论文：Joint Face Detection and –Alignment using Multi-task Cascaded Convolutional Networks

一、创新点

本文提出一个新的框架使用统一的级联 CNN 通过多任务学习来集成人脸检测和特征点定位两个任务。

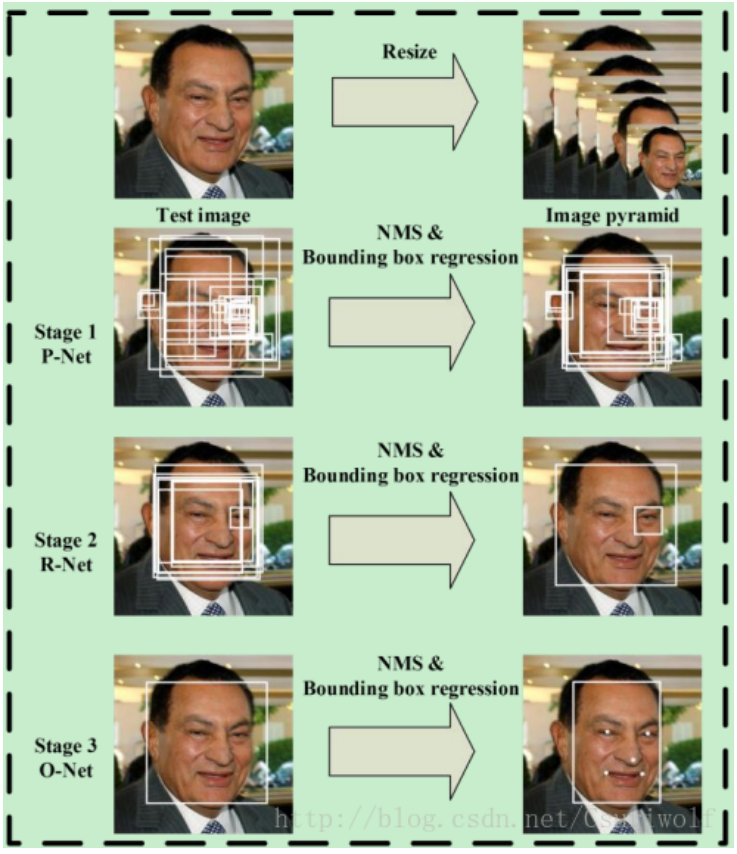
提出的 CNN 包括三个阶段：

（1）通过一个 shallow CNN 迅速产生 candidate windows.

（2）通过一个稍复杂的 CNN 丢弃大部分没有人脸的 windows.

（3）使用一个更强大的 CNN 精炼结果，同时显示面部特征点定位。设计 lightweight CNN 可以提高实时性能。

二、Approach



整体流程如上图所示，输入图片，首先 resize 到不同的尺寸建立一个 image pyramid，作为接下来三层级联框架的输入

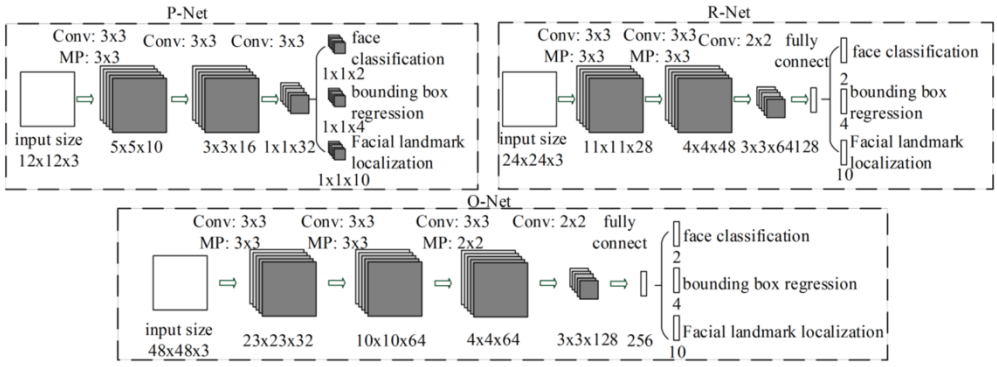


Fig. 2. The architectures of P-Net, R-Net, and O-Net, where "MP" means max pooling and "Conv" means convolution. The step size in convolution and pooling is 1 and 2, respectively.

step1: 采用全卷积神经网络，即 Proposal Network(P-Net),去获得候选窗体和它们的边界盒回归向量(bounding box regression vectors).然后使用估算的边界盒回归向量来校准候选窗体.然后利用 NMS(non-maximum suppression)方法去除高度重叠的窗体。

Step2: 所有候选窗体被送入另一个 CNN: Refine Network(R-Net)，更进一步地去除大量的错误候选窗体，再使用 bounding box regression 进行校准和 NMS 法。

Step3: 该阶段类似于第二阶段，但该阶段目的是描述脸部更多细节，显示五个脸部特征点位置。

三、Training

1、Face classification

对每个样本 使用 cross-entropy loss(交叉熵损失函数)：

$$L_i^{det} = -(y_i^{det}\log(p_i) + (1 - y_i^{det})(1 - \log(p_i)))$$

其中 Pi 表示样本是人脸的概率,y_i^{det} 表示 ground-truth label(样本是否为人脸的真实取值).

2、Bounding box regression

对每个候选窗体，得到窗体和最近的 ground truth 之间的差距(CNN 得出的候选窗口和真实窗口的差别)。回归问题，对每个样本 应用 Euclidean loss(欧几里得损失函数):

$$L_i^{box} = \left\| \hat{y}_i^{box} - y_i^{box} \right\|_2^2$$

其中 $\hat{y}_i^{box}$ regression target 来自网络，$y_i^{box} \in \mathbb{R}^4$ 是四维 ground-truth 坐标，包括左上坐标、高和宽。

3、Facial landmark localization

类似于包围盒回归任务，脸部特征点检测也是回归问题，需最小化 Euclidean loss:

$$L_i^{landmark} = \left\| \hat{y}_i^{landmark} - y_i^{landmark} \right\|_2^2$$

其中$\hat{y}_i^{landmark}$是网络得到的面部特征点坐标，$y_i^{landmark} \in \mathbb{R}^{10}$是 10 维 ground-truth 坐标，包括左眼、右眼、鼻子、左嘴角和右嘴角。

4、Multi-source training

对于级联架构的 CNN，每一层有不同的任务，在学习过程中也有不同类型的训练图片，并不是所有的损失函数都同时使用，这可以通过 a sample type indicator 实现。

整体学习目标如下：

$$\min \sum_{i=1}^{N} \sum_{j \in \{det, box, landmark\}} \alpha_j \beta_i^j L_i^j$$

其中 N 是样本数量,αj 代表任务重要性,βij 是 sample type indicator。运用 stochastic gradient descent(随机梯度下降)训练 CNNs。

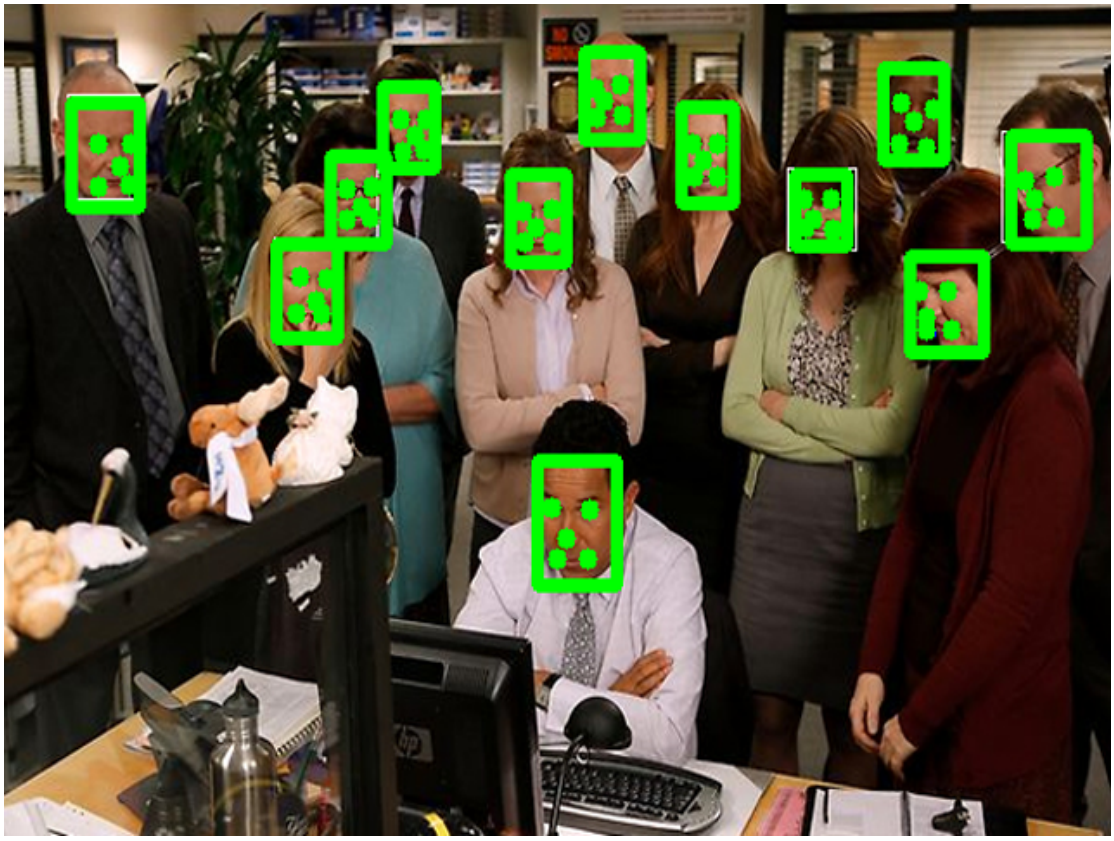
四、Experiments

在 github 上找到了作者的代码，搭建好作者给出的环境，实现了人脸检测。

实验环境：python3.5、tensorflow==0.12、opencv-python

具体步骤：代码中核心的主要是 detect_face.py 里面定义的三个网络层 PNet、RNet、ONet,也就是作者在论文提到的三层级联框架的输入。最后输出人脸位置以及人脸关键点位置。

实验结果



作者在论文中给出：
使用三个数据集进行训练对比结果：FDDB(Face Detection Set and Benchmark), WINDER FACE,AFLW(Annotated Facial Landmarks in the wild benchma.
本文的人脸检测(recall and precision)和人脸特征点定位(mean error)的效果都非常好。关键是这个算法速度很快，在 2.6GHZ 的 CPU 上达到 16fps，在 GPU 上可达到 99fps。

五、Conclusion
本文提出一种基于 multi-task cascaded CNNs 的框架进行联合人脸检测和特征点检测。实验结果表明该方法在一些比赛测试集上性能一贯优于当前先进算法且速度快，保持实时性能。未来可以研究人脸检测和其他人脸分析任务内在的联系，提升性能。

改进点
1、优化滤波器
2、优化网络结构