Project：Handwriting recognition based on the Mnist

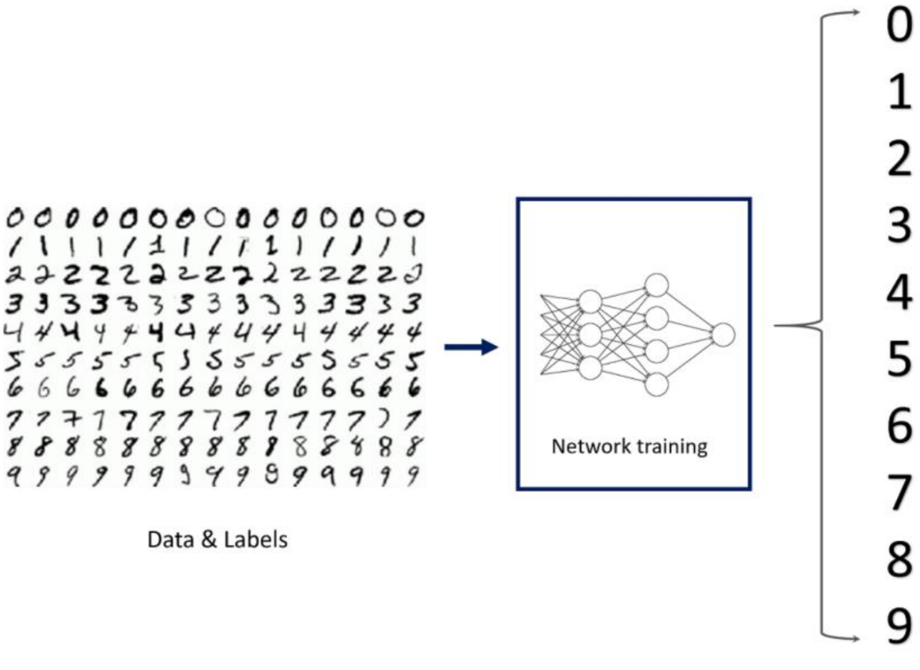
Students：Xi Wang

Teacher：Fan Zhang

Time period：12th July,2019-18th August,2019

1. Research background

(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.It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting.



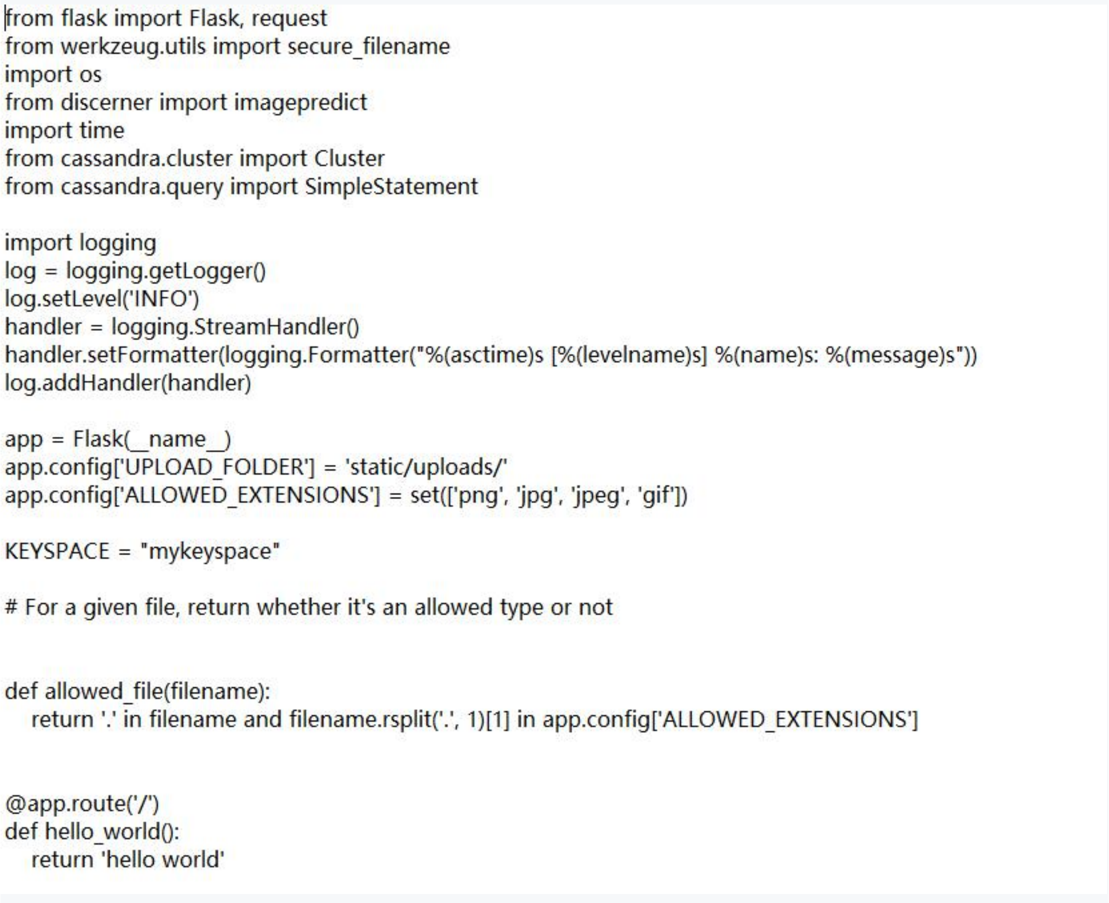
(2):Docker: In the Forrester New Wave ™: Enterprise Container Platform Software Suites, Q4 2018 report, Docker was cited as a leader in enterprise container platform category with Docker and our Docker Enterprise Container platform receiving a “differentiated” rating in eight criteria including runtime and orchestration, security, image management, user experience, vision and more.

2. Knowledge points

(1) .Flask：

1-using the “pip install flask” to install the flask

2-using the code paragraph like the “

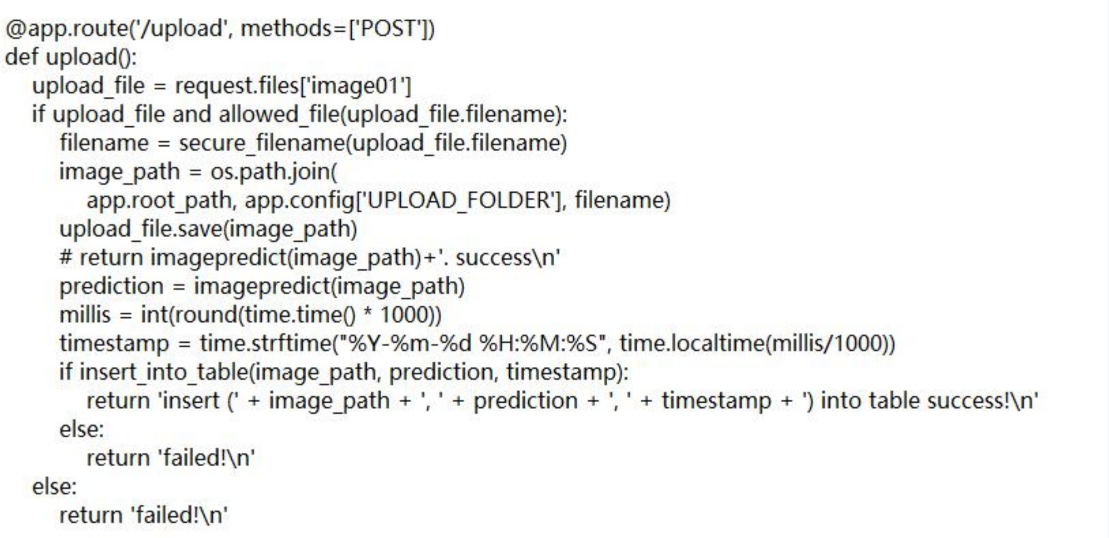


```
from flask import Flask, request
from werkzeug.utils import secure_filename
import os
from discerner import imagepredict
import time
from cassandra.cluster import Cluster
from cassandra.query import SimpleStatement

import logging
log = logging.getLogger()
log.setLevel('INFO')
handler = logging.StreamHandler()
handler.setFormatter(logging.Formatter("%(asctime)s [%(levelname)s] %(name)s: %(message)s"))
log.addHandler(handler)

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = 'static/uploads/'
app.config['ALLOWED_EXTENSIONS'] = set(['png', 'jpg', 'jpeg', 'gif'])

KEYSPACE = "mykeyspace"

# For a given file, return whether it's an allowed type or not


def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1] in app.config['ALLOWED_EXTENSIONS']


@app.route('/')
def hello_world():
    return 'hello world'
```

to create a flask program

3-using the code like the “ `Running on http://127.0.0.1:5000/`”to run a flask.

4-Using the code like the “`curl -F/--form <name>=@<filepath> <url>`” to upload a file,the uploading file paragraph is like the “



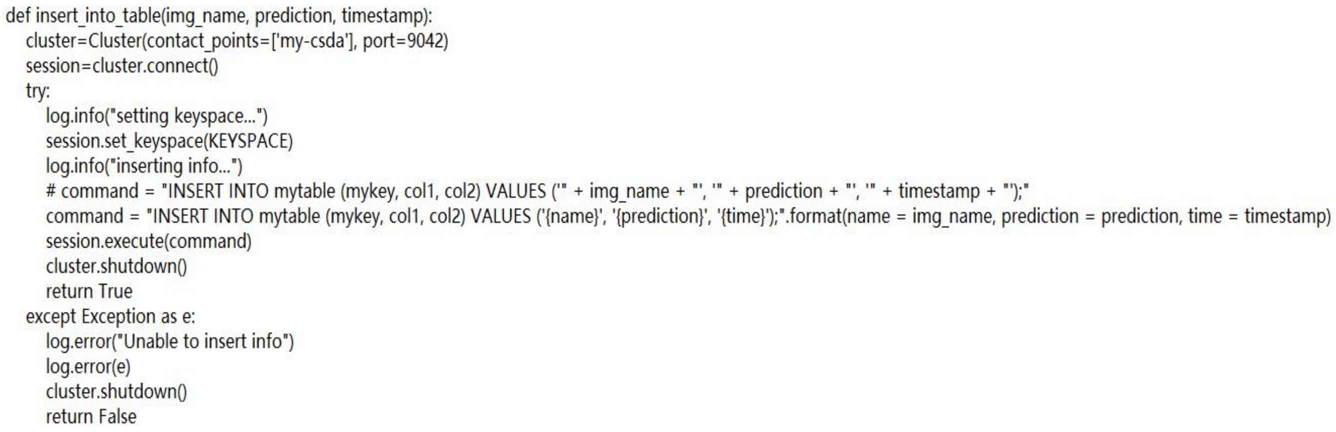
```
@app.route('/upload', methods=['POST'])
def upload():
    upload_file = request.files['image01']
    if upload_file and allowed_file(upload_file.filename):
        filename = secure_filename(upload_file.filename)
        image_path = os.path.join(
            app.root_path, app.config['UPLOAD_FOLDER'], filename)
        upload_file.save(image_path)
        # return imagepredict(image_path)+'. success\n'
        prediction = imagepredict(image_path)
        millis = int(round(time.time() * 1000))
        timestamp = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime(millis/1000))
        if insert_into_table(image_path, prediction, timestamp):
            return 'insert (' + image_path + ', ' + prediction + ', ' + timestamp + ') into table success!\n'
        else:
            return 'failed!\n'
    else:
        return 'failed!\n'
```

(2) Cassandra:

1-It can be inferred from the website:

https://www.w3cschool.cn/cassandra/cassandra_data_model.html

2-using the code like this to insert a keyspace



```
def insert_into_table(img_name, prediction, timestamp):
    cluster=Cluster(contact_points=['my-csda'], port=9042)
    session=cluster.connect()
    try:
        log.info("setting keyspace...")
        session.set_keyspace(KEYSPACE)
        log.info("inserting info...")
        # command = "INSERT INTO mytable (mykey, col1, col2) VALUES ('" + img_name + "', '" + prediction + "', '" + timestamp + "');"
        command = "INSERT INTO mytable (mykey, col1, col2) VALUES ('{name}', '{prediction}', '{time}');".format(name = img_name, prediction = prediction, time = timestamp)
        session.execute(command)
        cluster.shutdown()
        return True
    except Exception as e:
        log.error("Unable to insert info")
        log.error(e)
        cluster.shutdown()
        return False
```

(3) Docker:

3-Building a container:It can be referred from the website :

https://docs.docker.com/get-started/

4-Connecting containers:using the code like

```
″ docker run --link <container_name>:<alias> <img_name>″
```

(4) Mnist:

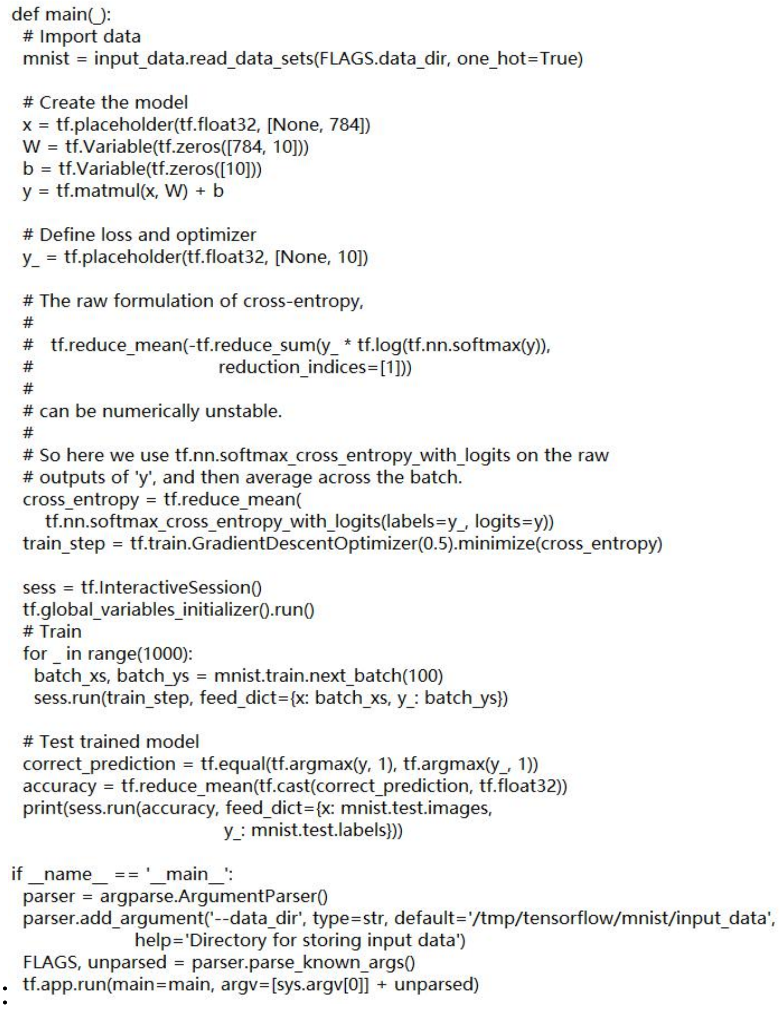
1-It can be referred from the website:

https://blog.csdn.net/u011389706/article/details/81455750

2-using the code like this to download the training collection:

```
"""
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import argparse
import sys

from tensorflow.examples.tutorials.mnist import input_data

import tensorflow as tf

import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

FLAGS = None
```

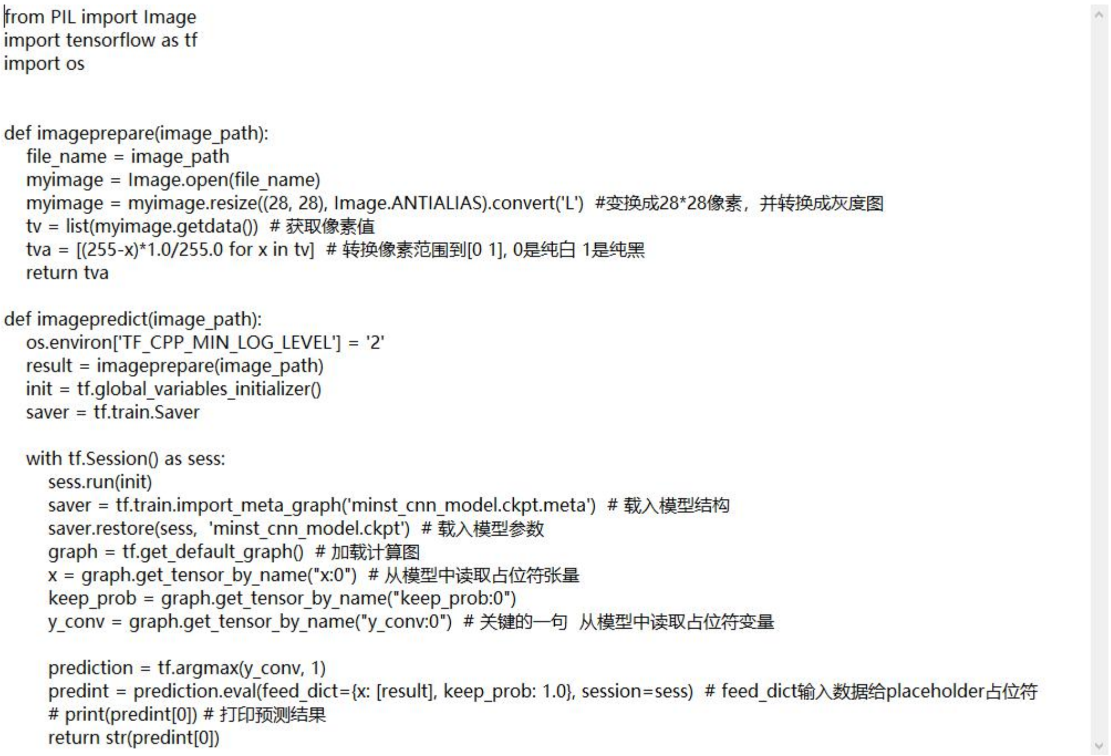
3-Using the code like this to build up your tensorflow



```
def main(_):
  # Import data
  mnist = input_data.read_data_sets(FLAGS.data_dir, one_hot=True)

  # Create the model
  x = tf.placeholder(tf.float32, [None, 784])
  W = tf.Variable(tf.zeros([784, 10]))
  b = tf.Variable(tf.zeros([10]))
  y = tf.matmul(x, W) + b

  # Define loss and optimizer
  y_ = tf.placeholder(tf.float32, [None, 10])

  # The raw formulation of cross-entropy,
  #
  #   tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(tf.nn.softmax(y)),
  #                                 reduction_indices=[1]))
  #
  # can be numerically unstable.
  #
  # So here we use tf.nn.softmax_cross_entropy_with_logits on the raw
  # outputs of 'y', and then average across the batch.
  cross_entropy = tf.reduce_mean(
      tf.nn.softmax_cross_entropy_with_logits(labels=y_, logits=y))
  train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)

  sess = tf.InteractiveSession()
  tf.global_variables_initializer().run()
  # Train
  for _ in range(1000):
    batch_xs, batch_ys = mnist.train.next_batch(100)
    sess.run(train_step, feed_dict={x: batch_xs, y_: batch_ys})

  # Test trained model
  correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_, 1))
  accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
  print(sess.run(accuracy, feed_dict={x: mnist.test.images,
                                      y_: mnist.test.labels}))

if __name__ == '__main__':
  parser = argparse.ArgumentParser()
  parser.add_argument('--data_dir', type=str, default='/tmp/tensorflow/mnist/input_data',
                      help='Directory for storing input data')
  FLAGS, unparsed = parser.parse_known_args()
  tf.app.run(main=main, argv=[sys.argv[0]] + unparsed)
```

model:

4-Using the code like this to assess your photo and save:



```
from PIL import Image
import tensorflow as tf
import os


def imageprepare(image_path):
    file_name = image_path
    myimage = Image.open(file_name)
    myimage = myimage.resize((28, 28), Image.ANTIALIAS).convert('L')  #变换成28*28像素，并转换成灰度图
    tv = list(myimage.getdata())  # 获取像素值
    tva = [(255-x)*1.0/255.0 for x in tv]  # 转换像素范围到[0 1], 0是纯白 1是纯黑
    return tva

def imagepredict(image_path):
    os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
    result = imageprepare(image_path)
    init = tf.global_variables_initializer()
    saver = tf.train.Saver

    with tf.Session() as sess:
        sess.run(init)
        saver = tf.train.import_meta_graph('minst_cnn_model.ckpt.meta')  # 载入模型结构
        saver.restore(sess,  'minst_cnn_model.ckpt')  # 载入模型参数
        graph = tf.get_default_graph()  # 加载计算图
        x = graph.get_tensor_by_name("x:0")  # 从模型中读取占位符张量
        keep_prob = graph.get_tensor_by_name("keep_prob:0")
        y_conv = graph.get_tensor_by_name("y_conv:0")  # 关键的一句  从模型中读取占位符变量

        prediction = tf.argmax(y_conv, 1)
        predint = prediction.eval(feed_dict={x: [result], keep_prob: 1.0}, session=sess)  # feed_dict输入数据给placeholder占位符
        # print(predint[0]) # 打印预测结果
        return str(predint[0])
```

3. Process

Firstly,we need to use “pip install *** ” to download the package we need like the pillow,flask,tensorflow and cassandra-driver and so on.Then,we should try to run the mnist traing code to download the traing collection from the official website.As we have successfully run the program,we could try to build the Image of the Mnist and flask.After that,we need to pull a Cassandra Image and create a network.After that,try to build a Cassandra container and then use the connecting code to link the Cassandra container with the Mnist and flask container.As these have been down,we could try to upload a picture and then check the answer in the containers.

4. Problems and solutions

1-Because my docker was installed under the virtualbox,so I was unable to use the flask with the url which the program showed like the "0.0.0.0:Port Number" .In that case,I have been in trouble for a long time.However,I found that I need to enter to set the virtualbox' s Port translation.The host IP could be set as the IP of the "VirtualBox Host-Only Network" ,and the IP of the subnet should be set as the IP of the what is shown at the Docker program' s beginning line.And the host and the subnet Port should be set as the port which I exposed.After that,I successfully realized the website' s interaction with flask,while the url should be the "subnet IP:Port Number" .

2-As I built the container of the flask and mnist first,I could not realize the connection of it with the cassandra container.In order to solve this,I have to build the cassandra container first,and then use the code like the "`docker run -p <port>:<port> --network <network>--link <container_name>:<alias> <img_name>`" to realize the connection and build the container of the flask and the mnist because they are under the same network now,which would allow their connection.

3-Because I am in China,so I need to use the VPN to download the package,which is so slow and not readily.Because of this,I changed the downloading source with the code "`RUN pip install -i https://pypi.tuna.tsinghua.edu.cn/simple -r requirements.tx`".And it became efficient soon.

5. My impressions

3-From this project,I have just tried to contact with the Big data and the technique of Docker.I have felt the advantages of containers rather than the virtual system because I could start several containers at the same time,while the virtual system have much higher requirements of the CPU.I also started to learn to find my problems and find the solution on my own,by searching the Internet and do some bold attempts.