WTF Daily Blog

斗大的熊猫

TensorFlow练习16: 根据大脸判断性别和年龄

本帖使用TensorFlow做一个根据脸部推断照片人物年龄和性别的练习，网上有很多类似app。

训练数据 – Adience数据集

Adience数据集来源为Flickr相册，由用户使用iPhone或者其它智能手机设备拍摄，该数据集主要用于进行年龄和性别的未经过滤的面孔估计。同时，里面还进行了相应的landmark的标注，其中包含2284个类别和26580张图片。

Adience数据集下载地址：http://www.openu.ac.il/home/hassner/Adience/data.html#agegender

由于数据源ftp站点被墙，我只能使用梯子，下载过程非常漫长和痛苦。为了让你免受折磨，我传了一份到网盘。

代码

```
1  import os
2  import glob
3  import tensorflow as tf # 0.12
```

```
4  from tensorflow.contrib.layers import *
5  from tensorflow.contrib.slim.python.slim.nets.inception_v3 import inception_v3_base
6  import numpy as np
7  from random import shuffle
8
9  age_table=['(0, 2)','(4, 6)','(8, 12)','(15, 20)','(25, 32)','(38, 43)','(48, 53)','(60, 100)']
10 sex_table=['f','m']  # f:女; m:男
11
12 # AGE==True 训练年龄模型，False,训练性别模型
13 AGE = False
14
15 if AGE == True:
16     lables_size = len(age_table) # 年龄
17 else:
18     lables_size = len(sex_table) # 性别
19
20 face_set_fold = 'AdienceBenchmarkOfUnfilteredFacesForGenderAndAgeClassification'
21
22 fold_0_data = os.path.join(face_set_fold, 'fold_0_data.txt')
23 fold_1_data = os.path.join(face_set_fold, 'fold_1_data.txt')
24 fold_2_data = os.path.join(face_set_fold, 'fold_2_data.txt')
25 fold_3_data = os.path.join(face_set_fold, 'fold_3_data.txt')
26 fold_4_data = os.path.join(face_set_fold, 'fold_4_data.txt')
27
28 face_image_set = os.path.join(face_set_fold, 'aligned')
29
30 def parse_data(fold_x_data):
31     data_set = []
32
33     with open(fold_x_data, 'r') as f:
34         line_one = True
35         for line in f:
36             tmp = []
37             if line_one == True:
38                 line_one = False
39                 continue
40
41             tmp.append(line.split('\t')[0])
42             tmp.append(line.split('\t')[1])
43             tmp.append(line.split('\t')[3])
44             tmp.append(line.split('\t')[4])
45
```

```
            file_path = os.path.join(face_image_set, tmp[0])
            if os.path.exists(file_path):
                filenames = glob.glob(file_path + "/*.jpg")
                for filename in filenames:
                    if tmp[1] in filename:
                        break
                if AGE == True:
                    if tmp[2] in age_table:
                        data_set.append([filename, age_table.index(tmp[2])])
                else:
                    if tmp[3] in sex_table:
                        data_set.append([filename, sex_table.index(tmp[3])])

    return data_set

data_set_0 = parse_data(fold_0_data)
data_set_1 = parse_data(fold_1_data)
data_set_2 = parse_data(fold_2_data)
data_set_3 = parse_data(fold_3_data)
data_set_4 = parse_data(fold_4_data)

data_set = data_set_0 + data_set_1 + data_set_2 + data_set_3 + data_set_4
shuffle(data_set)

# 缩放图像的大小
IMAGE_HEIGHT = 227
IMAGE_WIDTH = 227
# 读取缩放图像
jpg_data = tf.placeholder(dtype=tf.string)
decode_jpg = tf.image.decode_jpeg(jpg_data, channels=3)
resize = tf.image.resize_images(decode_jpg, [IMAGE_HEIGHT, IMAGE_WIDTH])
resize = tf.cast(resize, tf.uint8) / 255
def resize_image(file_name):
    with tf.gfile.FastGFile(file_name, 'r') as f:
        image_data = f.read()
    with tf.Session() as sess:
        image = sess.run(resize, feed_dict={jpg_data: image_data})
    return image

pointer = 0
# 有点慢(先睡了)，应该先处理好图片或使用string_input_producer
def get_next_batch(data_set, batch_size=128):
```

```
    global pointer
    batch_x = []
    batch_y = []
    for i in range(batch_size):
        batch_x.append(resize_image(data_set[pointer][0]))
        batch_y.append(data_set[pointer][1])
        pointer += 1
    return batch_x, batch_y

batch_size = 128
num_batch = len(data_set) // batch_size

X = tf.placeholder(dtype=tf.float32, shape=[batch_size, IMAGE_HEIGHT, IMAGE_WIDTH, 3])
Y = tf.placeholder(dtype=tf.int32, shape=[batch_size])

def conv_net(nlabels, images, pkeep=1.0):
    weights_regularizer = tf.contrib.layers.l2_regularizer(0.0005)
    with tf.variable_scope("conv_net", "conv_net", [images]) as scope:
        with tf.contrib.slim.arg_scope([convolution2d, fully_connected], weights_regularizer=weights_regular
            with tf.contrib.slim.arg_scope([convolution2d], weights_initializer=tf.random_normal_initializer(
                conv1 = convolution2d(images, 96, [7,7], [4, 4], padding='VALID', biases_initializer=tf.cons
                pool1 = max_pool2d(conv1, 3, 2, padding='VALID', scope='pool1')
                norm1 = tf.nn.local_response_normalization(pool1, 5, alpha=0.0001, beta=0.75, name='norm1')
                conv2 = convolution2d(norm1, 256, [5, 5], [1, 1], padding='SAME', scope='conv2')
                pool2 = max_pool2d(conv2, 3, 2, padding='VALID', scope='pool2')
                norm2 = tf.nn.local_response_normalization(pool2, 5, alpha=0.0001, beta=0.75, name='norm2')
                conv3 = convolution2d(norm2, 384, [3, 3], [1, 1], biases_initializer=tf.constant_initializer(
                pool3 = max_pool2d(conv3, 3, 2, padding='VALID', scope='pool3')
                flat = tf.reshape(pool3, [-1, 384*6*6], name='reshape')
                full1 = fully_connected(flat, 512, scope='full1')
                drop1 = tf.nn.dropout(full1, pkeep, name='drop1')
                full2 = fully_connected(drop1, 512, scope='full2')
                drop2 = tf.nn.dropout(full2, pkeep, name='drop2')
    with tf.variable_scope('output') as scope:
        weights = tf.Variable(tf.random_normal([512, nlabels], mean=0.0, stddev=0.01), name='weights')
        biases = tf.Variable(tf.constant(0.0, shape=[nlabels], dtype=tf.float32), name='biases')
        output = tf.add(tf.matmul(drop2, weights), biases, name=scope.name)
    return output

def training():
    logits = conv_net(lables_size, X)

```

```
130     def optimizer(eta, loss_fn):
131         global_step = tf.Variable(0, trainable=False)
132         optz = lambda lr: tf.train.MomentumOptimizer(lr, 0.9)
133         lr_decay_fn = lambda lr,global_step : tf.train.exponential_decay(lr, global_step, 100, 0.97, staircas
134         return tf.contrib.layers.optimize_loss(loss_fn, global_step, eta, optz, clip_gradients=4., learning_r
135 
136     def loss(logits, labels):
137         cross_entropy = tf.nn.sparse_softmax_cross_entropy_with_logits(logits, labels)
138         cross_entropy_mean = tf.reduce_mean(cross_entropy)
139         regularization_losses = tf.get_collection(tf.GraphKeys.REGULARIZATION_LOSSES)
140         total_loss = cross_entropy_mean + 0.01 * sum(regularization_losses)
141         loss_averages = tf.train.ExponentialMovingAverage(0.9)
142         loss_averages_op = loss_averages.apply([cross_entropy_mean] + [total_loss])
143         with tf.control_dependencies([loss_averages_op]):
144             total_loss = tf.identity(total_loss)
145         return total_loss
146     # loss
147     total_loss = loss(logits, Y)
148     # optimizer
149     train_op = optimizer(0.001, total_loss)
150 
151     saver = tf.train.Saver(tf.global_variables())
152     with tf.Session() as sess:
153         sess.run(tf.global_variables_initializer())
154 
155         global pointer
156         epoch = 0
157         while True:
158             pointer = 0
159             for batch in range(num_batch):
160                 batch_x, batch_y = get_next_batch(data_set, batch_size)
161                 _, loss_value = sess.run([train_op, total_loss], feed_dict={X:batch_x, Y:batch_y})
162                 print(epoch, batch, loss_value)
163             saver.save(sess, 'age.module' if AGE == True else 'sex.module')
164             epoch += 1
165 
166 training()
167 
168 """
169 # 检测性别和年龄
170 # 把batch_size改为1
171 def detect_age_or_sex(image_path):
```

```
    logits = conv_net(lables_size, X)
    saver = tf.train.Saver()

    with tf.Session() as sess:
        saver.restore(sess, './age.module' if AGE == True else './sex.module')

        softmax_output = tf.nn.softmax(logits)
        res = sess.run(softmax_output, feed_dict={X:[resize_image(image_path)]})
        res = np.argmax(res)

        if AGE == True:
            return age_table[res]
        else:
            return sex_table[res]
"""
```

后续：使用OpenCV检测提取人脸，然后使用训练好的模型判断性别和年龄。

相关资源：

- http://www.openu.ac.il/home/hassner/projects/cnn_agegender/
- https://github.com/GilLevi/AgeGenderDeepLearning
- https://cmusatyalab.github.io/openface/
- https://github.com/RiweiChen/DeepFace
- https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/
- http://blog.csdn.net/qq_14845119/article/details/51913171

相关文章

Ubuntu 16.04 安装 Tensorflow(GPU支持)

使用Python实现神经网络

TensorFlow练习2: 对评论进行分类

TensorFlow练习6: 基于WiFi指纹的室内定位（autoencoder）...

TensorFlow练习8: 生成音乐

2016年12月11日 wtf ML、coding TensorFlow、人脸

《TensorFlow练习16: 根据大脸判断性别和年龄》有6个想法

xwz

2017年6月2日 下午3:23

全部读到内存之后，内存需要11g
但是对于训练时的速度提升非常大

Molly

2017年5月17日 下午8:02

多谢数据集！！

shaozhong

2017年3月3日 上午9:46

你好，感谢你分享的程序，但是请问我运行你的程序为什么总是报这个错误ValueError: Only call sparse_softmax_cross_entropy_with_logits with named arguments (labels=…, logits=…, …)
错误提示在第188,168,157行

zxsimple

2017年3月16日 下午7:52

把这行代码
cross_entropy = tf.nn.sparse_softmax_cross_entropy_with_logits(logits, labels)
改成：
cross_entropy = tf.nn.sparse_softmax_cross_entropy_with_logits(logits = logits, labels = labels)

陈文辉

2016年12月17日 下午12:40

你好，请问一下能共享一下你的网盘数据源么？
非常谢谢，我的邮箱splendon@163.com

wtf

2016年12月17日 下午12:45

https://pan.baidu.com/s/1bpadgQV