

MTCNN face detection implementation

方桂安, 刘梦莎, 张焯霖, 刘玥, 唐迅

摘要—本次作业的目标是复现 MTCNN 代码，实现人脸检测及人脸关键点估计功能。我们通过课上内容与查阅相关文献，了解到 MTCNN 是一种用于人脸检测和人脸关键点估计的深度学习模型。它通过使用多个卷积神经网络来实现这些功能，并能够在实时应用中取得较高的准确率。除了理解、改善并整合课上老师提供的代码，我们还实现了 checkpoint 微调、学习率衰减、摄像头检测和一键式训练、测试等新功能，并和目前最先进的人脸识别模型进行了对比实验。

关键词—人脸检测, 人脸对齐, 级联卷积神经网络

I. 概述

人脸检测 (face detection) 是一种在任意数字图像中找到人脸的位置和大小的计算机技术。它可以检测出面部特征，并忽略诸如建筑物、树木和身体等其他任何东西。有时候，人脸检测也负责找到面部的细微特征，如眼睛、鼻子、嘴巴等的精细位置。

在说到人脸检测我们首先会想到利用 Harr 特征提取和 Adaboost 分类器进行人脸检测，其检测效果也是不错的，但是目前人脸检测的应用场景逐渐从室内演变到室外，从单一限定场景发展到广场、车站、地铁口等场景，人脸检测面临的要求越来越高，比如：人脸尺度多变、数量冗大、姿势多样包括俯拍人脸、戴帽子口罩等的遮挡、表情夸张、化妆伪装、光照条件恶劣、分辨率低甚至连肉眼都较难区分等。在这样复杂的环境下基于 Harr 特征的人脸检测表现的不尽人意。随着深度学习的发展，基于深度学习的人脸检测技术取得了巨大的成功，在这篇报告中我们将会介绍 MTCNN 算法，它是基于卷积神经网络的一种高精度的实时人脸检测和对齐技术。搭建人脸识别系统的第一步就是人脸检测，也就是在图片中找到人脸的位置。在这个过程中输入的是一张含有人脸的图像，输出的是所有人脸的矩形框。一般来说，人脸检测应该能够检测出图像中的所有人脸，

不能有漏检，更不能有错检。获得人脸之后，第二步我们要做的工作就是人脸对齐，由于原始图像中的人脸可能存在姿态、位置上的差异，为了之后的统一处理，我们要把人脸“摆正”。为此，需要检测人脸中的关键点，比如眼睛的位置、鼻子的位置、嘴巴的位置、脸的轮廓点等。根据这些关键点可以使用仿射变换将人脸统一校准，以消除姿势不同带来的误差。

MTCNN 算法是一种基于深度学习的人脸检测和人脸对齐方法，它可以同时完成人脸检测和人脸对齐的任务，相比于传统的算法，它的性能更好，检测速度更快。

MTCNN 算法包含三个子网络：Proposal Network、Refine Network、Output Network，这三个网络对人脸的处理依次从粗到细。

在使用这三个子网络之前，需要使用图像金字塔将原始图像缩放到不同的尺度，然后将不同尺度的图像送入这三个子网络中进行训练，目的是为了可以检测到不同大小的人脸，从而实现多尺度目标检测。下面，我们将详细介绍人脸检测和 MTCNN 算法。

II. 人脸检测

人脸检测的目标是找出图像中所有的人脸对应的位置，算法的输出是人脸外接矩形在图像中的坐标，可能还包括姿态如倾斜角度等信息。图 1 是一张图像的人脸检测结果：虽然人脸的结构是确定的，由眉毛、眼睛、

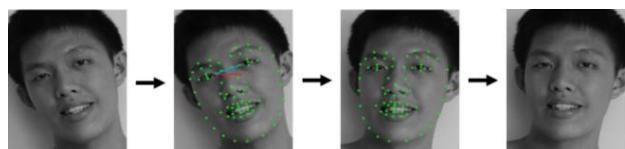


Fig. 1. 人脸检测实例

鼻子和嘴等部位组成，近似是一个刚体，但由于姿态和表情的变化，不同人的外观差异，光照，遮挡的影响，准确的检测处于各种条件下的人脸是一件相对困难的事情。

方桂安, 20354027, (e-mail: fanggan@mail2.sysu.edu.cn)。

刘梦莎, 20354091, (e-mail: liumsh6@mail2.sysu.edu.cn)。

张焯霖, 20354156, (e-mail: zhangzhlin8@mail2.sysu.edu.cn)。

唐迅, 20354121, (e-mail: tangx66@mail2.sysu.edu.cn)。

刘玥, 20354229, (e-mail: liuy2236@mail2.sysu.edu.cn)。

人脸检测算法要解决以下几个核心问题：

- 1) 人脸可能出现在图像中的任何一个位置
- 2) 人脸可能有不同的大小
- 3) 人脸在图像中可能有不同的视角和姿态
- 4) 人脸可能部分被遮挡

评价一个人脸检测算法好坏的指标是检测率和误报率。我们将检测率定义为：

$$\text{检测率} = \frac{\text{检测出的人脸数}}{\text{图像中所有人脸数}}$$

误报率定义为：

$$\text{误报率} = \frac{\text{误报个数}}{\text{图像中所有非人脸扫描窗口数}}$$

算法要在检测率和误报率之间做平衡，理想的情况是有高检测率，低误报率。

经典的人脸检测算法流程是这样的：用大量的人脸和非人脸样本图像进行训练，得到一个解决 2 类分类问题的分类器，也称为人脸检测模板。这个分类器接受固定大小的输入图片，判断这个输入图片是否为人脸，即解决是和否的问题。

由于人脸可能出现在图像的任何位置，在检测时用固定大小的窗口对图像从上到下、从左到右扫描，判断窗口里的子图像是否为人脸，这称为滑动窗口技术 (sliding window)。为了检测不同大小的人脸，还需要对图像进行放大或者缩小构造图像金字塔，对每张缩放后的图像都用上面的方法进行扫描。由于采用了滑动窗口扫描技术，并且要对图像进行反复缩放然后扫描，因此整个检测过程会非常耗时。

由于一个人脸附件可能会检测出多个候选位置框，还需要将检测结果进行合并去重，这称为非极大值抑制 (NMS)。

A. 典型应用

人脸检测是机器视觉领域被深入研究的经典问题，在安防监控、人证比对、人机交互、社交等领域都有重要的应用价值。数码相机、智能手机等端上的设备已经大量使用人脸检测技术实现成像时对人脸的对焦、图集整理分类等功能，各种虚拟美颜相机也需要人脸检测技术定位人脸，然后才能根据人脸对齐的技术确定人脸皮肤、五官的范围然后进行美颜。在人脸识别的流程中，人脸检测是整个人脸识别算法的第一步。

B. 早期算法

早期的人脸检测算法使用了模板匹配技术，即用一个人脸模板图像与被检测图像中的各个位置进行匹配，确定这个位置处是否有人脸；此后机器学习算法被用于该问题，包括神经网络，支持向量机等。以上都是针对图像中某个区域进行人脸-非人脸二分类的判别。

C. AdaBoost 框架

用级联 AdaBoost 分类器进行目标检测的思想是：用多个 AdaBoost 分类器合作完成对候选框的分类，这些分类器组成一个流水线，对滑动窗口中的候选框图像进行判定，确定它是人脸还是非人脸。

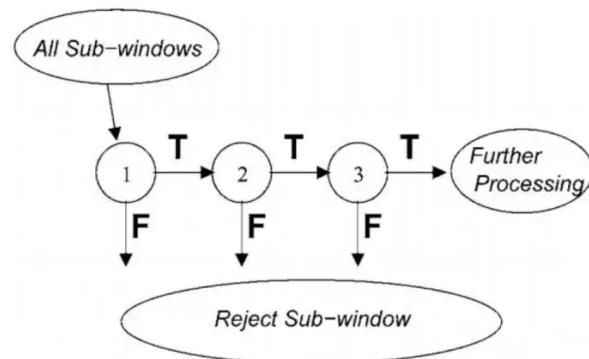


Fig. 2. 分类器级联系示意图

在这一系列 AdaBoost 分类器中，前面的强分类器设计很简单，包含的弱分类器很少，可以快速排除掉大量的不是人脸的窗口，但也可能会把一些不是人脸的图像判定为人脸。如果一个候选框通过了第一级分类器的筛选即被判定为人脸，则送入下一级分类器中进行判定，以此类推。如果一个待检测窗口通过了所有的强分类器，则认为是人脸，否则是非人脸。

这种思想的精髓在于用简单的强分类器在初期快速排除掉大量的非人脸窗口，同时保证高的召回率，使得最终能通过所有级强分类器的样本数很少。这样做的依据是在待检测图像中，绝大部分都不是人脸而是背景，即人脸是一个稀疏事件，如果能快速的把非人脸样本排除掉，则能大大提高目标检测的效率。

D. 深度学习算法

卷积神经网络在图像分类问题上取得成功之后很快被用于人脸检测问题，在精度上大幅度超越之前的 AdaBoost 框架，当前已经有一些高精度、高效的算法。直接用滑动窗口加卷积网络对窗口图像进行分类的方

案计算量太大很难达到实时，使用卷积网络进行人脸检测的方法采用各种手段解决或者避免这个问题。

下面，我们将介绍我们实验采用的算法，MTCNN。

III. MTCNN

MTCNN 模型利用多级联的结构，从粗到细预测人脸以及相应特征坐标位置，能够适用于各种自然条件下复杂的人脸场景检测，可以实现人脸检测和 5 个特征点的标定。主要包括三个网络子结构：P-Net (proposal networks)、R-Net (refine networks)、O-Net (output networks)。

A. 模型流程

该算法模型流程包括四部分（图像金字塔、P-Net、R-Net、O-Net）

1) 图像金字塔：为了检测到不同 size 的人脸，在进入 P-Net 之前，我们应该对图像进行金字塔操作。首先，根据设定的 min_face_size 尺寸，将 img 按照一定的尺寸缩小，每次将 img 缩小到前级 img 面积的一半，形成 scales 列表，直至边长小于 min_face_size，此时得到不同尺寸的输入图像。

```
def resize_image(self, img, scale):
    height, width, channels = img.shape
    new_height = int(height * scale)      # resized new height
    new_width = int(width * scale)        # resized new width
    new_dim = (new_width, new_height)
    img_resized = cv2.resize(img, new_dim, interpolation=cv2.INTER_LINEAR)  # resized image
    return img_resized
```

Fig. 3. 缩放图像形成金字塔

2) P-Net: 根据上述步骤得到的不同尺寸的图像，输入到 P-Net 网络中。如图 4 所示为 P-Net 网络结构

由图 4 可知，该层网络 anchor 大小为 12*12 (可更改)，代表 12*12 区域，经过 P-Net 全卷积层后，变成 1 (输入不同，则输出也不同。假设输出为 w*h，则输出的每个点都对应原图像中一个 12*12 的区域)

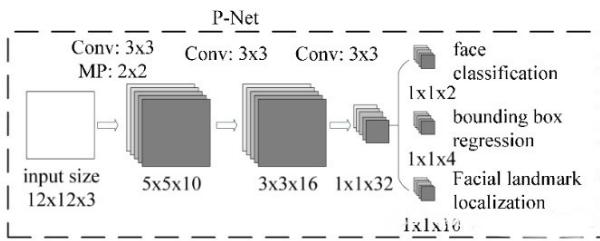


Fig. 4. P-Net 网络结构示意图

a. 将不同尺寸的金字塔图像输入到 p-net 中，最终得到 prob1 与 conv4-2。prob1 中包含 box 位置信息及其置信度，conv4-2 中包含 box 的回归系数信息。

b. 利用 a 中的 prob1 与 conv4-2 生成 box，设置阈值为 0.6 (初筛，阈值应偏小)，得到一系列点，影射回原图像，以此点为左上角，向右向下各扩展 12 pixel，得到 12*12 的矩形框。

c. 接下来对一帧图像上检测到的所有 12*12 矩形框进行 nms 运算。

d. 最后得到的所有 box 会放置在一个 number*9 的数组里，number 表示 box 的数量，9 代表 box 的坐标信息、score、坐标回归信息 [x1, y1, x2, y2, score, reg_x1, reg_y1, reg_x2, reg_y2]，利用 reg* 系列 (对应坐标的线性回归参数) 可对 box 进行坐标修正，修正过程可表示为：

$$\begin{aligned} new_x1 &= x1 + reg_x1 * width_{ofbox} \\ new_y1 &= y1 + reg_y1 * height_{ofbox} \\ new_x2 &= x2 + reg_x2 * width_{ofbox} \\ new_y2 &= y2 + reg_y2 * height_{ofbox} \end{aligned} \quad (1)$$

e. 目标框修正之后，先 rec2square、再 pad。rec2square 是将修正后不规则的框调整为正方形，pad 的目标是将超出原 img 范围的部分填充为 0，大小比例不变。

3) R-Net: 将 P-Net 最后输出的所有 box，resize 到 24*24 后输入 R-Net。经过 R-Net 后，输出与 P-Net 类似，prob1: box 坐标信息与置信度与 conv5-2 的回归系数信息。根据所得的置信度信息与该层阈值对比，小于阈值的直接 drop 掉，大于阈值的留下，经过 nms、再利用回归系数信息进行精修、rec2square、pad。

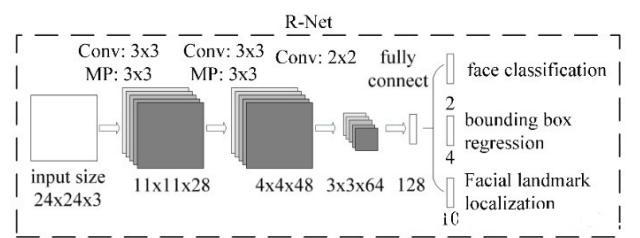


Fig. 5. R-Net 网络结构示意图

4) O-Net: 将 R-Net 最后输出的所有 box，resize 到 48*48 后输入 O-Net。经过 O-Net 后，输出 prob1: box 坐标信息与置信度、conv6-2 的回归系数信息、以及 conv6-3 的关键点坐标信息。conv6-3 是 number*10 的

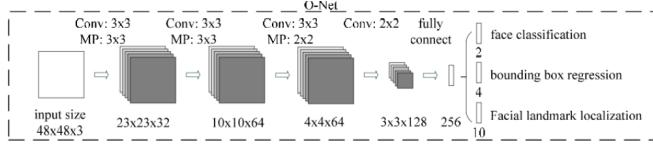


Fig. 6. O-Net 网络结构示意图

二维数组，number 代表 box 的数量，10 则包含了 5 个关键点信息的 x、y 坐标信息：[Rx1, Rx2, Rx3, Rx4, Rx5, Ry1, Ry2, Ry3, Ry4, Ry5]，此时的坐标为目标框内部的比例，最后影射回原 img 得到真实的坐标。

根据 prob1 置信度信息与该层阈值对比，小于阈值的直接 drop 掉，大于阈值的留下，再利用回归系数信息进行精修，最后再进行一次 nms。

最后，输出一副包含人脸框与人脸关键点的检测图像。

B. Loss 损失函数

训练中需要最小化的损失函数来自 3 方面：

face/non-face classification + bounding box regression + facial landmark localization，综合在一起，表示如下：

$$\text{Loss} = \min \sum_{i=1}^N \sum_{j \in \{\text{det, box, landmark}\}} \alpha_j \beta_i^j L_i^j \quad (2)$$

α_j 代表对应任务的重要性。

P-Net: $\alpha_{\text{det}} = 1, \alpha_{\text{box}} = 0.5, \alpha_{\text{landmark}} = 0.5$

R-Net: $\alpha_{\text{det}} = 1, \alpha_{\text{box}} = 0.5, \alpha_{\text{landmark}} = 0.5$

O-Net: $\alpha_{\text{det}} = 1, \alpha_{\text{box}} = 0.5, \alpha_{\text{landmark}} = 1$

β_j 代表样本类型。

边训练边选择出 hard sample，只有 hard samples 才进反向传播，其他样本不进行反向传播。具体做法：对每个小批量里所有样本计算 loss，对 loss 进行降序，前 70% samples 做为 hard samples 进行反向传播。

IV. 数据集

训练 MTCNN 的数据集论文里使用的是 WIDER FACE 和 CelebA，其中 WIDER FACE 用作回归人脸分类和 box，CelebA 用来回归关键点（landmark）。本次实验我们使用的数据集为 Wider Face 数据集和 CNN_FacePoint 数据集。

A. Wider Face 数据集

WIDE FACE 数据集是一个人脸检测基准数据集，该数据集的图片来源是 WIDES 数据集，从中挑选出了 32,203 图片并进行了人脸标注，总共标注了 393,703 个人脸数据。并且对于每张人脸都附带有更加详细的信息，包括 blur (模糊程度), expression (表情), illumination (光照), occlusion (遮挡), pose (姿态)。如 7 所示。



Fig. 7. wider face 数据集

在数据集中，根据事件场景的类型分为了 61 个类。接着根据每个类别按照 40% / 10% / 50% 的比例划分到训练集，验证集以及测试集中。

B. CNN_FacePoint

CNN_FacePoint 是由论文 Deep Convolutional Network Cascade for Facial Point Detection 提出的人脸关键点检测数据集。

训练集包含 5,590 张 LFW 图像和从网上下载的 7,876 张其他图像。数据集文本文件的每一行都以图像名称开头，然后是人脸检测器返回的人脸边界框的边界位置，最后是五个人的位置面部要点。

测试集包含测试中使用的 1,521 张 BioID 图像、781 张 LFPW 训练图像和 249 张 LFPW 测试图像，文本文件记录了人脸边界框的边界位置。

V. 实验结果与分析

A. 训练数据

每个网络的输入我们会有 4 种训练数据输入：

Positive face 数据：图片左上右下坐标和 label 的 $\text{IOU} > 0.65$ 的图片

part face 数据：图片左上右下坐标和 label 的 $0.65 > \text{IOU} > 0.4$ 的图片

negative face 数据：图片左上右下坐标和 label 的 $\text{IOU} < 0.3$ 的图片

landmark face 数据：图片带有 landmark label 的图片

将图片分类的用途为：

网络做人脸分类的时候，使用 positives 和 negatives 的图片，因为这两种数据分得开，中间隔着个 part face+0.1 IOU 的距离，容易使模型收敛。

网络做人脸 bbox 的偏移量回归的时候，使用 positives 和 parts 的数据，之所以不用 neg 数据。我们认为是因为 neg 的数据几乎没有人脸，用这个数据来训练人脸框 offset 的回归不靠谱，相反 pos 和 part 的数据里面人脸部分比较大，用来做回归，网络还能够看到鼻子、眼睛、耳朵等来进行回归。

网络做人脸 landmark 回归的时候，就只使用 landmark face 数据。

B. P-net 输出

图片金字塔输入 P-net，得到大量的候选（candidate）根据分类得分，筛选掉一大部分的候选，再根据得到的 4 个偏移量对 bbox 进行校准后得到 bbox 的左上右下的坐标，对这些候选根据 IOU 值再进行非极大值抑制（NMS）筛选掉一大部分候选。

根据分类得分从大到小排，得到 (num_left, 4) 的张量，即 num_left 个 bbox 的左上、右下绝对坐标。每次以队列里最大分数值的 bbox 坐标和剩余坐标求出 iou，干掉 iou 大于 0.6（阈值是提前设置的）的框，并把这个最大分数值移到最终结果，最终得到 (num_left_after_nms, 16) 个候选，这些候选需要根据 bbox 坐标去原图截出图片后，resize 为 24*24 输入到 R-net。

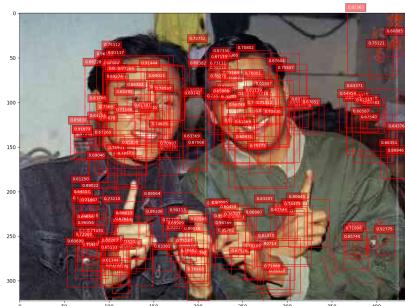


Fig. 8. P-net 输出

C. R-net 输出

经过 P-net 筛选出来的候选图片，经过 R-net 进行精调。根据 P-net 输出的坐标，去原图上截取出图片

（截取图片有个细节是需要截取 bbox 最大边长的正方形，这是为了保障 resize 的时候不产生形变和保留更多的人脸框周围细节），resize 为 24*24，输入到 R-net，进行精调。

R-net 仍旧会输出二分类 one-hot2 个输出、bbox 的坐标偏移量 4 个输出、landmark10 个输出，根据二分类得分干掉大部分不是人脸的候选、对截图的 bbox 进行偏移量调整后（说的简单点就是对左上右下的 x、y 坐标进行上下左右调整），再次重复 P-net 所述的 IOU NMS 干掉大部分的候选。最终 P-net 输出的也是 (num_left_after_R-net, 16)，根据 bbox 的坐标再去原图截出图片输入到 O-net，同样也是根据最大边长的正方形截取方法，避免形变和保留更多细节。



Fig. 9. R-net 输出

D. O-net 输出

经过 R-net 删除很多候选后的图片输入到 O-net，输出准确的 bbox 坐标和 landmark 坐标。这个过程与 P-net 的过程差不多，不过有区别的是这个时候我们除了关注 bbox 的坐标外，也要输出 landmark 的坐标。

经过分类筛选、框调整后的 NMS 筛选，就得到准确的人脸 bbox 坐标和 landmark 点。



Fig. 10. O-net 输出

E. 对比实验

复现 MTCNN 之后，我们希望了解更多人脸检测前沿内容，并使用 *hugging face* 上的 demo 测试了前沿模型与我们的模型性能对比。

SCRFN 是一种高效的高精度人脸检测方法，是 2021 年 *insightface* 提出的一个人脸检测模型，并被 ICLR-2022 接受。我们用老师提供的 mid.png 可视化效果测试，并输出了各个子网络结果，结果如下：

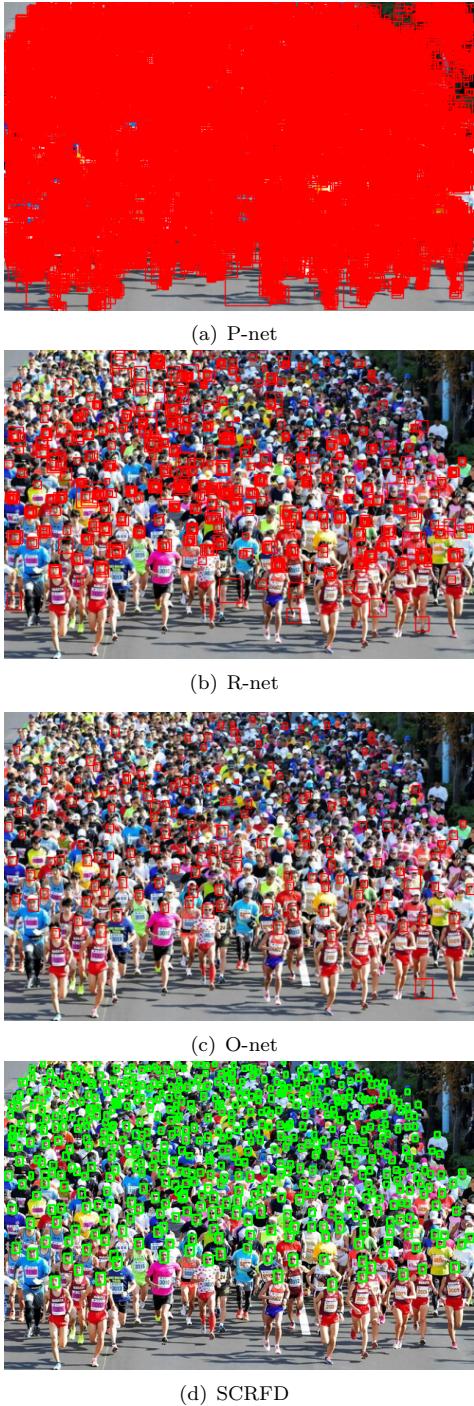


Fig. 11. 可视化效果测试

不难看出，新的 sota 检测效果非常优异，在原图很不清晰，很多人脸人眼都难以分辨的前提下，依旧几乎检测出了所有的人脸，虽然有少量漏检，但没有错检的情况。

而我们的 MTCNN 从 P-net、R-net 再到 O-net 逐步提高了精度，也展现出了较好的性能。即使有错检、漏检的情况，但 SCRFN 使用的是 *hugging face* 上 34G 最好的模型，而我们的模型大小只有 1.12MB。

VI. 实验总结

MTCNN 为了兼顾性能和准确率，避免滑动窗口加分类器等传统思路带来的巨大的性能消耗，先使用小模型生成有一定可能性的目标区域候选框，然后在使用更复杂的模型进行细分类和更高精度的区域框回归，并且让这一步递归执行，以此思想构成三层网络，分别为 P-Net、R-Net、O-Net，实现快速高效的人脸检测。在输入层使用图像金字塔进行初始图像的尺度变换，并使用 P-Net 生成大量的候选目标区域框，之后使用 R-Net 对这些目标区域框进行第一次精选和边框回归，排除大部分的负例，然后再用更复杂的、精度更高的网络 O-Net 对剩余的目标区域框进行判别和区域边框回归。

MTCNN 的优点：

- 1) 设备要求低：使用了级联思想，将复杂问题分解，使得模型能够在小型设备上运行，比如人脸识别模型可以在没有 GPU 的设备上运行。
- 2) 容易训练：三个级联网络都较小，训练模型时容易收敛。
- 3) 精度较高：使用了级联思想，逐步提高精度。

MTCNN 的缺点：

- 1) 误检率较高：因为采用了级联的思想，使得模型在训练过程中的负样本偏少，学到的模型不够 100% 准确。
- 2) 改进空间大：MTCNN 原论文模型在发表时距今已过去好几年了，随着技术的不断进步，对于原模型可以做出很多优化。

参考文献

- [1] Zhang K , Zhang Z , Li Z , et al. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks[J]. IEEE Signal Processing Letters, 2016, 23(10):1499-1503.
- [2] Guo J , Deng J , Lattas A , et al. Sample and Computation Redistribution for Efficient Face Detection[J]. 2021.