it only give me 300 truth positive, and the false positive is 16302 , this means that the program could detect the numbers out, but the accuracy is too as to the definition, I need to find out, what kind of reason that cause this result. Instead go to test the program with the un-cropped image which is shown in figure , I change my mind to find out the possible reason that cause this phenomenon.

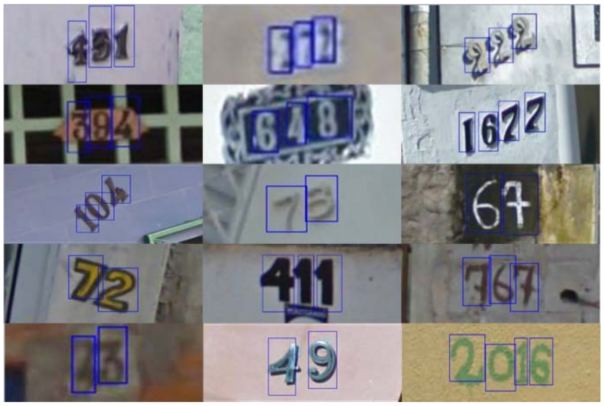


Figure 3:the original un-cropped house number image

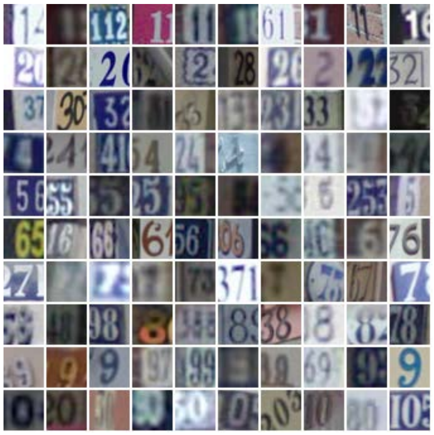


Figure 4: the cropped image

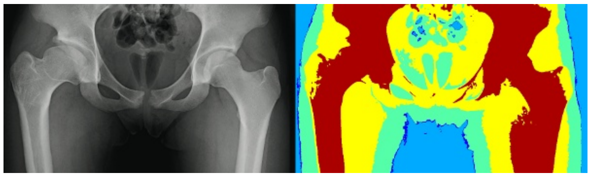


Figure 5: the effect of soft thresholding for image segmentation

The reason that causes such a high false positive and low true positive rate might be:

1. Noise on the background interfere the recognition.

   This might be possible because I run a test on the Minst data, I add Gaussian white noise to the image, the accuracy is suddenly decreased:

Image type	Non-noisy image	(M=1,V=1)	M=10 V=100	M=100, V=10
accuracy	981/1000	43/1000	0/1000	2/1000

Table1: this is the result of image with different noise, M means mean, and V means variance of the image

2. Segmentation is not good so that the error is huge.

   This is a good point, so I check the result of my segmentation on the SVHN data set, here is some images

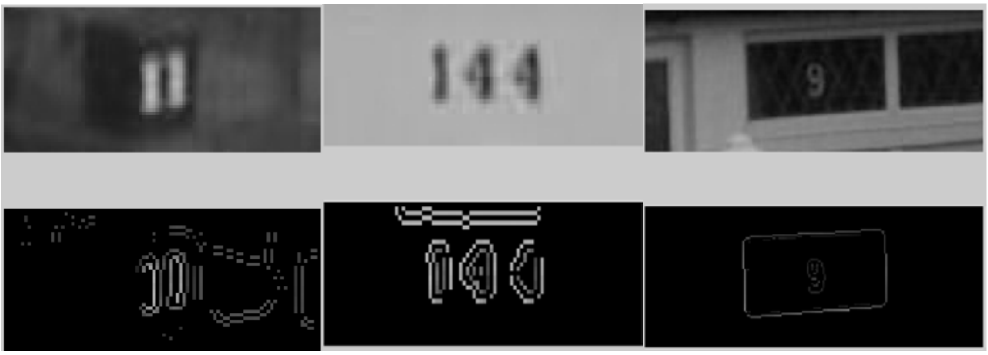


Figure: 6 the result by running seg.m

As we see from this image, the second one and third one is not bad. The number area has been selected, but for the first one, we can see there still a lot noise on the image, if most of the images are like this, that might be a reason why the false positive is very high, because if the cropped images are in these areas, the final recognition will not give back very good results.

3. Orientation error like figure 2

   Here is some sample pictures, these two pictures are both not in the direction that likes Mnist training data, that might be the problem.

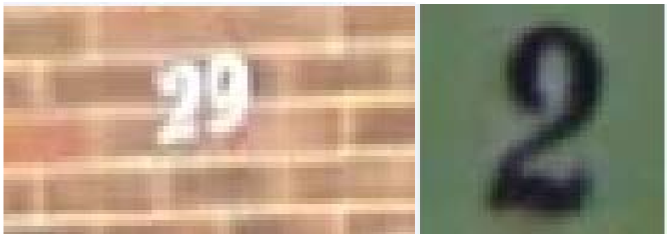


Figure7: these two pictures show that the house number is not oriented very good.

4. Training data incompatible

   We can still look at figure 7, and compare with figure 2, we can see that the door number is printed style, but the Mnist we rely on it is hand written. So that might be a good way we could improve the result too.

Conclusion

After analysis the data and did several experiment to the code data, we could find that Caffe could get a very good result from the clean background hand written digits. We could get a very good score from Caffe trained by Mnist dataset. But for the image like street view house number, Caffe's performance is not very good, several possible reason that might cause such error. Possible reasons are Noise on the background, Segmentation error, and orientation error. Training data is incompatible. In the future we could focus on these areas to do further study.

.

Team role

Shan Cao

References

[1] Casey, B.j., Jay N. Giedd, and Kathleen M. Thomas. "Structural and Functional Brain Development and Its Relation to Cognitive Development." Biological Psychology 54.1-3 (2000): 241-57. Web.

[2] Service, United States Postal. "2001 Comprehensive Statement on Postal Operations." USPS. USPS, 1 Jan. 2011. Web. 4 Dec. 2014.

[3] De Santana Pereira, C.; Cavalcanti, G.D.C., "Handwritten connected digits detection: An approach using instance selection," Image Processing (ICIP), 2011 18th IEEE International Conference on , vol., no., pp.2613,2616, 11-14 Sept. 2011

[4] Goodfellow I, Bulatov Y, Ibarz J, Arnoud S, Shet V. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. [serial online]. December 20, 2013;Available from: arXiv, Ipswich, MA. Accessed December 4, 2014.

[5] Lee, Y, "Handwritten Digit Recognition Using K Nearest-Neighbor, Radial-Basis Function, and Backpropagation Neural Networks," Neural Computation , vol.3, no.3, pp.440,449, Sept. 1991

[6]