人工智能应用实践课程中期报告

——吕程

1. 论文解读

1、Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks

概要

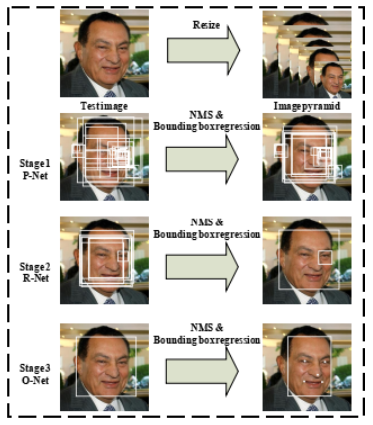
在本文中，提出新的级联架构来整合多任务卷积神经网络学习的问题。该算法有三个阶段组成：第一阶段，浅层的CNN快速产生候选窗体；第二阶段，通过更复杂的CNN精炼候选窗体，丢弃大量的重叠窗体；第三阶段，使用更加强大的CNN，实现候选窗体去留，同时显示五个面部关键点定位。

通过这个多任务学习框架，算法的性能可以显著提高。本文的主要贡献总结如下：

（1）提出了一种新的级联CNNs框架，用于联合人脸检测和对齐，并仔细设计了轻量级CNN架构以实现实时性能。  
（2）提出一种有效的在线难样本挖掘方法来提高性能。  
（3）对具有挑战性的基准进行了大量的实验，与在面部检测和面部对准任务中最先进的技术相比，该方法显示出显著的性能改进

总体框架

总体流程如下图所示：



给定一幅图像，我们首先调整它的大小以建立一个图像金字塔，这是以下三级级联框架的输入：

阶段1：利用完全卷积网络（称为建议网络（P-Net)）获得候选面部窗口及其边界框回归向量。然后基于估计的边界框回归向量校准候选窗口。之后，采用非最大抑制（NMS）来合并高度重叠的候选窗口。

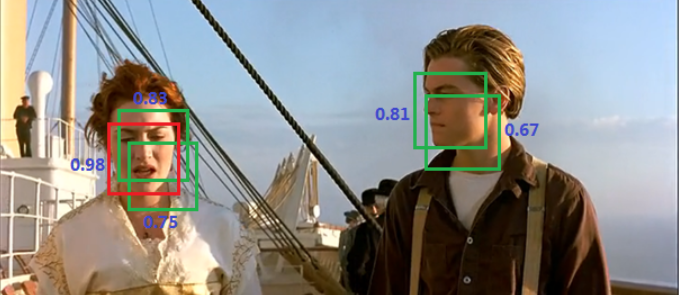
阶段2：所有候选窗口都被送到另一个叫做Refine Network（R-Net）的CNN，它进一步拒绝了大量错误的候选窗口，用边界框回归进行校准，并进行非最大抑制（NMS）。

阶段3：这个阶段与第二阶段相似，但在这个阶段我们的目标是通过更多的监督来识别人脸区域。 特别是，该网络将输出五个面部目标的位置。

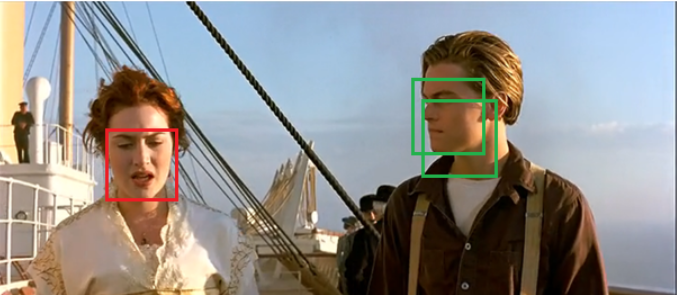
NMS——非极大值抑制

抑制的过程是一个迭代-遍历-消除的过程：

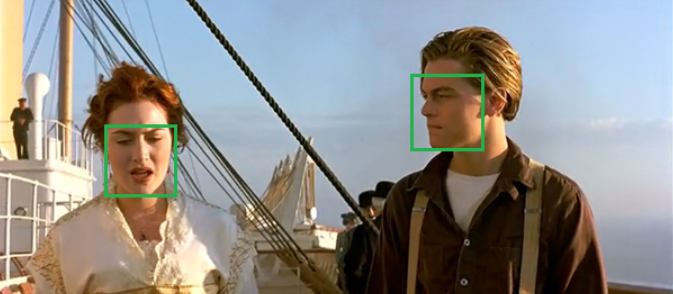
1. 将所有的框的得分排序，选中最高分及其对应框



1. 遍历其余的框，如果和当前最高分框的重叠面积（IOU）大于一定的阈值，就将框删除



1. 从未处理的框中继续选一个得分最高的，重复上述过程



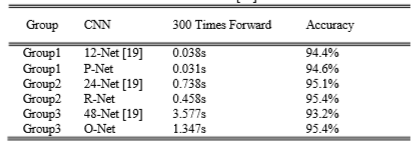
CNN网络结构

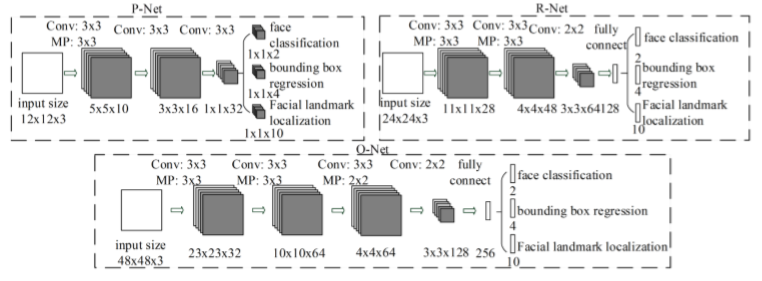
其中，多个CNN被设计用于人脸检测。但是，它的性能可能受到以下事实的限制：

（1）卷积层中的一些滤波器缺少可能限制其区分能力的多样性。

（2）与其他多类目标检测和分类任务相比，人脸检测是一项具有挑战性的二值分类任务，因此每层滤波器的数量可能会减少。为此，减少滤波器数量并将5×5滤波器更改为3×3滤波器，以减少计算量，同时增加深度以获得更好的性能。

通过这些改进，与之前的架构相比，可以在更少的运行时间内获得更好的性能（训练阶段的结果下表所示。为了公平比较，在每个组中使用相同的训练和验证数据）。CNN网络结构下图所示。将PReLU应用于卷积和完全连接层（输出层除外）之后作为非线性激活函数





训练

利用三项任务来训练CNN检测器：面部/非面部分类，边界框回归和面部标志定位。

1）脸部分类：学习目标表述为一个两级分类问题。对于每个样本https://img-blog.csdn.net/20180524135746962?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70，使用交叉熵损失：



其中https://img-blog.csdn.net/201805241548213?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70是由网络产生的表示样本https://img-blog.csdn.net/20180524135746962?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70是人脸的概率。符号https://img-blog.csdn.net/20180524154855592?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70∈{0,1}表示样本对应的真实标签。

2）边界框回归：对于每个候选窗口，我们预测它与最近的真实标签值之间的偏移（即边界框的左边，顶边，高度和宽度）。 学习目标被形容成为一个回归问题，使用每个样本https://img-blog.csdn.net/20180524135746962?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70的欧几里德损失：



其中https://img-blog.csdn.net/20180524160248375?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70是从卷积神经网络中获得的回归目标，https://img-blog.csdn.net/20180524160430539?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70是样本对应的真实坐标。 该真实坐标有四个坐标值，包括左上角坐标值、高度以及宽度值，因此https://img-blog.csdn.net/20180524160427245?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70∈ℝ4。

3）面部关键点定位：与边界框回归任务类似，将面部标志检测表示为回归问题，将欧几里得损失最小化：



https://img-blog.csdn.net/20180524161100970?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70是从卷积神经网络中获得的面部标志的坐标，https://img-blog.csdn.net/20180524161142305?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70是第https://img-blog.csdn.net/20180524161152523?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70个样本对应的真实面部关键点坐标。该真实面部关键点坐标包括左眼、右眼、鼻子、左嘴角和右嘴角等坐标，因此https://img-blog.csdn.net/20180524161142305?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70∈ℝ10。

4）多源训练：由于在每个CNN中完成不同的任务，所以在学习过程中存在不同类型的训练图像，如人脸，非人脸和部分对齐人脸。在这种情况下，上面列出的三个损失函数将不会使用。例如，对于背景区域的样本，只计算https://img-blog.csdn.net/20180524161800522?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3dlaXhpbl80MTY5NTU2NA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70，另外两个损失设置为0，这可以直接使用样本类型指标来实现。

5）在线难样本挖掘：与原始分类器训练完成后进行传统难样本挖掘不同，在适应训练过程的人脸/非人脸分类任务中进行在线难样本挖掘。特别是，在每个小样本批次中，对从所有样本向前传播中计算出的损失进行排序，并选择最高的70％作为难样本。然后只计算这些反向传播中难样本的梯度。这意味着忽略了在训练期间加强检测器的帮助不大的简单样本。

结论

在本文中，提出了一种基于CNNs的多任务级联联合人脸检测和对齐的框架。 实验结果表明，此方法始终在多个挑战性数据集（包括用于人脸检测的FDDB、WIDER FACE以及用于面部对齐的AFLW数据集）上面优于最先进的方法，同时对最小面部尺寸为20x20的640x480的VGA图像实现了实时性能。性能改进的三个主要贡献是仔细设计级联CNN架构，在线难样本挖掘策略和联合人脸对齐学习。

2、FaceNet：A Unified Embedding for Face Recognition and Clustering

概要

与其他的深度学习方法在人脸上的应用不同，FaceNet并没有用传统的softmax的方式去进行分类学习，然后抽取其中某一层作为特征，而是直接进行端对端学习一个从图像到欧式空间的编码方法，然后基于这个编码再做人脸识别、人脸验证和人脸聚类等。

FaceNet算法有如下要点：

去掉了最后的softmax，而是用元组计算距离的方式来进行模型的训练。使用这种方式学到的图像表示非常紧致，使用128位足矣。

元组的选择非常重要，选的好可以很快的收敛。

网络架构

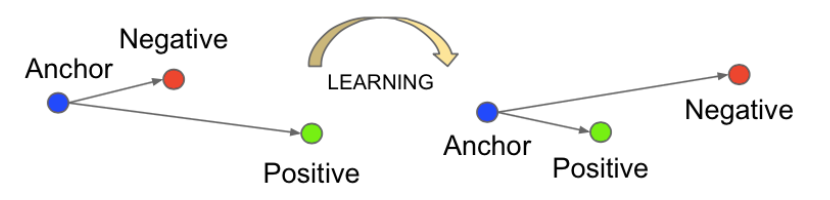
FaceNet使用深度卷积网络。讨论了两种不同的核心体系结构：Zeiler＆Fergus [22]型网络和最近的Inception [16]型网络。

给定模型细节，并将其视为黑匣子，方法的最重要部分在于整个系统的端到端学习。为此，采用三重损失，该三元组损失直接反映了在面部验证，识别和聚类中要实现的目标。争取一个嵌入f（x），从图像x到特征空间Rd中，以使相同身份的所有脸部之间的平方距离小，而与成像条件无关，而不同身份的一对图像之间的平方距离大。

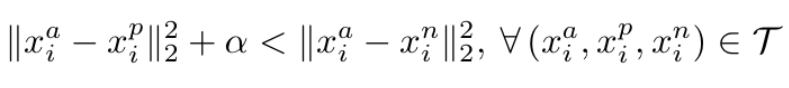


Deep Architecture就是卷积神经网络去掉sofmax后的结构，经过L2的归一化，然后得到特征表示，基于这个特征表示计算三元组损失。

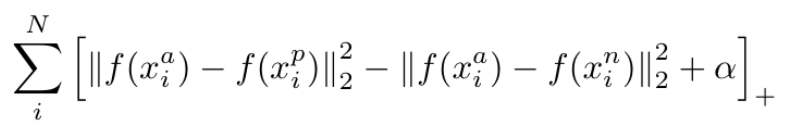
目标函数



所谓的三元组就是三个样例，如(anchor, pos, neg)，其中，x和p是同一类，x和n是不同类。那么学习的过程就是学到一种表示，对于尽可能多的三元组，使得anchor和pos的距离，小于anchor和neg的距离。即：



变换一下，得到目标函数：



目标函数的含义就是对于不满足条件的三元组，进行优化；对于满足条件的三元组，就pass先不管

三元组的选择

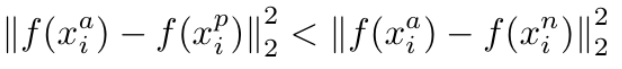
很少的数据就可以产生很多的三元组，如果三元组选的不得法，那么模型要很久很久才能收敛。因而，三元组的选择特别重要。

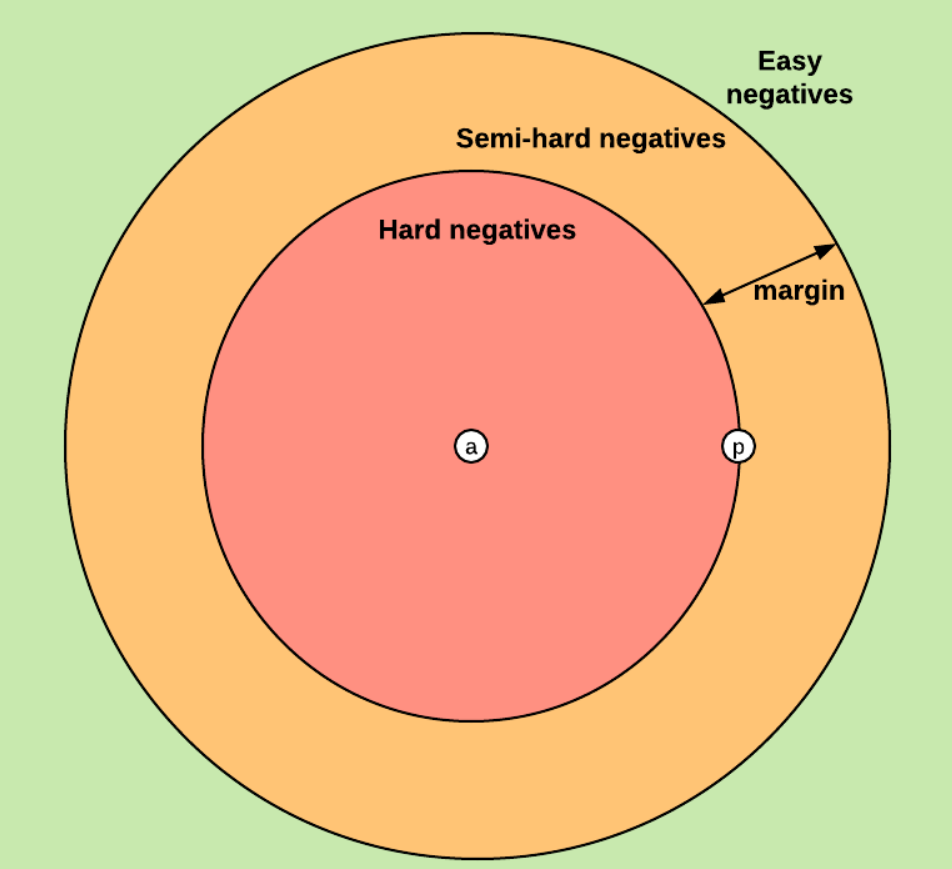
当然最暴力的方法就是对于每个样本，从所有样本中找出离他最近的反例和离它最远的正例，然后进行优化。这种方法有两个弊端：一是耗时，基本上选三元组要比训练还要耗时，二是容易受不好的数据的主导，导致得到的模型会很差。

所以，为了解决上述问题，论文中提出了两种策略：第一种是每N步线下在数据的子集上生成一些triplet，第二种是在线生成triplet，在每一个mini-batch中选择hard pos/neg 样例。

为了使mini-batch中生成的triplet合理，生成mini-batch的时候，保证每个mini-batch中每个人平均有40张图片。然后随机加一些反例进去。在生成triplet的时候，找出所有的anchor-pos对，然后对每个anchor-pos对找出其hard neg样本。这里，并不是严格的去找hard的anchor-pos对，找出所有的anchor-pos对训练的收敛速度也很快。

除了上述策略外，还可能会选择一些semi-hard的样例，所谓的semi-hard即不考虑alpha因素，即：





网络模型

论文使用了两种卷积模型：第一种是Zeiler&Fergus架构，22层，140M参数，1.6billion FLOPS(FLOPS是什么？)，称之为NN1。第二种是GoogleNet式的Inception模型。模型参数是第一个的20分之一，FLOPS是第一个的五分之一。

基于Inception模型，减小模型大小，形成两个小模型：

NNS1：26M参数，220M FLOPS。

NNS2：4.3M参数，20M FLOPS。

NN3与NN4和NN2结构一样，但输入变小了。

NN2原始输入：224×224

NN3输入：160×160

NN4输入：96×96

其中，NNS模型可以在手机上运行。

总结

三元组的目标函数并不是这篇论文首创，我在之前的一些Hash索引的论文中也见过相似的应用。可见，并不是所有的学习特征的模型都必须用softmax。用其他的效果也会好。

三元组比softmax的优势在于：softmax不直接，（三元组直接优化距离），因而性能也不好。softmax产生的特征表示向量都很大，一般超过1000维。

FaceNet并没有像DeepFace和DeepID那样需要对齐。

FaceNet得到最终表示后不用像DeepID那样需要再训练模型进行分类，直接计算距离就好了，简单而有效。

论文并未探讨二元对的有效性，直接使用的三元对。

3、Selective Refinement Network for High Performance Face Detection

概要

本文提出了一种新颖的单镜头人脸检测器，命名为选择性改进网络（SRN），它将新的两步分类和回归操作选择性地引入到基于锚的人脸检测器中，以同时减少误报并提高定位精度。特别是，SRN由两个模块组成：选择性两步分类（STC）模块和选择性两步回归（STR）模块。STC的目的是从低水平检测层中滤除大多数简单的负锚，以减少后续分类器的搜索空间，而STR则设计用于从高级检测层粗略调整锚的位置和大小，以便为随后的回归量。此外，还设计了一个感受野增强（RFE）块，以提供更多样化的接收场，这有助于更好地捕捉某些极端姿势的面部。因此，所提出的SRN检测器在所有广泛使用的面部检测基准上实现了最先进的性能，包括AFW，PASCAL面，FDDB和WIDER FACE数据集。

显著解决两个问题：1、减少False positive 2、捕获某些极端姿势的面部

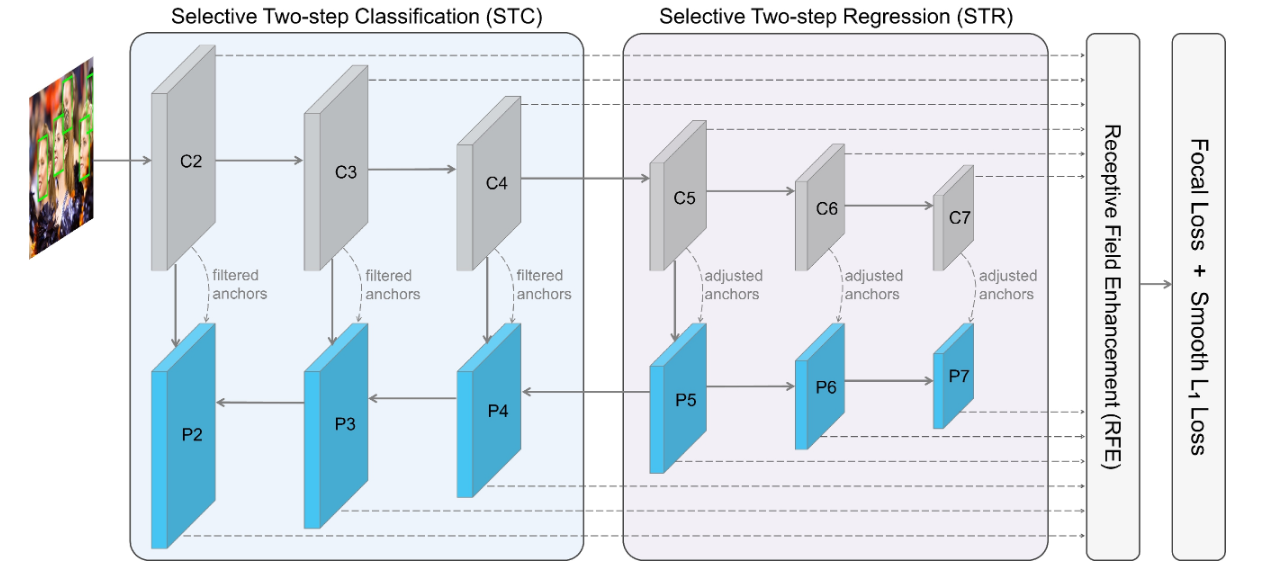
文章贡献有三：

•提出了一个STC模块，用于过滤掉来自低层的大多数简单负样本，以减少分类搜索空间。

•设计了一个STR模块，可以从高级层粗略调整锚点的位置和大小，为后续的回归量提供更好的初始化。

•引入RFE模块，为检测极端姿势面提供更多样化的感受野。

网络结构  
SRN的整体框架如图所示：



采用具有6级特征金字塔结构的ResNet-50作为SRN的骨干网络。从这四个残余块中提取的特征图分别表示为C2，C3，C4和C5。在C5之后，C6和C7只是由两个简单的下采样3x3卷积层提取。自下而上和自上而下通路之间的横向结构。P2，P3，P4和P5是从横向连接中提取的特征图，对应于分别具有相同空间大小的C2，C3，C4和C5，而P6和P7仅由两个3x3卷积层进行下采样。

STC模块选择C2，C3，C4，P2，P3和P4进行两步分类，而STR模块选择C5，C6，C7，P5，P6和P7进行两步回归,RFE模块负责丰富用于预测物体分类和位置的特征的接收领域。

在每个金字塔等级，使用两个特定的锚点尺度（即2S和2 2S，其中S代表每个金字塔等级的总步幅）和一个纵横比（即1.25）。总的来说，每个级别都有A=2锚点，它们覆盖了相对于网络输入图像的各级别的比例范围8-362像素。

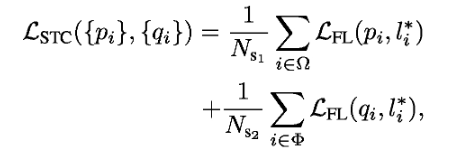
在深层架构的末尾添加混合损失，利用焦点损失和平滑L1损失的优点来推动模型更专注于更难的训练示例并学习更好的回归结果

STC

两步分类是一种通过两步网络架构实现的级联分类，其中第一步使用预设的负阈值0.99来减少大多数简单的负锚定以减少后续步骤的搜索空间。对于基于锚的人脸检测器，有必要在图像上平铺大量小锚以检测小脸，这导致正样本和负样本之间的极端类不平衡。例如，在具有1024x1024输入分辨率的SRN结构中，如果在每个锚点处平铺2个锚点，则样本总数将达到300k。其中，阳性样本的数量只有几十个或更少。为了减少分类器的搜索空间，必须进行两步分类以减少误报。

由于平铺在三个较高层（即P5，P6和P7）上的锚仅占11.1％并且相关特征更加充分。因此，在这三个更高的金字塔等级中，分类任务相对容易。因此，在三个较高的金字塔等级上应用两步分类是不必要的，并且如果应用，将导致计算成本的增加。相反，三个较低的金字塔等级（即P2，P3和P4）具有绝大多数样本（88.9％）并且缺乏足够的特征。迫切需要这些低金字塔等级进行两步分类，以减轻类不平衡问题并减少后续分类器的搜索空间。

STC的损失函数为：

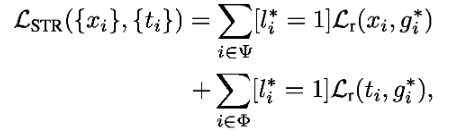


STR

在三个较低的金字塔等级中应用两步回归会损害性能，原因有两个：1）三个较低的金字塔等级与大量的小锚相关联，以检测小脸。这些小脸的特征是非常粗糙的特征表示，因此这些小锚非常难以执行两步回归; 2）在训练阶段，如果让网络过分关注低金字塔等级的困难回归任务，它将导致更大的回归损失并阻碍更重要的分类任务。

STR背后的动机是充分利用三个较高金字塔等级上的大面积的详细特征来回归更准确的边界框位置，并使三个较低的金字塔等级更加注重分类任务。这种分而治之的策略使整个框架更加高效。

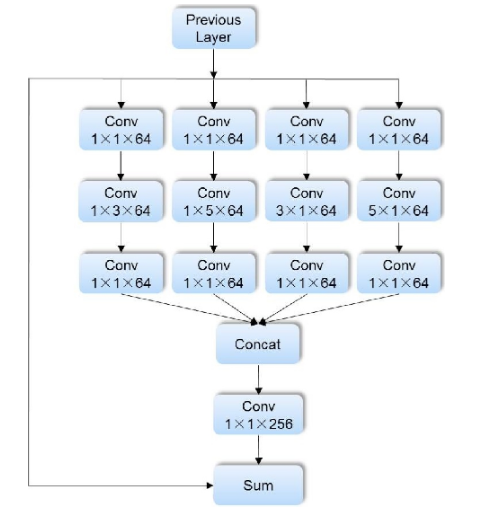
STR的损失函数也由两部分组成：



RFE

接收场的单一性影响具有不同纵横比的物体的检测。这个问题在人脸检测任务中似乎并不重要，因为在许多数据集中，人脸注释的纵横比约为1：1。尽管如此，统计数据显示，WIDER FACE训练集中有相当一部分面部长宽比大于2或小于0.5。因此，网络的接收场与面部的纵横比之间存在不匹配。

为了解决这个问题，提出了一个名为感知场增强（RFE）的模块，以在预测类和位置之前使特征的接收领域多样化。特别是，RFE模块取代了类子网中的两个卷积层和RetinaNet的子网。 RFE的结构如下图所示。RFE模块采用四分支结构，其灵感来自Inception模块。具体来说，首先，使用1x1卷积层将通道数减少到前一层的四分之一。其次，使用1xk和kx1（k=3和5）卷积层来提供矩形接收字段。通过另一个1x1卷积层，将来自四个分支的特征映射连接在一起。



1. 源码运行

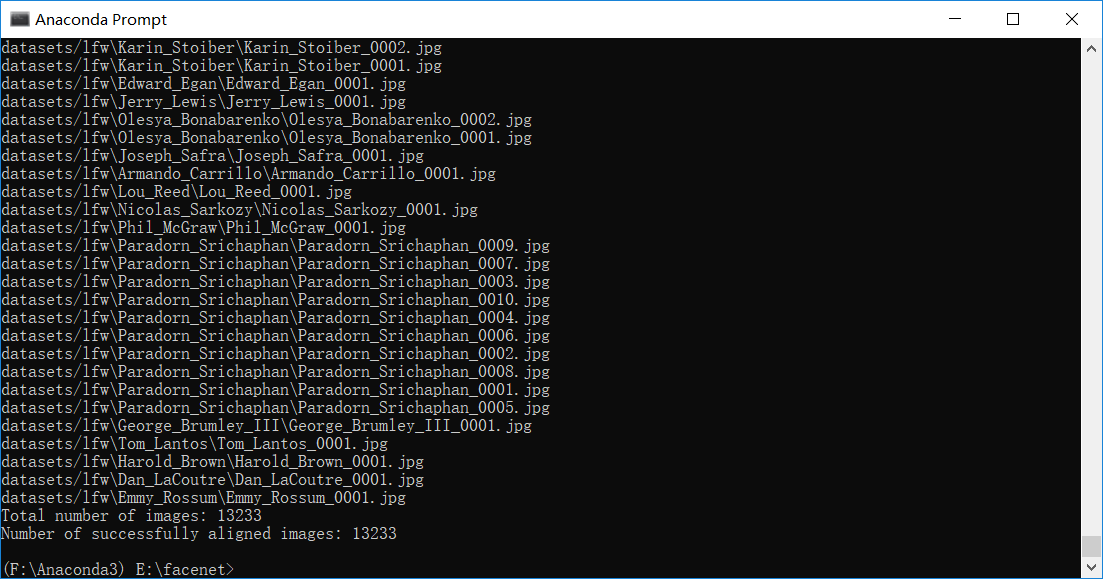
1、首先下载FaceNet源码到本地，然后配置好相应的环境

2、下载LFW数据集

LFW 是评估人脸识别算法效果的公开测试数据集，共有13233张图片，属于 5749 个不同人，其中图片的大小为250×250。

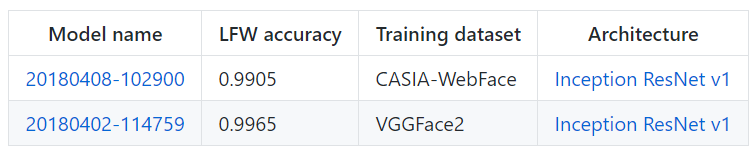
对LFW图片预处理

LFW图片原图尺寸为 250\*250，需要修改图片尺寸，使其大小符合预训练模型的输入尺寸，即160\*160。在src/align/align\_dataset\_mtcnn.py文件里，采用MTCNN人脸检测算法对人脸进行检测，进一步人脸对齐，然后再把人脸图片尺寸修改为160×160的尺寸，保存到datasets/ lfw\_160目录下。



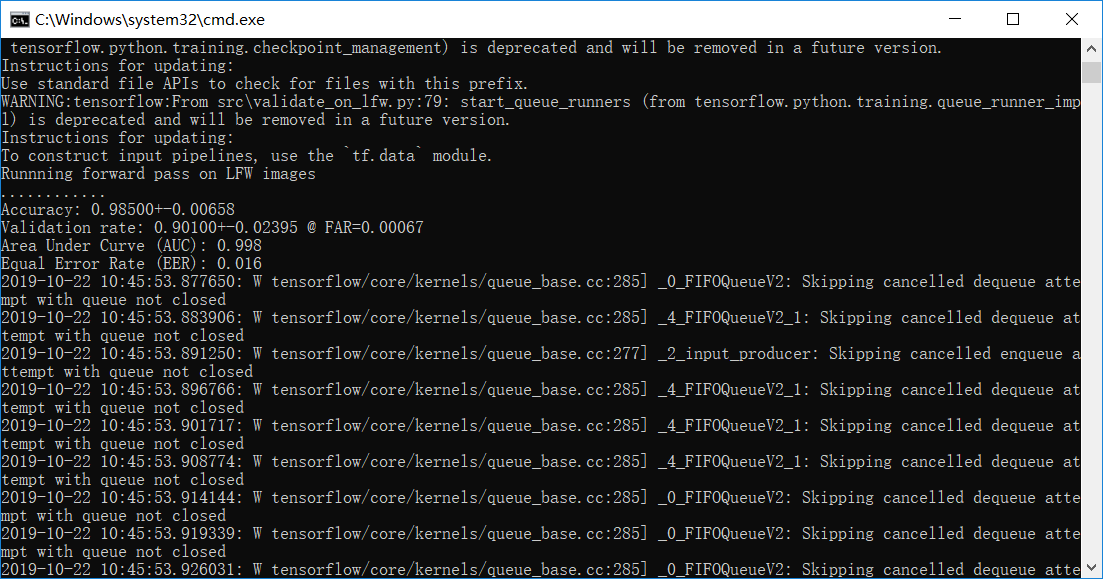
3、下载Google预训练的网络模型

官网提供了两个模型，这两个模型都是基于Inception ResNet V1网络架构下进行识别的，两个模型基于不同的训练集进行训练，这里我下载的是VGGFace2数据集的模型



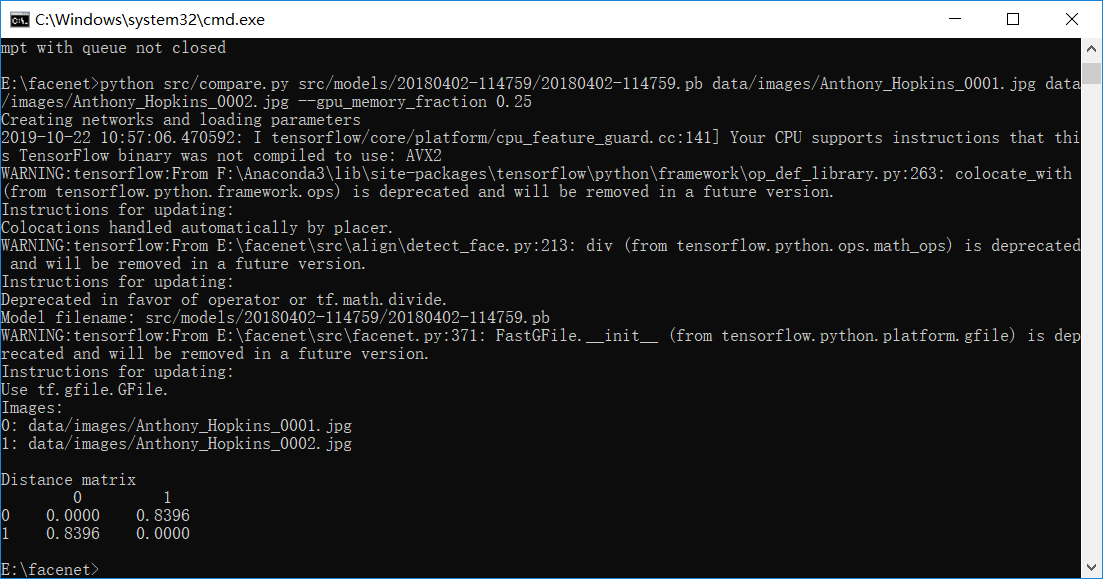
4、预训练模型准确率测试

可以看到，准确度达到了0.98500+-0.00658



5、比较两张图片的欧氏距离

使用欧氏距离来衡量两张人脸的相似程度，用来判别这两张图片是否为同一个人，；两张人脸图片越相似，空间距离越小，差别越大，则空间距离越大



6、制作自己的人脸数据库

6.1准备数据集

通过百度图片爬取一些明星图片，然后上传了我自己的一些图片，构成一个小的数据集

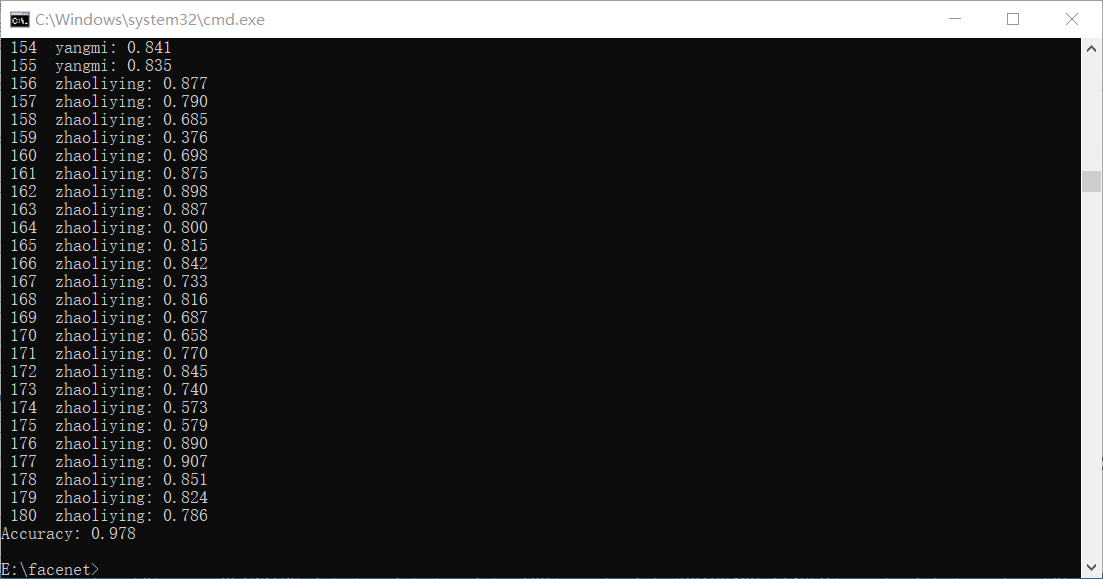


再通过align\_dataset\_mtcnn.py进行人脸检测，并把人脸图片尺寸设定为 160\*160，数据集里面的人脸图片数据集并不是可靠的，从网上下载的图片可能有些图片质量较差，或者有多张人脸的，然后导致人脸检测失败，所以还需要检测数据集里面的人脸图片，把不合格的图片给删掉。

6.2制作人脸识别库

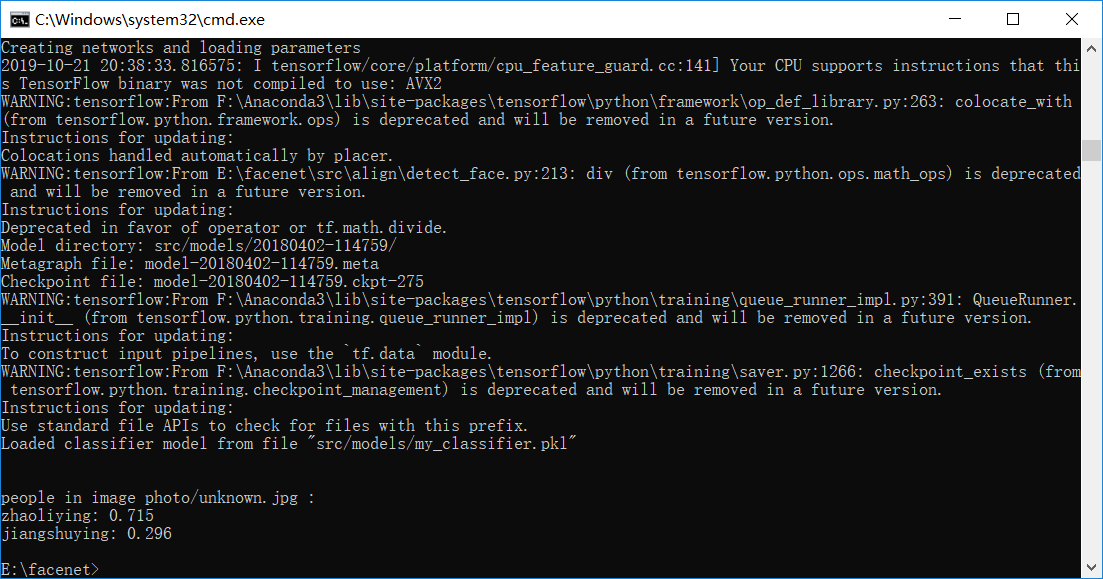
训练一个人脸识别库，生成.pkl文件，在生成.pkl文件的时候用到calssifier.py文件，里面用SVM来训练一个分类器。整体流程大致是：CNN forward 输出后经L2传入Embedding层，得到embedding ouput的特征进行传给SVM classifier来训练一个分类器。然后把训练好的分类器保存为pickle文件。在执行指令读取分类器的时候加载SVM分类器模型，自己所用的测试图片数据就会SVM分类器模型中的类别做对比判断。.pkl文件保存的参数包括模型和类别，是可以直接读出来的，代码里面SVM有两种模式：TRAIN（用来训练SVM模型）和CLASSIFY（用来加载SVM模型）。看代码会有更好的理解

然后使用.pkl文件来验证图片，看看效果：

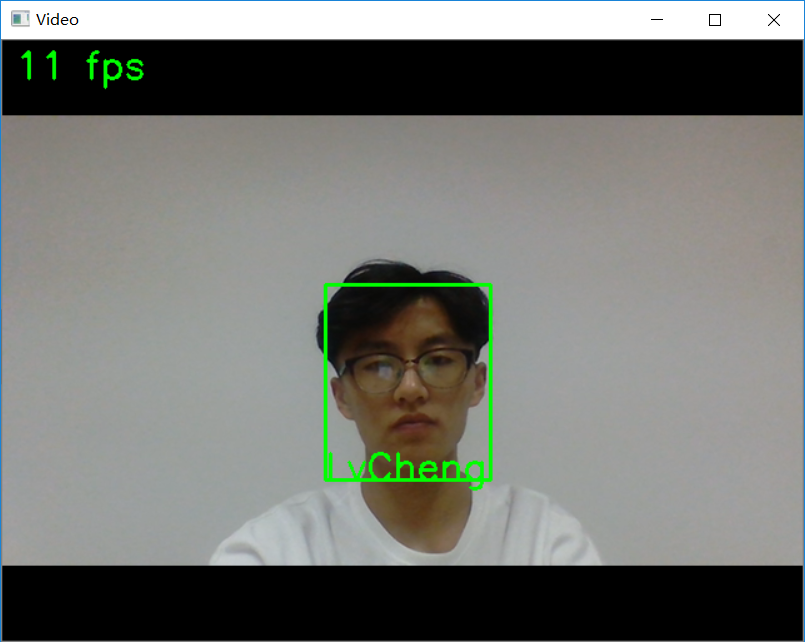


7、验证图片

网上下载一张图片，然后用predict.py代码来测试，结果显示能够正确识别



8、使用摄像头实时识别



三、预计进展

对代码理解的更加透彻，并且争取能够在此基础上加上一些自己的东西，人脸检测是我比较喜欢的一个方向，预计中期报告之后读更多的相关论文，争取能做一些改进，对模型加深印象，感觉目前还是了解的比较少，并没有深层次的了解网络架构，而且目前只读了两三篇论文，应该更深入地去学习人脸检测方面的东西，争取能自己实现一些小的demo。