



山人七



# 基于mtcnn和facenet的实时人脸检测与识别系统开发



狗头山人七 · 8 个月前

简介：本文主要介绍了实时人脸检测与识别系统的详细方法。该系统基于python/opencv2/tensorflow环境，实现了从摄像头读取视频，检测人脸，识别人脸的功能。本系统代码地址：[real time face detection and recognition](#)。

## 1,前言

人脸识别是计算机视觉研究领域的一个热点。目前，在实验室环境下，许多人脸识别已经赶上（超过）人工识别精度（准确率：0.9427~0.9920），比如face++，DeepID3，FaceNet等（详情可以参考：[基于深度学习的人脸识别技术综述](#)）。但是，由于光线，角度，表情，年龄等多种因素，导致人脸识别技术无法在现实生活中广泛应用。本文基于python/opencv/tensorflow



- 2, 采用opencv2实现从摄像头读取视频帧;
- 3, 对读取的视频帧采用mtcnn方法, 检测人脸;
- 4, 采用预训练的facenet对检测的人脸进行embedding, embedding成128维度的特征;
- 5, 对人脸embedding特征采用knn进行分类, 实现人脸识别;
- 6, 结果与改进;
- 7, 总结。



图1-1 人脸检测与识别系统概况

## 2,opencv2从摄像头读取视频帧

采用opencv2,可以很方便的实现从摄像头读取视频帧。下文代码实现了从摄像头读取视频帧,并转化为灰度图像的功能,更多内容可以参考:[Python-OpenCV 图像与视频处理](#)。

```
import numpy as np
import cv2

cap = cv2.VideoCapture(0)
```



```
    # Our operations on the frame come here
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Display the resulting frame
    cv2.imshow('frame', gray)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
```

### 3.人脸检测

人脸检测方法有许多，比如opencv自带的人脸Haar特征分类器和dlib人脸检测方法等。

对于opencv的人脸检测方法，有点是简单，快速；存在的问题是人脸检测效果不好。如图3-1所示，正面/垂直/光线较好的人脸，该方法可以检测出来，而侧面/歪斜/光线不好的人脸，无法检测。因此，该方法不适合现场应用。对于dlib人脸检测方法，效果好于opencv的方法，但是检测力度也难以达到现场应用标准。

本文中，我们采用了基于深度学习方法的mtcnn人脸检测系统（[mtcnn : Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Neural Networks](#)）。mtcnn人脸检测方法对自然环境中光线，角度和人脸表情变化更具有鲁棒性，人脸检测效果更好；同时，内存消耗不大，可以实现实时人脸检测。本文中采用mtcnn是基于python和tensorflow的实现（代



如图3-2所示，mtcnn方法成功检测出所有人脸。

下文代码采用opencv2的haarcascade\_frontalface\_alt2实现人脸检测。

```
%%time
#图片人脸检测
import cv2
import sys
# Get user supplied values
imagePath = "./multi_faces.jpg"
# Create the haar cascade
faceCascade = cv2.CascadeClassifier("./haarcascade_frontalface_alt2.xml")
# Read the image
image = cv2.imread(imagePath)#2
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)#3
# Detect faces in the image
faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.15,
    minNeighbors=5,
    minSize=(5,5),
    flags = cv2.cv.CV_HAAR_SCALE_IMAGE
) #4
print "Found {0} faces!".format(len(faces))#5
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2) #6
```



图3-1 opencv2人脸检测方法结果

图3-2 mtcnn人脸检测方法结果

mtcnn人脸检测部分代码摘录：

知

首发于  
山人·七-深度学习

写文章

登录

```
gpu_memory_fraction=1.0
with tf.Graph().as_default():
    gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=gpu_memory_fr
    sess = tf.Session(config=tf.ConfigProto(gpu_options=gpu_options, log_device
    with sess.as_default():
        pnet, rnet, onet = detect_face.create_mtcnn(sess, './davidsonberg_fac

bounding_boxes, _ = detect_face.detect_face(img, minsize, pnet, rnet, onet, t
nrof_faces = bounding_boxes.shape[0]#人脸数目
print('找到人脸数目为：{}'.format(nrof_faces))

crop_faces=[]
for face_position in bounding_boxes:
    face_position=face_position.astype(int)
    print(face_position[0:4])
    cv2.rectangle(img_color, (face_position[0], face_position[1]), (face_posi
    crop=img_color[face_position[1]:face_position[3],
                  face_position[0]:face_position[2],]


    crop = cv2.resize(crop, (96, 96), interpolation=cv2.INTER_CUBIC )

    print(crop.shape)
    crop_faces.append(crop)
plt.imshow(crop)
plt.show()
```

知



首发于  
山人.七.深度学习

 写文章

登录

知

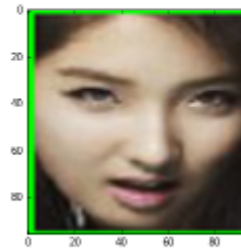
首发于  
山人·七·深度学习

写文章

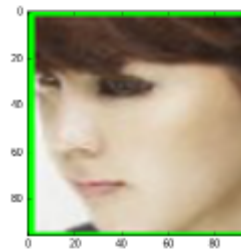
登录



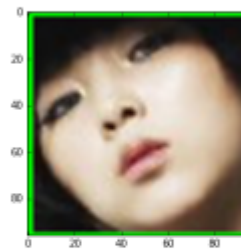
[298 203 360 286]  
(96, 96, 3)



[ 86 9 145 82]  
(96, 96, 3)



[551 266 625 345]  
(96, 96, 3)



[874 217 929 282]  
(96, 96, 3)







图3-3 剪切人脸区域

#### 4, facenet embedding

Facenet是谷歌研发的人脸识别系统，该系统是基于百万级人脸数据训练的深度卷积神经网络，可以将人脸图像embedding（映射）成128维度的特征向量。以该向量为特征，采用knn或者svm等机器学习方法实现人脸识别。Facenet在LFW数据集上识别准确率为0.9963，详情可以参见：[谷歌人脸识别系统FaceNet解析](#)。本文采用的是daidsandberg基于FaceScrub and CASIA-WebFace数据集预训练的Facenet模型，LFW测试集准确率为0.98。下文代码实现了恢复预训练facenet模型和使用模型进行embedding，应用代码参考了yobiface实现的代码。

```
#建立facenet embedding模型
print('建立facenet embedding模型')
tf.Graph().as_default()
sess = tf.Session()
images_placeholder = tf.placeholder(tf.float32, shape=(batch_size,
                                                    image_size,
                                                    image_size, 3), name=':

phase_train_placeholder = tf.placeholder(tf.bool, name='phase_train')
```



```

phase_train=phase_train_placeholder)

ema = tf.train.ExponentialMovingAverage(1.0)
saver = tf.train.Saver(ema.variables_to_restore())
#ckpt = tf.train.get_checkpoint_state(os.path.expanduser(model_dir))
#saver.restore(sess, ckpt.model_checkpoint_path)

model_checkpoint_path='./model-20160506.ckpt-500000'
#ckpt = tf.train.get_checkpoint_state(os.path.expanduser(model_dir))
#model_checkpoint_path='model-20160506.ckpt-500000'

#saver.restore(sess, ckpt.model_checkpoint_path)
saver.restore(sess, model_checkpoint_path)
print('facenet embedding模型建立完毕')

#####省略部分代码
emb_data = sess.run([embeddings],
                     feed_dict={images_placeholder: face_data, phase_t

```

## 5,人脸识别

对人脸进行embedding后，得到128维度的特征向量。以该特征向量为基础，可以采用任何机器学习的方法进行分类和识别。本文中，选取了knn（k-NearestNeighbor）方法（你可以换其



首先，需要训练分类器。训练数据为：类别1：目标人脸1；类别2：目标人脸2...，类别n：其他人脸。本代码中，类别1：我的人脸数据（经过人脸检测和embedding，共98个样本）；类别2：其它人脸（采用lfw数据集随机选取的人脸数据，共69个样本）。训练代码如下：

*#训练KNN分类*

```
from sklearn import metrics
from sklearn.externals import joblib

X_train, X_test, y_train, y_test = train_test_split(X, train_y, test_size=.3,

# KNN Classifier
def knn_classifier(train_x, train_y):
    from sklearn.neighbors import KNeighborsClassifier
    model = KNeighborsClassifier()
    model.fit(train_x, train_y)
    return model

classifiers = knn_classifier

model = classifiers(X_train,y_train)
predict = model.predict(X_test)

accuracy = metrics.accuracy_score(y_test, predict)
print ('accuracy: %.2f%%' % (100 * accuracy) )
```



训练后的分类器即可对人脸进行识别，代码如下：

```
model = joblib.load('knn.model')  
predict = model.predict(X_test)  
print ('识别结果为：{}'.format(predict))
```

## 6,结果与改进


### 6.1 运行结果

如图6-1所示，系统可以从摄像头获取视频，实时检测出人脸，并识别。系统不管对正面人脸，还是歪斜的人脸，以及不同表情的人脸均能有效的检测并识别，具有一定的鲁棒性。

知



首发于  
山人·七·深度学习


 写文章

登录

知



首发于  
山人·七·深度学习

 写文章

登录



图6-1 成功检测和识别出我和其他人

## 6.2 改进

本系统虽然采用了高准确率的人脸检测（mtcnn）与识别方法（facenet），可以实现实时人脸检测与识别，具有一定的鲁棒性。但是，如果要在现实生活中应用本系统，还需要做许多改进。

### 6.2.1 人脸检测

mtcnn人脸检测方法精确度很高，但是依然存在一些问题。

知



首发于  
山人·七·深度学习

写文章

登录

片。解决的方法可以有两种：（1）对训练照片进行旋转，在但前mtcnn网络的基础上进行finetune；（2）在视频帧读取后，旋转不同角度后，分别传入mtcnn进行人脸检测。前一种方法需要处理大量的人脸数据；后一种方法会影响实时检测的速度。

图6-2 难以检测出倾斜度过大的人脸

二，mtcnn人脸检测方法还存在另外的小问题：有时可能会被汪星人欺骗（好吧，汪星人的脸





图6-3 检测出潜伏在人类星球的汪星人脸

### 6.2.2 人脸识别

目前，基于深度神经网络方法的人脸识别系统在实验环境下准确率很高，已经达到或者超越人类的准确率，但是在现实环境中的准确率仅在60%-70%左右。现实环境中，光线/表情/角度/化妆等多种因素均会影响识别效果，有待研究新方法（参见：[基于深度学习的人脸识别技术综述](#)）。

本系统采用的是[davidsandberg](#)预训练的Facenet网络模型，如果需要更高准确率的facenet模型，可以考虑[openface](#)。

同时，当前系统的识别分类器是基于仅仅167个正负样本训练的knn分类器，测试准确率仅为94%左右。进一步改进可以考虑：（1）采集更多的类别与数量的训练样本训练分类器，实现



### 6.2.3 其它改进

本系统的核心方法是基于mtcnn和facenet，均是基于深度学习的方法。虽然神经网络模型不算大，但是相对于opencv内置的人脸检测和识别方法来讲内存消耗还是比较大的。本文运行环境为Ubuntu 16.04.1，8G内存，仅能设置为每5帧进行一次检测和识别（不影响实际效果）。可以考虑在不影响深度神经网络方法精确度的情况下，减少内存消耗，实现在移动设备（手机/平板等）运行。

同时，本系统代码采用的是ipython notebook（为了代码结构更加清晰，易读），运行一段时间后会导致内存满，kernel die的问题。

### 7,总结

本文主要介绍了实时人脸检测与识别系统的详细方法，该系统基于python/opencv2/tensorflow环境，实现了从摄像头读取视频，检测人脸，识别人脸的功能。实现类似功能的代码有openface，但是，openface核心是基于torch和lua。本人更喜欢python和tensorflow，因此，才有本文的产生。

本系统相对于opencv自带的人脸检测和识别的准确率更高，鲁棒性更好；但距离实际应用，还需要许多改进。同时，实时人脸检测与识别技术还需要许多改进和研究，期待和大家一起探索，谢谢。

「真诚赞赏，手留余香」

知

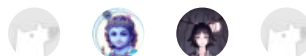


首发于  
山人.七-深度学习

写文章

登录

4 人赞赏



深度学习 ( Deep Learning )

人脸识别



69

☆ 收藏

分享

⚠ 举报



文章被以下专栏收录



山人.七-深度学习

追寻谷歌deepmind巨人脚步，研究深度学习。

进入专栏

113 条评论

写下你的评论...

知

首发于  
山人·七-深度学习

写文章

登录

basestring 改为str

tf.mul改成了乘×

因为它们在我的版本下面是报错的！！

然后

就报错！！

```
UnicodeError: Unpickling a python object failed: UnicodeDecodeError('ascii',
b'!W\xc0\xbbX\x1d\x7f<\x7f\xf0\xa3;5:\x81\xbb\x9b\x06>;L\x8d\x0c=j$\% \xbb\xf4[\x9a\xbcq\
xa7k\xbc\xe2#&\xbc\xa8W\x04;O\xa6\xd7;\xa1\x9c\xc8\xbb\xd1+\x17\xbc\xa1Y\x00\xba\x
80\x9e3;U\x07\r\xbc\xf6\xeaJ>m\x897\xbd\x11\xd2D\xbd
\xf2\x1b;\x1f\xb7f\xba\x92\xbcU\xac\xc9;\xd5\x97\x9f\xbb\xb6\xec(;a\xc5\x1f\xbd\x8d\xbd
"\xbe4$\x81;\x07\x7f\xe7;5L\xa7; \x07\xf7<\x96\xa0\r\xbcR\x11\x1b<+8\xa6;"
<\xf8\xbb5\x03M:\x07\xe5\xea:\x9drb\xbd\x16p&=W\x9f\xc4\xbb\x06\xc1\x89\xbc\xd2\x17\
x82\xbbf\xc22<\xae\x9f1:\xed\x10\x9b=\xfcb9K9\xbe\xff\xe6;"\x90\xc9\xbb\x11\xd0D>6\x
f1\x01\xbeS\xef\xbcz
\b9;\xe7<\x8e\xbb\xb2\xc7\xa3\xbc\x8cH\x1b<\nT\x81\xbbO\x8b\xfd;w\xc9\xc0\xbd\xe0\x
89\xbb=Vw\x84;N\x03\x96;
<sJ;Z\xaf\xbd\xbc\xfa\x02\x97;\x95}\x97<\x0fR\x12\xbc\xe3:$<k\xea\x86;\xfeo\xe3;\xb1\xbf
\x0e<yG\xcb\xbd\x9cEm<*\xe3\x8f;\x04\x1a\x87<\xabel\xb7\xbb\xfb\xef\xa3;
(\x87\x1f<\x91Y\xae:\x84\xbb\xb1:v\xa6\x1a<\xbe\xf5\x91=Zu\x0b>\x02i\x94\xbb\xe8\xde\
xa2\xbb!\xcb\r;pd\x04<\xa9\xd3\x8e\xbbX7\x8f;\x9b&\xc3<o\x9e\x0f\xbe\x9b\xb2_\xbd\xa6
t\xb1\xbaV\x1b\x84\xbb\x86\x95\x9a\xba%#@=\x87\xf0\xc2;\xc5\r\x7f<\xa3\xdd\x00\xbc\xf
8\x8f\x14<*\xe5\x19;E\n\x0f\xbd\xfc\x0b:\xbd\xaeW
\xbd\xf4)!=\x1a0\xbf\x8d\x9f\x9e<\xb5\x80A\xbc\xb9\x1b<\xd5:\xf1=\xe1\x01\xb9:\xae\x1
d\xae;\xc3g\xc9;\xa4\m\xbd!*H\xbd\x83\x91\xd0<Wa\xef\xbb\xc9\x8d\x08<\x11\x12\xb2;\x
a9~\xc3\xbbY\x8d1<\xd9X\xb3<&R\x06\xbe\xb7\xbc
>xZT;\x89\xa8\xed\xbbEL\x88\xbb4k\x97<', 2, 3, 'ordinal not in range(128)') You may need
to pass the encoding= option to numpy.load
```

知

首发于  
山人·七-深度学习

写文章

登录



香蕉蕉

model-20160506.ckpt-500000 and checkpoint from yobiface was put in  
model\_check\_point.

NotFoundError: Tensor name

"incept3b/in4\_conv1x1\_15/batch\_norm/cond/incept3b/in4\_conv1x1\_15/batch\_norm/moments/moment s\_1/variance/ExponentialMovingAverage/biased" not found in checkpoint files  
./model\_check\_point/model-20160506.ckpt-50000

7 个月前



狗头山人七（作者） 回复 香蕉蕉

查看对话

这个bug是因为tensorflow版本升级后的改变：tf.mul, tf.sub and tf.neg are deprecated in favor of tf.multiply, tf.subtract and tf.negative.将tf.mul()改为tf.multiply即可。我在升级完tensorflow1.0后也出现这个bug，详情可以参考

<https://github.com/tensorflow/tensorflow/releases>

。

7 个月前



狗头山人七（作者） 回复 香蕉蕉

查看对话

该bug也是因为tensorflow版本更新后的改变。但是，我暂时还没有找到解决的方法。新版本还不太熟悉，一时间也找不到好的办法，不好意思。

7 个月前

知



首发于  
山人.七-深度学习

写文章

登录



tensorflow升级为1.0后，可以采用官方提供的转换代码对原代码进行转化，地址：

<https://github.com/tensorflow/tensorflow/tree/r1.0/tensorflow/tools/compatibility>

。本文中的代码经转化后仍存在一些bug。

7 个月前

1 赞



狗头山人七（作者）

bug1：facenet.py文件经转化后，`array_ops.concat(3, net, name=name)`仍存在问题，需要改为`array_ops.concat(net,3 name=name)`。

7 个月前



香蕉蕉 回复 狗头山人七（作者）

查看对话

要用新的model文件，我用官方的compare和lfw的可以跑通哇

7 个月前

2 赞



jeff soong

截取人脸照片后，不用对齐吗？

7 个月前



狗头山人七（作者） 回复 jeff soong

查看对话

mtcnn截取的人脸部分，可以理解为对齐。

知



首发于  
山人.七-深度学习

写文章

登录



感觉不对吧，这个只是截取人脸部分，没有做变换，最起码得也得把人眼和嘴角拉平吧，不这么对齐人脸识别部分精度会下降很多吧

7 个月前

1

2

3

4

...

12

下一页

## 推荐阅读



### YOLO：实时快速目标检测

论文笔记：You Only Look Once: Unified, Real-Time Object Detection

n评论：基于深度学习方... [查看全文](#) >

狗头山人七 · 8 个月前 · 发表于 山人.七-深度学习

### 无可挑剔的法式鸭胸肉

鸭胸肉由于其浓郁风味深受欢迎，经过恰当的处理，成本30~50元的平价肉品也能登上大雅之堂。... [查看全文](#) >

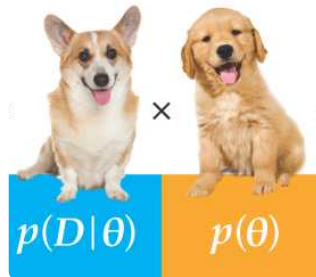
知



首发于  
山人·七-深度学习

写文章

登录



## Frequentist vs Bayesian 2 之不，是你的贝叶斯

在详细介绍什么是贝叶斯，贝叶斯有什么用等等之前，基于系列第一篇（为什么心理学可以是科学... [查看全文](#) >

Sharon · 15 天前 · 编辑精选