MNIST-PROJECT REPORT

Zhang Kehan

2019.8.17

1 INTRODUCTION

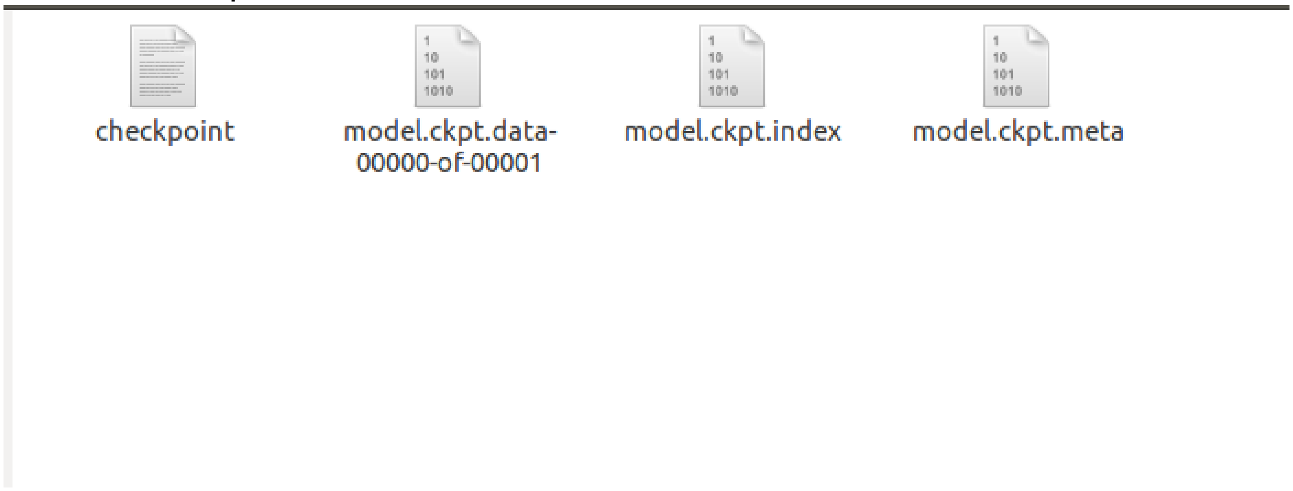
This python program is a project that use the mnist model that have been trained to help users to recognize the images that they upload to the server.

2 REQUIREMENT

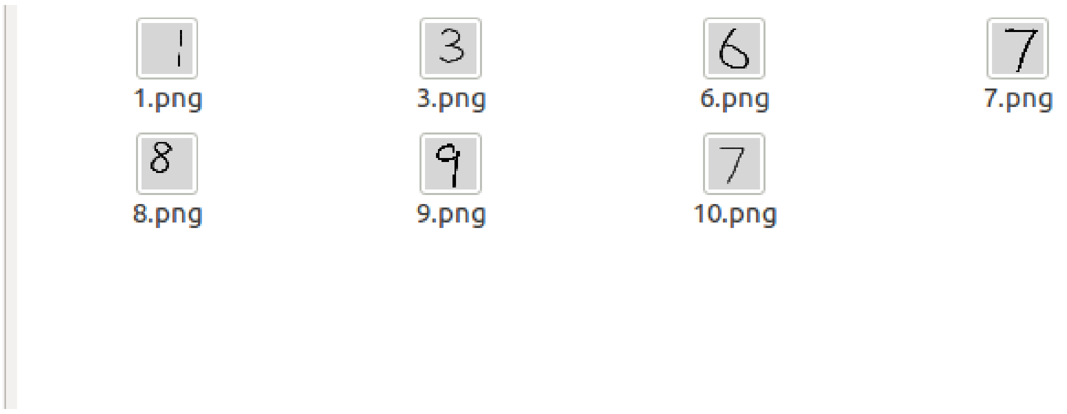
This project have to implement the following requirements:
1)User uploads a image of any format to the server, the server will be able to recognize the digit in the image and return the result of recognization on the HTML page.
2)The program is based on flask, the lite web program framework.
3)The result(including time of recognization, name of uploaded file and the result of recogization) should be recorded in cassandra.
4)The whole main program should be deployed and run in docker and is able to communicate with cassandra.

3 REALIZATION

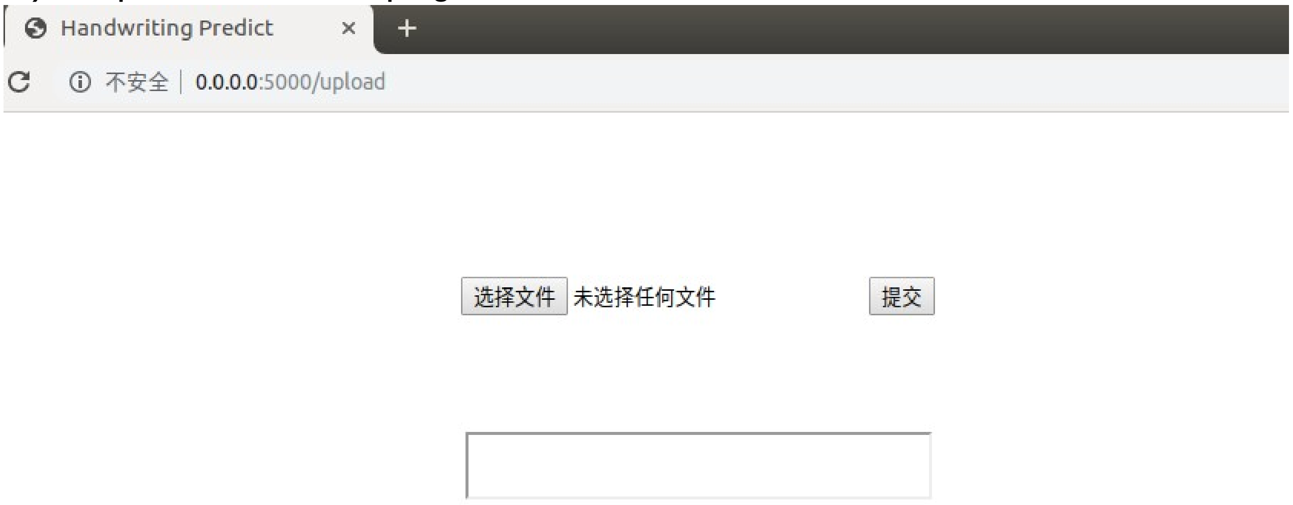
1) train and save model
use 20000 images from MNIST_DATA to train the model.
After access the model, we know that the accuracy of this model is up to 99.2%. saved models are as followed.



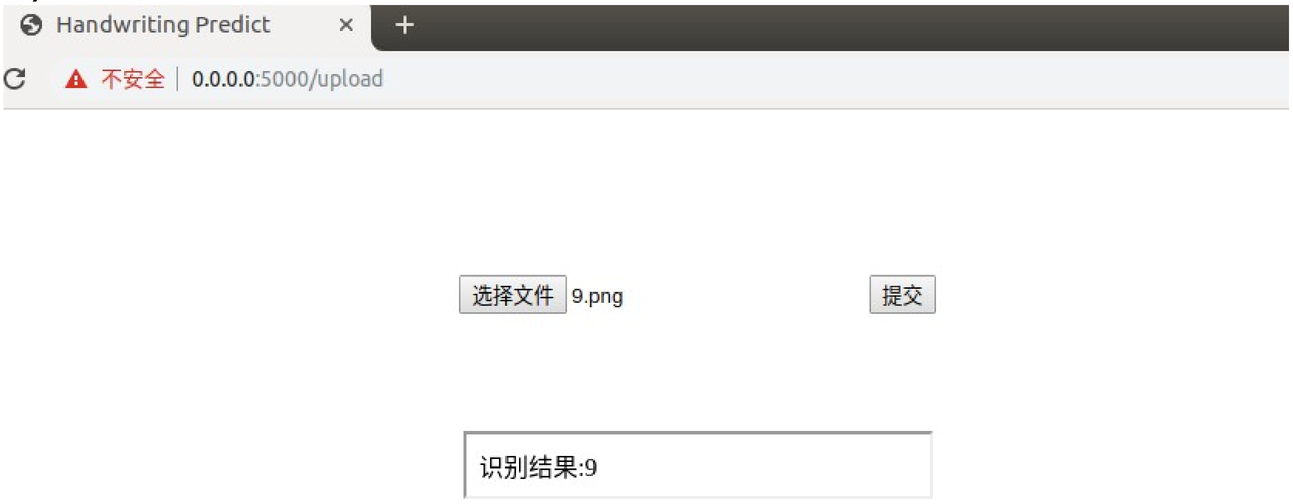
2) manage images
There are several procedures that have to be done to manage the image uploaded by the user. First, I convert RGB image into gray image, and then I convert the image to 28 x 28 pixels. The processed images are as followed.



3) implement web page with FLASK

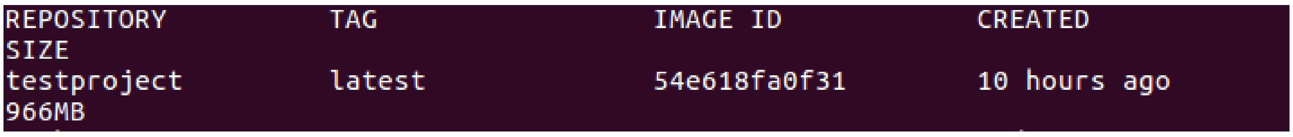


4) use model

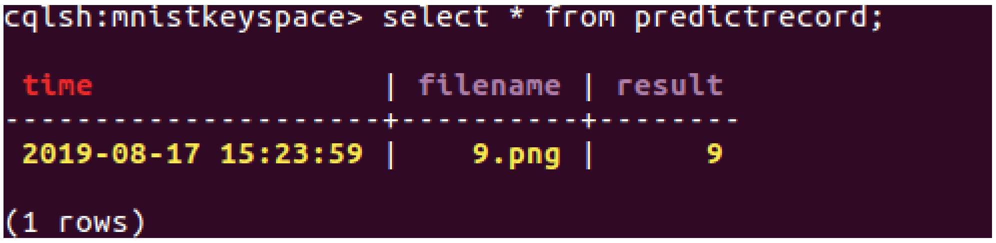


识别结果:9

5) deploy the whole program into container
use the dockerfile to make program images


```
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
testproject         latest              54e618fa0f31        10 hours ago
966MB
```

5) connect to cassandra


```
cqlsh:mnistkeyspace> select * from predictrecord;

 time                | filename | result
---------------------+----------+--------
 2019-08-17 15:23:59 |    9.png |      9

(1 rows)
```

4 PROBLEMS AND SOLUTIONS

During the accomplishment of the project, I encounter several problems and under the help of my teacher ZHANG FAN and blogs on the Internet, I solved them and finished the project successfully. Problems and solutions are as followed.

1) error:Key Variable_*** not found in checkpoint when use the model

solution1:First, I checked if the name of variables are correspond to the name of those that in the saved model

solution2:Then, I find that replace ‘tf.get_variable()’ with ‘tf.Variable()’ can solve the problem. Because when we use tf.Variable(), if the system inspect a conflict in naming, it will handle the conflict automatically.

2) image lack fidelity

solution:sharpening the image using ‘im=ImageEnhance.Sharpness(im).enhance(3)’

3)error:templateNotFound when use flask

solution:create ‘templates’ file folder, and put the html file into the ‘templates’ file folder. Because when execute ’render_template('handwriting_webpage.html')’ , the program will find the html file in the default file folder,which is ‘templates’.

4)problems concerning with communication between containers—unable to connect to any server

solution1: unify the port that is exposed by the container and the port that is run in the program.

solution2: run the two container in the same network

solution3: specify the ip of cassandra and force the program to moniter the ip.

There is a problem impede me for a long time. After we run the cassandra container, we should wait the cassandra container to be fully started for seconds, and then run the program container. Otherwise, two container will not be able to communicate with each other.