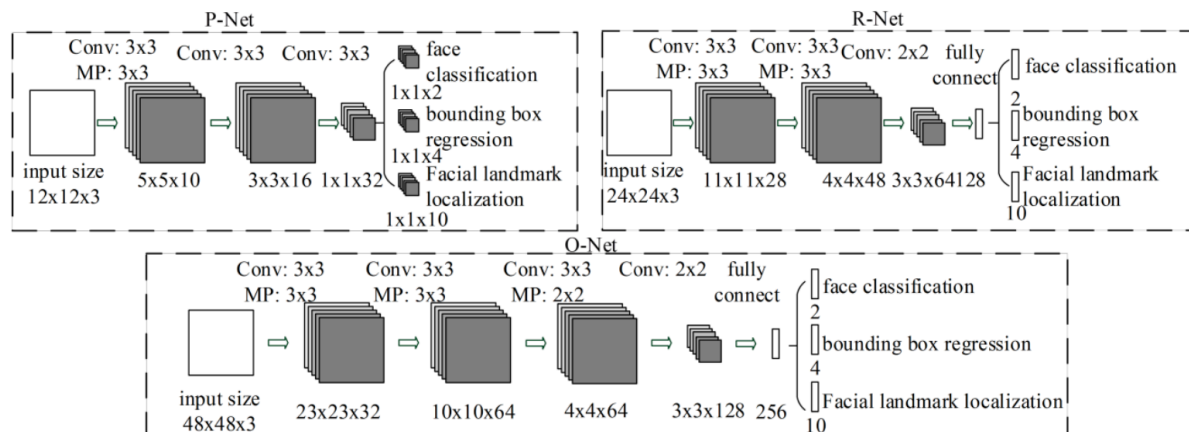


MTCNN训练



MTCNN 训练复杂的地方之一在于训练数据的准备，后一阶段训练数据生成依赖前一阶段，因此整个网络的训练时串行的，必须训练好前一阶段网络之后再训练后一阶段网络。而且RNet, ONet训练数据生成都依赖网络前向运行，更加更加了耗时。MTCNN 中训练数据分为三种，negative, part, positive, negative 样本和原始标注的 IoU 小于0.3, part 样本和原始标注的 IoU 在 0.4, 0.65之间, positive 样本和原始标注的 IoU 在 0.65 以上。使用 negative 和 positive 来训练人脸非人脸分类分支，使用 part 和 positive 来训练人脸框回归分支。如果有关键点数据还有一类landmark数据，用来训练关键点分支。

生成PNet数据

PNet数据直接从原图中 crop 出来的，确保每张图片crop出来的positive, negative, part样本数量在一定比例范围内。防止出现样本不均衡的问题。

训练PNet

PNet 的样本输入很小，因此可以设置很大的batch size。PNet 的作用时滑窗在图像金字塔上生成 proposal以便给后面的网络，因此准确率不会很高，我自己训练可以达到90%的accuracy。

生成RNet数据

训练好 PNet 后就可以用PNet来生成proposal，这些proposal大多是positive,part样本，可以在在原图上裁剪出一定negative样本，构建negative, part, positive成一定比例额的RNet训练数据。

训练RNet

生成RNet训练数据就可以训练RNet, RNet的训练样本经过PNet筛选，质量更高，因此可以得到更高的准确率，我自己训练达到了

生成ONet数据

训练好PNet和RNet后就可以生成ONet训练数据，ONet训练数据生成和RNet类似，只不过要经过PNet, RNet两个步骤。

训练ONet

ONet 训练过程和PNet, RNet类似。ONet训练数据经过两个阶段筛选可以得到更高的准确度。

MTCNN 推理

训练好PNet, RNet, ONet之后整个MTCNN都训练完了，可以用来推理。其实前面训练数据生成也用到了部分推理流程，只不过不是很完整。

最重要的是理解PNet的作用，PNet训练的时候输入是12*12的图像，但是推理的时候输入的是原图，最后输出层是一个feature map。PNet有一个stride=2的max pooling，其他层stride=1, 相当于一个12*12滑窗通过stride=2的步长在图像上滑动，feature map上的每个cell对应不同位置的人脸。feature map的每个cell感受野是12*12, 同时在推理的时候会构建图像金字塔对每个金字塔图像进行推理，这样可以保证固定大小的感受野检测不同大小的人脸。

PNet的输出对应着图像上的位置，这个位置裁剪出proposal经过nms筛选缩放到24*24送入RNet，同样RNet预测的位置缩放到48*48送入ONet，最后作为输出。

MTCNN 优点和缺点

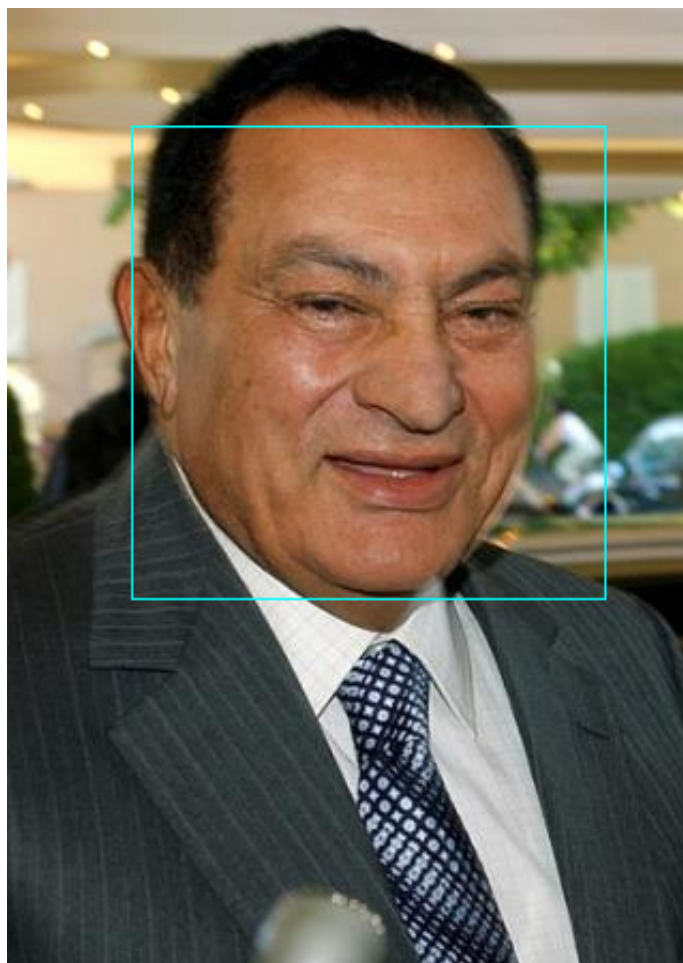
MTCNN 网络简单速度很快，结合了很多任务有助于提升网络的性能。借鉴了很多传统人脸检测的方法，是传统方法到深度学习的过渡作品。经过一些剪枝压缩优化，MTCNN甚至可以在手机上实时跑起来。

MTCNN 的问题主要在于PNet要经过多个金字塔图像的推理，速度很慢，金字塔越多速度越慢。而图像金字塔主要是为了检测多个尺度的人脸，而后期的算法都通过 FPN 来提取特征金字塔，对应不同的感受野，同时结合不同尺度的 anchor, 可以通过一次推理来检测多个尺度的人脸。

此外MTCNN 推理时候的 proposal 生成是通过 阈值来卡的，人脸越多 proposal 越多，会导致后面的RNet, ONet运行速度越慢。而单阶段基于 anchor 的检测网络，不会有这个问题，检测速度是和 anchor 数量有关，一般是固定的。

MTCNN测试

下面的测试结果是1000个循环的时间，减少误差，分别测试了下面这三张图片，分别有1个人脸，3个人脸，5个人脸。输入前都缩放到宽高：640x480



测试的时候三个网路的阈值分别设置为[0.6, 0.7, 0.8],

运行的时候每个子网络proposal个数

人脸个数	pnet	rnet	One
1	24	1	1
3	91	3	3
5	93	8	5

人脸数目，设备类型和检测时间的关系

人脸个数\设备	CPU	GPU
1	16.30s	13.22s
3	29.59s	23.75s
5	31.16s	26.37s

从上面两个图可以看出来，图中人脸越多，每个网络往后输出的人脸个数越多，会加重后面网络的负担，运行时间会越长。这就是MTCNN的缺点。

在GPU上运行比CPU上运行速度稍微快点。

我们可以通过调整每个阶段的检测阈值来控制proposal的数目，阈值越高，速度会越快，但是可能有漏检的情况。阈值越低速度越慢，但是漏检的情况更少。