**软件名称strdsty源程序**

训练数据获取

from scipy.io import loadmat

import glob

import cv2

import os

from sklearn.neighbors import NearestNeighbors

import numpy as np

import matplotlib.pyplot as plt

# 生成密度图所使用高斯核大小

Gauss\_ksize = 3

class MallDataset(object):

def \_\_init\_\_(self):

self.filenames = sorted(glob.glob('../data/mall\_dataset/frames/\*.jpg'), key=lambda x: int(x[-8:-4]))

def get\_train\_num(self):

return int(len(self.filenames) \* 0.8)

def get\_valid\_num(self):

return len(self.filenames) - int(len(self.filenames) \* 0.8)

def get\_annotation(self):

mat\_annotation = loadmat('../data/mall\_dataset/mall\_gt.mat')

count\_data, position\_data = mat\_annotation['count'], mat\_annotation['frame'][0]

return count\_data, position\_data

def get\_pixels(self, img, img\_index, positions, size):

h, w = img.shape[:-1]

proportion\_h, proportion\_w = size / h, size / w # 输入层需求与当前图片大小对比

pixels = np.zeros((size, size))

for p in positions[img\_index][0][0][0]:

# 取出每个人的坐标

now\_x, now\_y = int(p[0] \* proportion\_w), int(p[1] \* proportion\_h) # 按照输入层要求调整坐标位置

if now\_x >= size or now\_y >= size:

# 越界则向下取整

print("Sorry skip the point, its index of all is {}".format(img\_index))

else:

pixels[now\_y, now\_x] += 1

pixels = cv2.GaussianBlur(pixels, (15, 15), 0)

return pixels

def get\_img\_data(self, index, size):

\_, positions = self.get\_annotation()

img = cv2.imread(self.filenames[index])

density\_map = np.expand\_dims(self.get\_pixels(img, index, positions, size // 4), axis=-1)

img = cv2.resize(img, (size, size)) / 255.

return img, density\_map

def gen\_train(self, batch\_size, size):

\_, position = self.get\_annotation()

index\_all = list(range(int(len(self.filenames) \* 0.8))) # 取出所有训练数据下标，默认数据的前80%为训练集

i, n = 0, len(index\_all)

if batch\_size > n:

raise Exception('Batch size {} is larger than the number of dataset {}!'.format(batch\_size, n))

while True:

if i + batch\_size >= n:

np.random.shuffle(index\_all)

i = 0

continue

batch\_x, batch\_y = [], []

for j in range(i, i + batch\_size):

x, y = self.get\_img\_data(index\_all[j], size)

batch\_x.append(x)

batch\_y.append(y)

i += batch\_size

yield np.array(batch\_x), np.array(batch\_y)

def gen\_valid(self, batch\_size, size):

\_, position = self.get\_annotation()

index\_all = list(range(int(len(self.filenames) \* 0.8), len(self.filenames)))

i, n = 0, len(index\_all)

if batch\_size > n:

raise Exception('Batch size {} is larger than the number of dataset {}!'.format(batch\_size, n))

while True:

if i + batch\_size >= n:

np.random.shuffle(index\_all)

i = 0

continue

batch\_x, batch\_y = [], []

for j in range(i, i + batch\_size):

x, y = self.get\_img\_data(index\_all[j], size)

batch\_x.append(x)

batch\_y.append(y)

i += batch\_size

yield np.array(batch\_x), np.array(batch\_y)

def gen\_all(self, pic\_size):

x\_data = []

y\_data = []

for i in range(len(self.filenames)):

image, map\_ = self.get\_img\_data(i, pic\_size)

x\_data.append(image)

y\_data.append(map\_)

x\_data, y\_data = np.array(x\_data), np.array(y\_data)

return x\_data, y\_data

class ShanghaitechDataset(object):

def \_\_init\_\_(self, part='A'):

if part == 'A':

self.folder = '../data/ShanghaiTech/part\_A/'

else:

self.folder = '../data/ShanghaiTech/part\_B/'

def get\_annotation(self, folder, index):

mat\_data = loadmat(folder + 'ground\_truth/GT\_IMG\_{}.mat'.format(index))

positions, count = mat\_data['image\_info'][0][0][0][0][0], mat\_data['image\_info'][0][0][0][0][1][0][0]

return positions, count

def get\_pixels(self, folder, img, img\_index, size):

positions, \_ = self.get\_annotation(folder, img\_index)

h, w = img.shape[0], img.shape[1]

proportion\_h, proportion\_w = size / h, size / w # 输入层需求与当前图片大小对比

pixels = np.zeros((size, size))

for p in positions:

# 取出每个人的坐标

now\_x, now\_y = int(p[0] \* proportion\_w), int(p[1] \* proportion\_h) # 按照输入层要求调整坐标位置

if now\_x >= size or now\_y >= size:

# 越界则向下取整

pass

# print("Sorry skip the point, image index of all is {}".format(img\_index))

else:

pixels[now\_y, now\_x] += 1

pixels = cv2.GaussianBlur(pixels, (15, 15), 0)

return pixels

def gen\_train(self, batch\_size, size):

folder = self.folder + 'train\_data/'

index\_all = [i+1 for i in range(len(glob.glob(folder + 'images/\*.jpg')))]

i, n = 0, len(index\_all)

if batch\_size > n:

raise Exception('Batch size {} is larger than the number of dataset {}!'.format(batch\_size, n))

while True:

if i + batch\_size >= n:

np.random.shuffle(index\_all)

i = 0

continue

batch\_x, batch\_y = [], []

for j in range(i, i + batch\_size):

img = cv2.imread(folder + 'images/IMG\_{}.jpg'.format(index\_all[j]))

density = np.expand\_dims(self.get\_pixels(folder, img, index\_all[j], size // 4), axis=-1)

img = cv2.resize(img, (size, size)) / 255.

batch\_x.append(img)

batch\_y.append(density)

i += batch\_size

yield np.array(batch\_x), np.array(batch\_y)

def gen\_valid(self, batch\_size, size):

folder = self.folder + 'test\_data/'

index\_all = [i + 1 for i in range(len(glob.glob(folder + 'images/\*.jpg')))]

i, n = 0, len(index\_all)

if batch\_size > n:

raise Exception('Batch size {} is larger than the number of dataset {}!'.format(batch\_size, n))

while True:

if i + batch\_size >= n:

np.random.shuffle(index\_all)

i = 0

continue

batch\_x, batch\_y = [], []

for j in range(i, i + batch\_size):

img = cv2.imread(folder + 'images/IMG\_{}.jpg'.format(index\_all[j]))

density = np.expand\_dims(self.get\_pixels(folder, img, index\_all[j], size // 4), axis=-1)

img = cv2.resize(img, (size, size)) / 255.

batch\_x.append(img)

batch\_y.append(density)

i += batch\_size

yield np.array(batch\_x), np.array(batch\_y)

def get\_train\_num(self):

return len(glob.glob(self.folder + 'train\_data/' + 'images/\*'))

def get\_valid\_num(self):

return len(glob.glob(self.folder + 'test\_data/' + 'images/\*'))

class CrowDataset(object):

def \_\_init\_\_(self):

mat\_annotation = loadmat(os.path.join('../data/Crowdata', 'crow.mat'))

self.img\_dir = '../data/Crowdata/img'

self.filenames = mat\_annotation['img']

self.counts = mat\_annotation['count']

self.positions = mat\_annotation['points']

def get\_train\_num(self):

return int(len(self.filenames) \* 0.8)

def get\_valid\_num(self):

return len(self.filenames) - int(len(self.filenames) \* 0.8)

def get\_densemap(self, img, img\_index, size):

h, w = img.shape[:-1]

proportion\_h, proportion\_w = size / h, size / w # 输入层需求与当前图片大小对比

gts = self.positions[0][img\_index].copy()

for i, p in enumerate(self.positions[0][img\_index]):

# 取出每个人的坐标

now\_x, now\_y = int(p[0] \* proportion\_w), int(p[1] \* proportion\_h) # 按照输入层要求调整坐标位置

gts[i][0] = now\_x

gts[i][1] = now\_y

res = np.zeros(shape=[size, size])

bool\_res = (gts[:, 0] < size) & (gts[:, 1] < size)

for k in range(len(gts)):

gt = gts[k]

if bool\_res[k] == True:

res[int(gt[1])][int(gt[0])] = 1

pts = np.array(list(zip(np.nonzero(res)[1], np.nonzero(res)[0])))

map\_shape = [size, size]

density = np.zeros(shape=map\_shape, dtype=np.float32)

if len(pts) == 1:

sigmas = [0]

else:

neighbors = NearestNeighbors(n\_neighbors=1, algorithm='kd\_tree', leaf\_size=1200)

neighbors.fit(pts.copy())

distances, \_ = neighbors.kneighbors()

sigmas = distances.sum(axis=1) \* 0.3

for i in range(len(pts)):

pt = pts[i]

pt2d = np.zeros(shape=map\_shape, dtype=np.float32)

pt2d[pt[1]][pt[0]] = 1

density += cv2.GaussianBlur(pt2d, (Gauss\_ksize, Gauss\_ksize), sigmas[i])

# density += filters.gaussian\_filter(pt2d, sigmas[i], mode='constant')

return density

def get\_img\_data(self, index, size):

img = cv2.imread(os.path.join(self.img\_dir, self.filenames[index]))

density\_map = np.expand\_dims(self.get\_densemap(img, index, size // 4), axis=-1)

img = cv2.resize(img, (size, size)) / 255.

return img, density\_map

def gen\_train(self, batch\_size, size):

index\_all = list(range(int(len(self.filenames) \* 0.8))) # 取出所有训练数据下标，默认数据的前80%为训练集

i, n = 0, len(index\_all)

if batch\_size > n:

raise Exception('Batch size {} is larger than the number of dataset {}!'.format(batch\_size, n))

while True:

if i + batch\_size >= n:

np.random.shuffle(index\_all)

i = 0

continue

batch\_x, batch\_y = [], []

for j in range(i, i + batch\_size):

x, y = self.get\_img\_data(index\_all[j], size)

batch\_x.append(x)

batch\_y.append(y)

i += batch\_size

yield np.array(batch\_x), np.array(batch\_y)

def gen\_valid(self, batch\_size, size):

index\_all = list(range(int(len(self.filenames) \* 0.8), len(self.filenames)))

i, n = 0, len(index\_all)

if batch\_size > n:

raise Exception('Batch size {} is larger than the number of dataset {}!'.format(batch\_size, n))

while True:

if i + batch\_size >= n:

np.random.shuffle(index\_all)

i = 0

continue

batch\_x, batch\_y = [], []

for j in range(i, i + batch\_size):

x, y = self.get\_img\_data(index\_all[j], size)

batch\_x.append(x)

batch\_y.append(y)

i += batch\_size

yield np.array(batch\_x), np.array(batch\_y)

def gen\_all(self, pic\_size):

x\_data = []

y\_data = []

for i in range(len(self.filenames)):

image, map\_ = self.get\_img\_data(i, pic\_size)

x\_data.append(image)

y\_data.append(map\_)

x\_data, y\_data = np.array(x\_data), np.array(y\_data)

return x\_data, y\_data

def show\_data(self):

import random

index = [i for i in range(100)]

random.shuffle(index)

index = index[0:5]

maps = []

raw\_images = []

true\_counts = []

counts = []

for i in index:

im = cv2.imread(os.path.join(self.img\_dir, self.filenames[i]))

raw\_images.append(cv2.cvtColor(im, cv2.COLOR\_BGR2RGB))

pixel = self.get\_densemap(im, i, 224)

maps.append(pixel)

counts.append(int(np.sum(pixel)))

true\_counts.append(self.counts[0][i])

plt.figure(figsize=(15, 9))

for i in range(len(maps)):

plt.subplot(2, 5, i + 1)

plt.imshow(raw\_images[i])

plt.title('people true num {}'.format(int(true\_counts[i])))

plt.subplot(2, 5, i + 1 + 5)

plt.imshow(maps[i])

plt.title('people pred num {}'.format(counts[i]))

plt.show()

if \_\_name\_\_ == '\_\_main\_\_':

MallDataset().gen\_valid(16, 224)

神经网络模型

from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, concatenate, Activation, Dense, BatchNormalization

from tensorflow.keras.models import Model

from tensorflow.keras.regularizers import l2

def MSB(filter\_num):

def f(x):

params = {

'strides': 1,

'activation': 'relu',

'padding': 'same',

'kernel\_regularizer': l2(5e-4)

}

x1 = Conv2D(filters=filter\_num, kernel\_size=(9, 9), \*\*params)(x)

x2 = Conv2D(filters=filter\_num, kernel\_size=(7, 7), \*\*params)(x)

x3 = Conv2D(filters=filter\_num, kernel\_size=(5, 5), \*\*params)(x)

x4 = Conv2D(filters=filter\_num, kernel\_size=(3, 3), \*\*params)(x)

x = concatenate([x1, x2, x3, x4])

x = BatchNormalization()(x)

return x

return f

def MSB\_mini(filter\_num):

def f(x):

params = {

'strides': 1,

'activation': 'relu',

'padding': 'same',

'kernel\_regularizer': l2(5e-4)

}

x2 = Conv2D(filters=filter\_num, kernel\_size=(7, 7), \*\*params)(x)

x3 = Conv2D(filters=filter\_num, kernel\_size=(5, 5), \*\*params)(x)

x4 = Conv2D(filters=filter\_num, kernel\_size=(3, 3), \*\*params)(x)

x = concatenate([x2, x3, x4])

x = BatchNormalization()(x)

x = Activation('relu')(x)

return x

return f

def MSCNN(input\_shape=(224, 224, 3)):

input\_tensor = Input(shape=input\_shape)

# block1

x = Conv2D(filters=64, kernel\_size=(9, 9), strides=1, padding='same', activation='relu')(input\_tensor)

# block2

x = MSB(4\*16)(x)

x = MaxPooling2D(pool\_size=(2, 2), strides=(2, 2))(x)

x = MSB(4\*32)(x)

x = MSB(4\*32)(x)

x = MaxPooling2D(pool\_size=(2, 2), strides=(2, 2))(x)

x = MSB\_mini(3\*64)(x)

x = MSB\_mini(3\*64)(x)

x = Conv2D(1000, (1, 1), activation='relu', kernel\_regularizer=l2(5e-4))(x)

x = Conv2D(1, (1, 1))(x)

x = Activation('sigmoid')(x)

x = Activation('relu')(x)

model = Model(inputs=input\_tensor, outputs=x)

return model

if \_\_name\_\_ == '\_\_main\_\_':

mscnn = MSCNN((224, 224, 3))

print(mscnn.summary())

#数据计算部分

import os

# os.environ["CUDA\_VISIBLE\_DEVICES"] = "-1"

os.environ['TF\_CPP\_MIN\_LOG\_LEVEL'] = '2'

import os

import cv2

import numpy as np

import time

import datetime

from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, concatenate, Activation, BatchNormalization

from tensorflow.keras.models import Model

from tensorflow.keras.regularizers import l2

import requests

URL = 'http://101.132.74.147:8081/data/setDataList'

def fdd():

model = MSCNN((224, 224, 3))

model.load\_weights('../models/model\_weights.h5')

# 当前六个设备

devices\_name = ['Surveillance\_camera1', 'Surveillance\_camera2', 'Surveillance\_camera3', 'Surveillance\_camera4',

'Surveillance\_camera5', 'Surveillance\_camera6', 'Surveillance\_camera7', 'Surveillance\_camera8',

'Surveillance\_camera9', 'Surveillance\_camera10']

gc = []

# 设备初始化

for i in range(len(devices\_name)):

gc.append(Get\_Capture(devices\_name[i]))

nowMinute = datetime.datetime.now().minute

while True:

# 对比当前时间 若时间更新则进行数据更新

if datetime.datetime.now().minute != nowMinute:

# 延时5s 防止网络问题导致图片上传不成功

time.sleep(5)

nowMinute = datetime.datetime.now().minute

print('-------------------------------')

print(nowMinute)

# 待发送数据

sendData = []

for i in range(len(devices\_name)):

# 加载图片

img = gc[i].loadImg(datetime.datetime.now().minute)

if img is not None:

# 图像预处理

img = cv2.resize(img, (224, 224)) / 255.

img = np.expand\_dims(img, axis=0)

# 预测人数

dmap = np.squeeze(model.predict(img), axis=-1) # 降维

dmap = cv2.GaussianBlur(dmap, (15, 15), 0) # 高斯模糊

gc[i].num = int(np.sum(dmap))

else:

gc[i].num = -1

# 打包数据

sendData.append(jsonData(gc[i].id, gc[i].latitude, gc[i].longitude, gc[i].num).\_\_dict\_\_)

print(sendData)

res = requests.post(url=URL, json=sendData)

print(res)

time.sleep(1)

class Get\_Capture:

def \_\_init\_\_(self, dirname):

self.folder = '../data/'

self.dirname = dirname

self.Image = None

self.num = -1

# 读取位置信息

f = open(self.folder + self.dirname + "/Location.txt", "r")

Str\_ = f.read().split()

self.latitude = float(Str\_[0])

self.longitude = float(Str\_[1])

self.id = Str\_[2]

f.close()

def loadImg(self, index):

if os.path.exists(self.folder + self.dirname + '/Capture ({}).jpg'.format(index)):

self.Image = cv2.imread(self.folder + self.dirname + '/Capture ({}).jpg'.format(index))

return self.Image

else:

return None

class jsonData:

def \_\_init\_\_(self, id, latitude, longitude, num):

self.cameraId = id

self.latitude = latitude

self.longitude = longitude

self.num = num

def MSB(filter\_num):

def f(x):

params = {

'strides': 1,

'activation': 'relu',

'padding': 'same',

'kernel\_regularizer': l2(5e-4)

}

x1 = Conv2D(filters=filter\_num, kernel\_size=(9, 9), \*\*params)(x)

x2 = Conv2D(filters=filter\_num, kernel\_size=(7, 7), \*\*params)(x)

x3 = Conv2D(filters=filter\_num, kernel\_size=(5, 5), \*\*params)(x)

x4 = Conv2D(filters=filter\_num, kernel\_size=(3, 3), \*\*params)(x)

x = concatenate([x1, x2, x3, x4])

x = BatchNormalization()(x)

return x

return f

def MSB\_mini(filter\_num):

def f(x):

params = {

'strides': 1,

'activation': 'relu',

'padding': 'same',

'kernel\_regularizer': l2(5e-4)

}

x2 = Conv2D(filters=filter\_num, kernel\_size=(7, 7), \*\*params)(x)

x3 = Conv2D(filters=filter\_num, kernel\_size=(5, 5), \*\*params)(x)

x4 = Conv2D(filters=filter\_num, kernel\_size=(3, 3), \*\*params)(x)

x = concatenate([x2, x3, x4])

x = BatchNormalization()(x)

x = Activation('relu')(x)

return x

return f

def MSCNN(input\_shape=(224, 224, 3)):

# 构建模型

input\_tensor = Input(shape=input\_shape)

# block1

x = Conv2D(filters=64, kernel\_size=(9, 9), strides=1, padding='same', activation='relu')(input\_tensor)

# block2

x = MSB(4 \* 16)(x)

x = MaxPooling2D(pool\_size=(2, 2), strides=(2, 2))(x)

# block3

x = MSB(4 \* 32)(x)

x = MSB(4 \* 32)(x)

x = MaxPooling2D(pool\_size=(2, 2), strides=(2, 2))(x)

x = MSB\_mini(3 \* 64)(x)

x = MSB\_mini(3 \* 64)(x)

x = Conv2D(1000, (1, 1), activation='relu', kernel\_regularizer=l2(5e-4))(x)

x = Conv2D(1, (1, 1))(x)

x = Activation('sigmoid')(x)

x = Activation('relu')(x)

model = Model(inputs=input\_tensor, outputs=x)

return model

if \_\_name\_\_ == '\_\_main\_\_':

fdd()

测试模块

import os

os.environ['TF\_CPP\_MIN\_LOG\_LEVEL'] = '2'

from argparse import ArgumentParser

import matplotlib.pyplot as plt

import os

import random

import glob

import cv2

import numpy as np

from model import MSCNN

import scipy.io as sio

def parse\_params():

ap = ArgumentParser()

ap.add\_argument('-s', '--show', default='yes', help='if show test result map')

ap.add\_argument('-d', '--dataset', default='malldataset', help='')

args\_ = ap.parse\_args()

args\_ = vars(args\_)

return args\_

def get\_samples\_malldataset(num):

def get\_annotation():

mat\_annotation = sio.loadmat('../data/mall\_dataset/mall\_gt.mat')

count\_data, position\_data = mat\_annotation['count'], mat\_annotation['frame'][0]

return count\_data, position\_data

counts\_true, \_ = get\_annotation()

datasize = len(glob.glob('../data/mall\_dataset/frames/\*.jpg'))

samples\_index = random.sample([i for i in range(int(0.8 \* datasize), datasize)], num)

samples = [glob.glob('../data/mall\_dataset/frames/\*.jpg')[i] for i in samples\_index]

images = []

counts = []

for i in range(num):

filename = samples[i]

print(filename)

print(samples\_index[i])

img = cv2.resize(cv2.imread(filename), (224, 224)) / 255.

img = np.expand\_dims(img, axis=0)

images.append(img)

counts.append(counts\_true[samples\_index[i]])

return images, counts

def get\_samples\_shanghaitech(num):

def get\_annotation(index):

mat\_data = sio.loadmat('../data/ShanghaiTech/part\_A/test\_data/ground\_truth/GT\_IMG\_{}.mat'.format(index))

position\_data, count\_data = mat\_data['image\_info'][0][0][0][0][0], mat\_data['image\_info'][0][0][0][0][1][0][0]

return count\_data, position\_data

datasize = len(glob.glob('../data/ShanghaiTech/part\_A/test\_data/images/\*.jpg'))

# 从验证集选取测试图片

samples\_index = random.sample([i for i in range(datasize)], num)

samples = ['../data/ShanghaiTech/part\_A/test\_data/images/IMG\_{}.jpg'.format(i) for i in samples\_index]

images = []

counts = []

for i in range(num):

filename = samples[i]

img = cv2.resize(cv2.imread(filename), (224, 224)) / 255.

img = np.expand\_dims(img, axis=0)

images.append(img)

counts.append(get\_annotation(samples\_index[i])[0])

return images, counts

def get\_samples\_crowdataset(num):

import random

mat\_annotation = sio.loadmat(os.path.join('../data/Crowdata', 'crow.mat'))

counts\_true, position\_data = mat\_annotation['count'], mat\_annotation['points']

filenames = mat\_annotation['img']

samples\_index = [i for i in range(len(filenames))]

random.shuffle(samples\_index)

samples\_index = samples\_index[0:num]

images = []

counts = []

for index in samples\_index:

filename = os.path.join('../data/Crowdata', 'img', filenames[index])

img = cv2.resize(cv2.imread(filename), (224, 224)) / 255.

img = np.expand\_dims(img, axis=0)

images.append(img)

counts.append(counts\_true[0][index])

return images, counts

def plot\_sample(raw\_images, maps, counts, true\_counts):

plt.figure(figsize=(30, 10))

for i in range(len(maps)):

plt.subplot(2, 10, i + 1)

plt.imshow(np.squeeze(raw\_images[i], axis=0))

plt.title('people true num {}'.format(int(true\_counts[i])))

plt.subplot(2, 10, i + 1 + 10)

plt.imshow(maps[i][0])

plt.title('people pred num {}'.format(counts[i]))

plt.savefig('../results/rst.png')

def save\_result(raw\_images, maps, counts, args\_, true\_counts):

if not os.path.exists('../results'):

os.mkdir('../results')

if args\_['show'] == 'yes':

plot\_sample(raw\_images, maps, counts, true\_counts)

def test(args\_):

model = MSCNN((224, 224, 3))

if os.path.exists('../models/best\_model\_weights.h5'):

model.load\_weights('../models/best\_model\_weights.h5')

if args\_['dataset'] == 'malldataset':

samples, true\_counts = get\_samples\_malldataset(10)

elif args\_['dataset'] == 'Crowdataset':

samples, true\_counts = get\_samples\_crowdataset(10)

else:

samples, true\_counts = get\_samples\_shanghaitech(10)

maps = []

counts = []

for sample in samples:

dmap = np.squeeze(model.predict(sample), axis=-1)

dmap = cv2.GaussianBlur(dmap, (15, 15), 0)

counts.append(int(np.sum(dmap)))

maps.append(dmap)

save\_result(samples, maps, counts, args\_, true\_counts)

else:

print("Sorry, cannot find model file in root\_path/models/, please download my model or train your model")

if \_\_name\_\_ == '\_\_main\_\_':

args = parse\_params()

test(args)

训练模块

import os

os.environ['TF\_CPP\_MIN\_LOG\_LEVEL'] = '2'

import warnings

warnings.filterwarnings('ignore')

from argparse import ArgumentParser

from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping, ModelCheckpoint

from tensorflow.keras.optimizers import Adam, SGD

from model import MSCNN

from data import MallDataset, ShanghaitechDataset, CrowDataset

import tensorflow as tf

if tf.config.list\_physical\_devices('GPU'):

print("use gpu 0")

else:

print("no gpu")

def parse\_command\_params():

parser = ArgumentParser()

parser.add\_argument('-e', '--epochs', default=5, help='how many epochs to fit')

parser.add\_argument('-v', '--show', default='yes', help='if show training log')

parser.add\_argument('-b', '--batch', default=8, help='batch size of train')

parser.add\_argument('-d', '--dataset', default='malldataset', help='which dataset to train')

parser.add\_argument('-p', '--pretrained', default='no', help='load your pretrained model in folder root/models')

args\_ = parser.parse\_args()

args\_ = vars(args\_)

return args\_

def get\_callbacks():

early\_stopping = EarlyStopping(monitor='val\_loss', patience=20)

reduce\_lr = ReduceLROnPlateau(monitor='loss', factor=0.0005, patience=5, min\_lr=1e-7, verbose=True)

if not os.path.exists('../models'):

os.mkdir('../models')

model\_checkpoint = ModelCheckpoint('../models/best\_model\_weights.h5', monitor='val\_loss',

verbose=True, save\_best\_only=True, save\_weights\_only=True)

callbacks = [early\_stopping, reduce\_lr, model\_checkpoint]

return callbacks

def train(args\_):

model = MSCNN((224, 224, 3))

model.compile(optimizer=SGD(lr=3e-4, momentum=0.9), loss='mse')

if args\_['pretrained'] == 'yes':

model.load\_weights('../models/best\_model\_weights.h5')

print("load model from ../models/")

callbacks = get\_callbacks()

batch\_size = int(args\_['batch'])

if args\_['dataset'] == 'malldataset':

model.fit(MallDataset().gen\_train(batch\_size, 224),

steps\_per\_epoch=MallDataset().get\_train\_num() // batch\_size,

validation\_data=MallDataset().gen\_valid(batch\_size, 224),

validation\_steps=MallDataset().get\_valid\_num() // batch\_size,

epochs=int(args\_['epochs']),

callbacks=callbacks)

elif args\_['dataset'] == 'shanghaitechdataset':

model.fit(ShanghaitechDataset().gen\_train(batch\_size, 224),

steps\_per\_epoch=ShanghaitechDataset().get\_train\_num() // batch\_size,

validation\_data=ShanghaitechDataset().gen\_valid(batch\_size, 224),

validation\_steps=ShanghaitechDataset().get\_valid\_num() // batch\_size,

epochs=int(args\_['epochs']),

callbacks=callbacks)

elif args\_['dataset'] == 'Crowdataset':

model.fit(CrowDataset().gen\_train(batch\_size, 224),

steps\_per\_epoch=CrowDataset().get\_train\_num() // batch\_size,

validation\_data=CrowDataset().gen\_valid(batch\_size, 224),

validation\_steps=CrowDataset().get\_valid\_num() // batch\_size,

epochs=int(args\_['epochs']),

callbacks=callbacks)

else:

print('not support this dataset')

if \_\_name\_\_ == '\_\_main\_\_':

args = parse\_command\_params()

train(args)

#服务端代码

启动类

package website.abyss.flow\_density\_detection;

import org.mybatis.spring.annotation.MapperScan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

@MapperScan("website.abyss.flow\_density\_detection.mapper")

public class FlowDensityDetectionApplication {

public static void main(String[] args) {

SpringApplication.run(FlowDensityDetectionApplication.class, args);

}

}

配置文件

server.port=8081

spring.datasource.url=jdbc:mysql://101.132.\*\*.\*\*\*:3306/flow\_density\_detection?useUnicode=true&characterEncoding=utf-8&serverTimezone=GMT&useSSL=false

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.username=root

spring.datasource.password=password

spring.datasource.type= com.alibaba.druid.pool.DruidDataSource

mybatis.mapper-locations=classpath:website/abyss/flow\_density\_detection/mapper/\*.xml

mybatis.type-aliases-package=website.abyss.flow\_density\_detection.entities

mybatis.configuration.map-underscore-to-camel-case=true

mybatis.configuration.use-generated-keys=true

spring.jackson.date-format=yyyy-MM-dd HH:mm:ss

spring.jackson.default-property-inclusion=non\_null

spring.jackson.deserialization.fail-on-unknown-properties=false

Maven配置文件

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.6.5</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>website.abyss</groupId>

<artifactId>flow\_density\_detection</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>flow\_density\_detection</name>

<description>flow\_density\_detection</description>

<properties>

<java.version>11</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.mybatis.spring.boot</groupId>

<artifactId>mybatis-spring-boot-starter</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>druid</artifactId>

<version>1.2.4</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.16</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<configuration>

<excludes>

<exclude>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

</exclude>

</excludes>

</configuration>

</plugin>

</plugins>

</build>

</project>

数据包类

package website.abyss.flow\_density\_detection.utils;

import lombok.Data;

import lombok.EqualsAndHashCode;

@Data

@EqualsAndHashCode(callSuper = false)

public class Result {

//状态码

private int code;

//提示消息

private String msg;

//数据

private Object result;

public void setSuccess(String msg, Object result){

this.code=200;

this.msg=msg;

this.result=result;

}

public void setInfo(String msg, Object result){

this.code=400;

this.msg=msg;

this.result=result;

}

}

实体类

package website.abyss.flow\_density\_detection.entities;

@lombok.Data

public class Data {

private String cameraId;

private Double latitude;

private Double longitude;

private Integer num;

}

Controller类

package website.abyss.flow\_density\_detection.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import website.abyss.flow\_density\_detection.entities.Data;

import website.abyss.flow\_density\_detection.service.DataService;

import website.abyss.flow\_density\_detection.utils.Result;

import java.util.List;

@RestController

@RequestMapping("/data")

public class DataController {

private DataService dataService;

private Result result = new Result();

@Autowired

public DataController(DataService dataService) {

this.dataService = dataService;

}

/\*\*

\* 通过经纬度查询位置的人数情况

\* @param data

\* @return

\*/

@RequestMapping("/getData")

public Result getData(@RequestBody Data data){

Data data1 = dataService.getData(data);

if(data1!=null){

result.setSuccess("查询成功",data1);

}else result.setInfo("信息缺失",null);

return result;

}

/\*\*

\* 查询所有位置的人数列表

\* @return

\*/

@RequestMapping("/getDataList")

public Result getDataList(){

List<Data> dataList = dataService.getDataList();

if(dataList.size()>0){

result.setSuccess("查询成功",dataList);

}else result.setInfo("信息缺失",null);

return result;

}

/\*\*

\* 更新对应位置的人数情况

\* @param dataList

\* @return

\*/

@RequestMapping("/setDataList")

public Result setDataList(@RequestBody List<Data> dataList){

for(Data data:dataList){

Data data1 = dataService.getData(data);

if(data1!=null){

dataService.updateData(data);

}else dataService.insertData(data);

}

result.setSuccess("数据更新成功",null);

return result;

}

}

#Service类

package website.abyss.flow\_density\_detection.service;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import website.abyss.flow\_density\_detection.entities.Data;

import website.abyss.flow\_density\_detection.mapper.DataMapper;

import java.util.List;

@Service

@Transactional

public class DataService {

private DataMapper dataMapper;

@Autowired

public DataService(DataMapper dataMapper) {

this.dataMapper = dataMapper;

}

public Data getData(Data data) {

return dataMapper.getData(data);

}

public List<Data> getDataList() {

return dataMapper.getDataList();

}

public void updateData(Data data) {

dataMapper.updateData(data);

}

public void insertData(Data data) {

dataMapper.insertData(data);

}

}

#Mapper接口类

package website.abyss.flow\_density\_detection.mapper;

import org.apache.ibatis.annotations.Mapper;

import website.abyss.flow\_density\_detection.entities.Data;

import java.util.List;

@Mapper

public interface DataMapper {

Data getData(Data data);

List<Data> getDataList();

void updateData(Data data);

void insertData(Data data);

}

#Mapper实现文件

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="website.abyss.flow\_density\_detection.mapper.DataMapper">

<resultMap id="BaseResultMap" type="website.abyss.flow\_density\_detection.entities.Data">

<constructor>

<arg column="camera\_id" javaType="java.lang.String" jdbcType="VARCHAR" />

<arg column="latitude" javaType="java.lang.Double" jdbcType="DOUBLE" />

<arg column="longitude" javaType="java.lang.Double" jdbcType="DOUBLE" />

<arg column="num" javaType="java.lang.Integer" jdbcType="INTEGER" />

</constructor>

</resultMap>

<insert id="insertData" parameterType="website.abyss.flow\_density\_detection.entities.Data">

insert into `data`(camera\_id,latitude,longitude,num) values (#{cameraId},#{latitude},#{longitude},#{num})

</insert>

<update id="updateData" parameterType="website.abyss.flow\_density\_detection.entities.Data">

update `data` set num = #{num} where latitude = #{latitude} and longitude = #{longitude}

</update>

<select id="getData" parameterType="website.abyss.flow\_density\_detection.entities.Data" resultType="website.abyss.flow\_density\_detection.entities.Data">

select \* from `data` where latitude = #{latitude} and longitude = #{longitude}

</select>

<select id="getDataList" resultType="website.abyss.flow\_density\_detection.entities.Data">

select \* from `data`

</select>

</mapper>

#数据库设计

SET FOREIGN\_KEY\_CHECKS=0;

-- ----------------------------

-- Table structure for data

-- ----------------------------

DROP TABLE IF EXISTS `data`;

CREATE TABLE `data` (

`camera\_id` varchar(50) NOT NULL,

`latitude` double(12,6) NOT NULL,

`longitude` double(12,6) NOT NULL,

`num` int(11) NOT NULL DEFAULT '0',

PRIMARY KEY (`camera\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

#客户端

// app.js

App({

  onLaunch() {

    // 展示本地存储能力

    const logs = wx.getStorageSync('logs') || []

    logs.unshift(Date.now())

    wx.setStorageSync('logs', logs)

    // 登录

    wx.login({

      success: res => {

        // 发送 res.code 到后台换取 openId, sessionKey, unionId

      }

    })

  },

  globalData: {

    userInfo: null

  }

})

//app.json

{

    "pages": [

        "pages/homepage/homepage",

        "pages/map/map"

    ],

    "window": {

        "backgroundTextStyle": "light",

        "navigationBarBackgroundColor": "#fff",

        "navigationBarTitleText": "",

        "navigationBarTextStyle": "black"

    },

    "style": "v2",

    "tabBar": {

        "list": [

            {

                "pagePath": "pages/homepage/homepage",

                "text": "我的",

                "iconPath": "/images/myoff.png",

                "selectedIconPath": "/images/myon.png"

            },

            {

                "pagePath": "pages/map/map",

                "text": "地图",

                "iconPath": "/images/mapoff.png",

                "selectedIconPath": "/images/mapon.png"

            }

        ]

    },

    "sitemapLocation": "sitemap.json"

}

/\*\*app.wxss\*\*/

@import "./pages/homepage/main.wxss";

@import "./pages/homepage/icon.wxss";

.container {

  height: 100%;

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: space-between;

  padding: 200rpx 0;

  box-sizing: border-box;

}

//homepage.wxss

@import "/app.wxss";

.intro {

  margin: 30px;

  text-align: center;

}

.pageContainer{

  min-height: 100vh;

  background-color: rgb(242, 242, 242);

}

.box {

  width: 702rpx;

  margin: 10rpx auto;

  overflow: hidden;

  background: #fff;

  border-radius: 5px;

  padding: 5px;

  box-sizing: border-box;

}

.head{

  widows: 100px;

  height: 100px;

  border-radius:50px;

}

.r {

  white-space: nowrap;

  animation: 10s loop linear infinite normal;

  display: inline-block;

  vertical-align: top;

  font-size: 26rpx;

  color: #333;

}

.button{

  width:80%;

  margin:30px auto 0;

  text-align: center;

 }

 .nickname{

  width: 80%;

  margin:0 auto;

  text-align: center;

  font-size: 2em;

 }

.headimg{

  margin: 0 auto;

  display: flex;

  justify-content: center;

  overflow: hidden;

  width: 161rpx;

  height: 161rpx;

  border-radius: 200px;

 }

.line1{

    white-space:nowrap;

    overflow:hidden;

    text-overflow:ellipsis;

}

.open-data-userAvatarUrl {

  width: 160rpx;

  height: 160rpx;

  display: block;

  overflow: hidden;

  border-radius: 50%;

  margin: 40rpx auto;

}

.open-data-userNickName{

  max-width: 105rpx;

  display:block;

}

.usermotto {

  margin-top: 200px;

}

@keyframes loop {

  0% {

    transform: translateX(350px);

    -webkit-transform: translateX(350px);

  }

  100% {

    transform: translateX(-100%);

    -webkit-transform: translateX(-100%);

  }

}

@-webkit-keyframes loop {

  0% {

    transform: translateX(300px);

    -webkit-transform: translateX(300px);

  }

  100% {

    transform: translateX(-100%);

    -webkit-transform: translateX(-100%);

  }}

//homepage.wxml

<view class="pageContainer">

  <view class="box flex-row" >

    <view class="padding-sm">

      <text class="r">欢迎使用strdsty小程序!   轻轻一点方便您的日常出行小助手。本页面为用户自愿授权，方便我们统计数据，授权成功后会自动跳转地图页面，不授权不影响地图功能使用。</text>

    </view>

  </view>

  <text>\n\n\n\n</text>

  <view class="nickname"> <text>{{nickName}}</text></view>

    <text>\n\n</text>

  <image class="headimg" src="{{avatarUrl}}"></image>

  <view > <text>{{gender}}</text></view>

  <text>\n\n</text>

  <button  bindtap="login" >授权登录</button>

</view>

//homepage.json

{

  "usingComponents": {},

  "enablePullDownRefresh": true,

  "backgroundColor": "#ddd",

  "backgroundTextStyle": "light"

}

//homepage.js

  Page({

    data: {

     canIUse: wx.canIUse('button.open-type.getUserInfo'),

     head:"",

     motto: 'Hello World',

     userInfo: {},

     hasUserInfo: false,

     canIUse: wx.canIUse('button.open-type.getUserInfo'),

     canIUseGetUserProfile: false,

     canIUseOpenData: wx.canIUse('open-data.type.userAvatarUrl') && wx.canIUse('open-data.type.userNickName')// 如需尝试获取用户信息可改为true

    },

    //请求API授权，获得用户头像和昵称

    login() {

      console.log("login")

      wx.getUserProfile({

        desc: '用于完善会员资料', // 声明获取用户个人信息后的用途，后续会展示在弹窗中，请谨慎填写

        success: (res) => {

          console.log("success",res)

          this.setData({

              avatarUrl:res.userInfo.avatarUrl,

              nickName:res.userInfo.nickName,

              city:res.userInfo.city,

              hasUserInfo: true

          })

          wx.switchTab({

            url: '/pages/map/map',

          })

      } ,

       fail(err) { //接口调用失败的回调函数 用户拒绝授权登录后，出现的提示窗

            console.error(err) //打印输出错误数据

            console.error("123")

            wx.showToast({ // 拒绝登录 显示消息提示框

              image: "/images/error.jpeg",

              title: '您拒绝了授权！', // 提示用户：'用户拒绝授权'

              duration: 1000 // 提示语出现时间延迟1s

            });

        },

    })

    }

  })

//map.js

const normalCallout = {

}

const customCallout1 = {

}

const customCallout2 = {

}

const customCallout3 = {

}

const customCallout4 = {

}

  const customCallout5 = {

}

const customCallout6 = {

}

      const customCallout7 = {

      }

      const customCallout8 = {

      }

      const customCallout9 = {

       }

const allMarkers = [normalCallout, customCallout1, customCallout2, customCallout3, customCallout4, customCallout5, customCallout6, customCallout7, customCallout8, customCallout9]

var ress0,ress1,ress2,ress3,ress4,ress5,ress6,ress7,ress8,ress9

Page({

  data: {

      latitude: 32.19994,

      longitude: 119.516520,

      list: [{cameraId:"", latitude:"", longitude:"", num:""}

  ],

      markers: [],

      customCalloutMarkerIds: [],

      num: 1

  },

  onReady: function (e) {

    this.mapCtx = wx.createMapContext('myMap')

    const markers = allMarkers

    let flag=0

    wx.request({

      url: 'http://www.abyss.website:8081/data/getDataList',

      headers: {

        'Content-Type': 'application/json'

      },

      success: (res) => {

        if( res.data.result[0].num<0){

          res.data.result[0].num=='\n“当前设备故障”'

          ress0='\n“当前设备故障”'

         }

         else{

           ress0=res.data.result[0].num+"人"

         }

       if( res.data.result[2].num<0){

        res.data.result[2].num=='\n“当前设备故障”'

        ress1='\n“当前设备故障”'

       }

       else{

         ress1=res.data.result[2].num+"人"

       }

       if( res.data.result[3].num<0){

        res.data.result[3].num=='\n“当前设备故障”'

        ress2='\n“当前设备故障”'

       }

       else{

         ress2=res.data.result[3].num+"人"

       }

       if( res.data.result[4].num<0){

        res.data.result[4].num=='\n“当前设备故障”'

        ress3='\n“当前设备故障”'

       }

       else{

         ress3=res.data.result[4].num+"人"

       }

       if( res.data.result[5].num<0){

        res.data.result[5].num=='\n“当前设备故障”'

        ress4='\n“当前设备故障”'

       }

       else{

         ress4=res.data.result[5].num+"人"

       }

       if( res.data.result[6].num<0){

        res.data.result[6].num=='\n“当前设备故障”'

        ress5='\n“当前设备故障”'

       }

       else{

         ress5=res.data.result[6].num+"人"

       }

       if( res.data.result[7].num<0){

        res.data.result[7].num=='\n“当前设备故障”'

        ress6='\n“当前设备故障”'

       }

       else{

         ress6=res.data.result[7].num+"人"

       }

       if( res.data.result[8].num<0){

        res.data.result[8].num=='\n“当前设备故障”'

        ress7='\n“当前设备故障”'

       }

       else{

         ress7=res.data.result[8].num+"人"

       }

       if( res.data.result[9].num<0){

        res.data.result[9].num=='\n“当前设备故障”'

        ress8='\n“当前设备故障”'

       }

       else{

         ress8=res.data.result[9].num+"人"

       }

       if( res.data.result[1].num<0){

        res.data.result[1].num=='\n“当前设备故障”'

        ress9='\n“当前设备故障”'

       }

       else{

         ress9=res.data.result[1].num+"人"

       }

       const ms=1000;

       setTimeout(()=>{

       this.setData({

        markers:[

          {

            id: 1,

            latitude: 32.2054,

            longitude: 119.511520,

            iconPath: '/images/normal.png',

              callout: {

                content: " ID:1\n 一食堂 \n"+  ress0,

                padding: 10,

                display: 'ALWAYS',

                textAlign: 'center',

              },

          }

        ,  {

          id: 2,

          latitude: 32.19714,

          longitude: 119.524090,

          iconPath: '/images/normal.png',

            callout: {

              content: "ID:2\n  二食堂 \n"+  ress1,

              padding: 10,

              display: 'ALWAYS',

              textAlign: 'center',

            },

        },

        {

          id: 3,

          latitude: 32.19494,

          longitude: 119.510520,

          iconPath: '/images/normal.png',

            callout: {

              content: " ID:3\n 三食堂 \n"+  ress0,

              padding: 10,

              display: 'ALWAYS',

              textAlign: 'center',

            },

        }

      ,

      {

        id: 4,

        latitude: 32.19394,

        longitude: 119.52100,

        iconPath: '/images/normal.png',

          callout: {

            content: " ID:4\n 四食堂 \n"+  ress3,

            padding: 10,

            display: 'ALWAYS',

            textAlign: 'center',

          },

      }

    ,

    {

      id: 5,

      latitude: 32.196094,

      longitude: 119.51820,

      iconPath: '/images/normal.png',

        callout: {

          content: " ID:5\n 五食堂 \n"+  ress4,

          padding: 10,

          display: 'ALWAYS',

          textAlign: 'center',

        },

    }

  ,

  {

    id: 6,

    latitude: 32.1984,

    longitude: 119.51390,

    iconPath: '/images/normal.png',

      callout: {

        content: " ID:6\n 六食堂 \n"+  ress5,

        padding: 10,

        display: 'ALWAYS',

        textAlign: 'center',

      },

  }

,

{

  id: 7,

  latitude: 32.201944,

  longitude: 119.514520,

  iconPath: '/images/normal.png',

    callout: {

      content: " ID:7\n 七食堂 \n"+  ress6,

      padding: 10,

      display: 'ALWAYS',

      textAlign: 'center',

    },

}

,

{

  id: 8,

  latitude: 32.198094,

  longitude: 119.509020,

  iconPath: '/images/normal.png',

    callout: {

      content: " ID:8\n 江大超市 \n"+  ress7,

      padding: 10,

      display: 'ALWAYS',

      textAlign: 'center',

    },

}

,

{

  id: 9,

  latitude: 32.202224,

  longitude: 119.518520,

  iconPath: '/images/normal.png',

    callout: {

      content: " ID:9\n 水果店 \n"+  ress8,

      padding: 10,

      display: 'ALWAYS',

      textAlign: 'center',

    },

}

,

{

  id: 10,

  latitude: 32.194594,

  longitude: 119.51520,

  iconPath: '/images/normal.png',

    callout: {

      content: " ID:10\n 学工书店 \n"+  ress9,

      padding: 10,

      display: 'ALWAYS',

      textAlign: 'center',

    },

}

,

  ],

        customCalloutMarkerIds: [2, 3, 4, 5, 6, 7, 8, 9, 10],

      })

       },ms)

      }

    })

  },

  changeMarkerId() {

    const customCalloutMarkerIds = this.data.customCalloutMarkerIds.slice()

    const top = customCalloutMarkerIds.shift()

    customCalloutMarkerIds.push(top)

    this.setData({

      customCalloutMarkerIds

    })

  },

  markertap(e) {

    console.log('@@@ markertap', e)

  },

  callouttap(e) {

    console.log('@@@ callouttap', e)

  },

  labeltap(e) {

    console.log('@@@ labeltap', e)

  },

  changeContent() {

    const num = Math.floor(Math.random() \* 10)

    this.setData({num})

  },

  login() {

    console.log("login")

    wx.getUserProfile({

      desc: '用于完善会员资料', // 声明获取用户个人信息后的用途，后续会展示在弹窗中，请谨慎填写

      success: (res) => {

        console.log("success",res)

           this.setData({

            avatarUrl:res.userInfo.avatarUrl,

            nickName:res.userInfo.nickName,

            city:res.userInfo.city,

            hasUserInfo: true

        })

      }

    })

  }

})

//map.json

{

  "usingComponents": {},

  "navigationBarBackgroundColor": "#f5c92c",

  "navigationBarTitleText": "",

  "navigationStyle":"default"

}

//map.wxml

<view class="page-body">

  <view class="page-section page-section-gap">

    <map

      id="myMap"

      style="width: 750rpx; height:1390rpx;"

      latitude="{{latitude}}"

      longitude="{{longitude}}"

      bindmarkertap="markertap"

      bindcallouttap="callouttap"

      bindlabeltap="labeltap"

      markers="{{markers}}"

      scale="14"

    >

      <cover-view slot="callout">

        <block wx:for="{{customCalloutMarkerIds}}" wx:key="\*this">

          <cover-view  class="customCallout" marker-id="{{item}}" >

            <cover-view class="content">

              {{num}}-{{item}}-{{index}}

            </cover-view>

          </cover-view>

        </block>

      </cover-view>

    </map>

  </view>

</view>

//map.wxss

.page-section-gap{

  box-sizing: border-box;

  padding: 0 0rpx;

}

.page-body-button {

  margin-bottom: 0rpx;

}

.customCallout {

  box-sizing: border-box;

  background-color: #fff;

  border: 1px solid #ccc;

  border-radius: 30px;

  width: 150px;

  height: 40px;

  display: inline-flex;

  padding: 5px 20px;

  justify-content: center;

  align-items: center;

}

.circle {

  width: 20px;

  height: 20px;

  border: 1px solid #ccc;

  border-radius: 50%;

}

.icon {

  width: 16px;

  height: 16px;

}

.content {

  flex: 0 1 auto;

  margin: 0 10px;

  font-size: 14px;

}