ECEN 765 FINAL PROJECT

REPORT

Name: Anrui Liang

UIN: 726007240

Abstract

The project is to use machine learning knowledge learned in ECEN 765 to classify captchas. There are two main process: data collection and fitting model. Captchas are created by myself and there are five progress to simplify common captchas classification. For models, captchas classification problem is a typical image processing problem. So I use K-Nearest Neighbors(KNN) method, Support Vectors Machine(SVM) method, Naive Bayes method and Multilayer Perceptron method to deal with this problem.

Keywords: Captchas KNN MLP

1. Introduction

I saw a news that blind people are confused when they were facing captchas. They can use software to help them read emails and news. However, the software cannot read captchas because captchas are pictures and cannot be detected. In the last semester, I read a paper about this problem and I want to achieve this thing with the knowledge I learned in ECEN 765.

2. Progress

Machine learning program mainly contains two processes --- data collection and building model. For data collection, there are many types of captcha and it is hard to collect them all. So I program by myself to create one common type of captcha with letters, numbers and lines and I will explain a common method to simplify data. For building model, because it is my first time to attend machine learning class, it is hard for me to explore a new algorithm or confirm a theory. So I will use confirmed algorithms such as KNN to solve the problem.

2.1 Data Collection

2.1.1 Introduction

There may be lines, letters and numbers in one captcha. Different regions may use different characters in captcha.(Shown in figure 1)

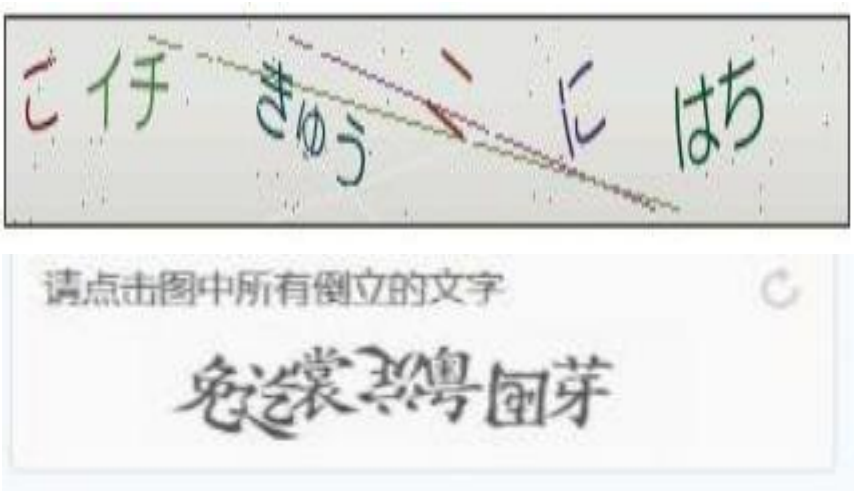

Figure 1. Different characters in captcha

To solve the problem more efficiently, I will choose capital letters and numbers with lines in captcha.

Captcha classification is different from handwriting classification. Captcha picture can be generated by computer easily. It also means that I can program to get a large number of training data and test data in this project.

To increase the complexity of the captcha, I add some lines as noise and change the shape of captcha.

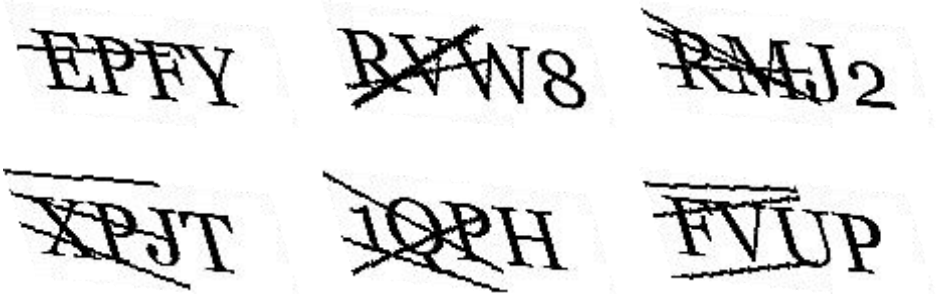

Figure 2. Captcha

2.2.2 Data Processing

Captcha classification is more difficult than hand writing classification. In homework 1, hand writings only contain numbers in every picture but, besides different shape of letters and numbers, captchas contain numbers, lines and letters. It means that captcha classification may have more labels than hand writing classification has in homework 1.

I think data processing in captcha classification should contain five parts. 1) Transfer picture to black white picture. 2) Delete frame. 3) Eliminate point noise. 4) Transfer picture to 0-1 matrix. 5) Divide picture to every digit.

1) Transfer Picture to Black White Picture

Color is useless for us to classify captcha in this project. However, color may increase complexity of project. I use OpenCV to deal with captchas. I use the function cvtColor to change the color and the result is shown as:

Figure 3. Capture with White and Black

And the picture can be also shown as a matrix which only contains 0 (means black pixel) and 255 (means white pixel). And then, the matrix can be used to transfer the picture to 0-1 matrix easily.

```
[[255 255 255 ... 255 255 255]
 [255 255 255 ... 255 255 255]
 [  0 255 255 ... 255 255 255]
 ...
 [255 255 255 ...   0 255 255]
 [255 255 255 ...   0 255 255]
 [255 255 255 ...   0 255 255]]
```

Figure 4. Capture Shown as Matrix

2) Delete Frame and Gain 0-1 Matrix

Because the matrix is gained, I can transfer it to matrix with 0 and 1. Why this part is needed is that matrix with 0 and 1 can be more easily calculated and I can use logic calculation in the program which will be shown in next step. The process of this part is easy. I divide the matrix by 255 and transfer result from floating data to integers data.

Figure 5. Captcha Without Frame

The same as the color, frame is useless for us to classified captchas. So frame should be eliminated. Normally, captcha may have square frame which is easy to eliminate or not have frame. In the project, I generate a captcha without text and change it to 0-1 matrix. And then, I use captcha without frame to xor other captchas. After calculation, the frame can be eliminated.

3) Eliminate Point Noise

Point noise may be generated because of changing colors or original picture. There are many methods to solve this problem and I directly use a filter function to solve this problem.

4) Divide Picture to Digits

I want to divide picture to every digits, so I can solve this problem as MNIST problem and make it easy. For this part, I directly divide the picture which contains 4 digits to 4 parts. Although it is the easiest method, it makes results not good enough because font size is a little bigger than numbers' size.

2.2 Fitting Models

I chose Naive Bayes method and KNN method first, because they are easy to understand. However, I think both methods use too many memories. So i tried other methods such as SVM method. After learning deep learning, I also want to try neural network. So I tried simple multilayer perceptron (MLP) to do classification.

3. Results

3.1 Results

Here is the result of every algorithm.

Method	KNN	SVM	Naive Bayes	MLP
Error Rate(%)	33.3	50.0	40.9	48.6
Memory(MB)	1.195	14.33	5.701	2.586

Table 1. Results of Every Algorithm

3.2 Conclusion

As we can see in table 1, KNN owns the lowest error rate and uses the fewest memory among algorithms. And SVM owns the highest error rate and uses the most memory. Here are my conclusions.

First, the error rate is very high for four algorithms. Even the lowest error rate is bigger than 30%. I think it is because of how I divide captcha. As I mentioned, the letters are a little bigger than numbers but I divided captcha to four parts equally. Some digits may lose a little information. Lost information will be added to another part and become noise. So the result is not good.

Second, we know that deep learning is the most popular to classify pictures. However, MLP in this project does not good. Does that mean deep learning is not good for this problem? I think it is not. In homework 3, we used CNN to deal with cifar-10 problem. Cifar-10 is not a good set but the highest error rate for CNN is only 30%. The reason may be the number of layer is not big because of time.

Finally, SVM use too many memory in this project. I think this problem is because I use Gaussian Kernel and it does not fit in this problem.

4. Summary

Captchas classification is a typical image classification problem. For image classification, there are two main process. One is data collection and another one is fitting model. Data collection is an important point in this project because there are many types of captcha and a common method needs to be find out to deal with all these types. There are too many models for solving image classification problem and we cannot say which one is the best for image classification problem. So I tried most models which I knew to solve this problem. Thank you for your grading this project. And thank you for your class which teaches me about basic machine learning.

5. Reference

[1] Y. Wang, Y. Huang, W. Zheng, Z. Zhou, D. Liu and M. Lu, "Combining convolutional neural network and self-adaptive algorithm to defeat synthetic multi-digit text-based CAPTCHA," *2017 IEEE International Conference on Industrial Technology (ICIT)*, Toronto, ON, 2017, pp. 980-985.
[2] Saikirthiga and S. Vaithyasubramanian, "Review on development of some strong visual CAPTCHAs and breaking of weak audio CAPTCHAs," 2016 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, 2016, pp. 1-4.
[3] Aurelien Geron “Hands-On Machine Learning with Scikit-Learn and TensorFlow” Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.