|  |  |
| --- | --- |
| MTCNN算法解构报告 | 文件类型：技术文档 |
| 文件编号： |
| 面向的部门： 研发中心 |
| 保密等级：高 |
| 作者： 刘凯 |
| 日期：2018-08-28 |
| 版本：1.0V |

MTCNN算法解构报告



目 录

[1 MTCNN算法原理 3](#_Toc523408927)

[1.1 算法框架 3](#_Toc523408928)

[1.2 网络结构 4](#_Toc523408929)

[1.3 训练 6](#_Toc523408930)

[2 MTCNN算法实现 6](#_Toc523408931)

[参考文献 1](#_Toc523408932)

# MTCNN算法原理

* 1. 算法框架

MTCNN(Multi Task Convolutional Neural Network)[1]算法是2016年提出的一种基于级联(Cascade)CNN的人脸检测和对齐算法。算法分为三个步骤，每个步骤用一个CNN网络。第一步使用的网络叫P-Net(Proposal Network)，通过一个浅层的CNN网络，产生候选窗。第二步使用的网络叫R-Net(Refine Network)，用了一个CNN网络排除大量非人脸窗，并校准。最后使用O-Net(Output Network)，进一步提高窗口质量，并输出5给人脸标志的位置(双眼，鼻子和嘴的左右角)，实现人脸检测和对齐。实现示意图如图1所示：



图1：MTCNN实现步骤

* 1. 网络结构

第一步的P-Net如图2所示，输入图片的大小为，经过三个卷积层得到特征，然后经过2个的卷积层得到2个输出(实际的网络模型和图略有不同，最后只有人脸分类和人脸边框回归两个输出)。具体的网络结构如表1所示，其中K为pooling层kernel的大小，S为步幅strike，Conv4-1和Conv4-2的输入均为Conv3，其输出分别为人脸分类和边框回归的结果。

表1：P-Net各层参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 层 | 权重 | K | S | 输出 |
| **输入** | NA | NA | NA |  |
| **Conv1** |  | NA | 1 |  |
| **Pool1** | NA | 2 | 2 |  |
| **Conv2** |  | NA | 1 |  |
| **Conv3** |  | NA | 1 |  |
| **Conv4-1** |  | NA | 1 |  |
| **Conv4-2** |  | NA | 1 |  |



图2：P-Net网络结构

第二步的R-Net如图3所示。输入图片为，经过3个卷积层和1个全连接层得到特征，然后经过两个全连接层得到2个输出，其详细的网络结构如表2所示。其中fc5-1和fc5-2的输入均为fc4，其输出分别为人脸分类和边框回归的结果。

表2：R-Net各层参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 层 | 权重 | K | S | 输出 |
| **输入** | NA | NA | NA |  |
| **Conv1** |  | NA | 1 |  |
| **Pool1** | NA | 3 | 2 |  |
| **Conv2** |  | NA | 1 |  |
| **Pool2** | NA | 3 | 2 |  |
| **Conv3** |  | NA | 1 |  |
| **fc4** |  | NA | NA |  |
| **fc5-1** |  | NA | NA |  |
| **fc5-2** |  | NA | NA |  |



图3：R-Net网络结构

第三步的O-Net如图4所示。输入图片为，经过4个卷积层和1个全连接层得到特征，然后经过三个全连接层得到3个输出，其详细的网络结构如表3所示。其中fc6-1，fc6-2和fc6-3的输入均为fc5，其输出分别为人脸分类和边框回归和人脸对齐的结果。

表3：O-Net各层参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 层 | 权重 | K | S | 输出 |
| **输入** | NA | NA | NA |  |
| **Conv1** |  | NA | 1 |  |
| **Pool1** | NA | 3 | 2 |  |
| **Conv2** |  | NA | 1 |  |
| **Pool2** | NA | 3 | 2 |  |
| **Conv3** |  | NA | 1 |  |
| **Pool3** | NA | 2 | 2 |  |
| **Conv4** |  | NA | 1 |  |
| **fc5** |  | NA | NA |  |
| **fc6-1** |  | NA | NA |  |
| **fc6-2** |  | NA | NA |  |
| **fc6-3** |  | NA | NA |  |



图4：O-Net网络结构

* 1. 训练

训练包括三个训练目标：人脸/非人脸分类(face/non-face classification)，人脸边框回归(bunding box regression)和人脸特征定位(facial landmark location)。

对于人脸/非人脸分类的目标，采用交叉熵(cross entropy)：



其中，为样本为人脸的概率，为真实的标签。

对于人脸边框回归，主要目标是边框的左上高宽四个数值，采用欧几里得距离： 



其中为网络输出的坐标值，为真实的坐标值。

和人脸边框回归类似，人脸特征定位采用欧几里得距离：



其中为网络输出的人脸特征坐标，为真实的人脸特征坐标。

# MTCNN算法实现

MTCNN的算法实现流程图如图5所示：

图片预处理

P-Net

人脸框处理

Stage 1

图片预处理

R-Net

人脸框处理

Stage 2

画图输出

图片预处理

O-Net

人脸框处理

Stage 3

特征点处理

图片输入

图像金字塔

图5：MTCNN算法流程

从图5可以看到，MTCNN的算法流程分为三层，层层串联输出，每层的流程很类似，都是经过图片预处理、前向网络和人脸框处理几个模块(最后一层多了人脸特征点处理模块)。以下对各层的模块进行说明。

* 1. 第一层P-Net

**图像金字塔**(Image Pyramid)

在第一层P-Net中，输入的图像需要做图像金字塔，即将输入的图片等比例缩放，得到一组不同尺度的输入图像；图像金字塔结合后面的目标框生成可以产生不同尺度的人脸候选框。代码里设置的人脸最小尺寸为20，而P-Net的感知野为12，因此初始的缩放尺寸为，以后每次缩放的比例为0.709，一直缩放到图片的最小尺寸小于12为止。对每个尺度的输入图片都运行一次P-Net。

**图片预处理**

对不同尺度的输入图片进行预处理，包括减去均值和归一化处理。在代码中的均值为127.5，归一化时将减去均值的图片除以128，使得输入图片的像素值在[-1,1]的范围内。

**前向P-Net**

前向P-Net的网络结构参考1.2节，P-Net的计算采用caffe框架，输出两个结果，第一个为候选框是人脸/非人脸的概率，维度为2；第二个是候选框的回归值，维度为4。

**人脸框处理**

这里的人脸框处理包括5个步骤：人脸框生成，非极大抑制(NMS, Non-Maximum Suppression)，人脸框回归，人脸框方形化和人脸框padding。

*人脸框生成*。根据P-Net的第一个输出，选择人脸概率大于某阈值(代码中为0.6)的候选框的坐标，以这个坐标为左上的起点，形成大小的矩形框，并映射到原图上，示例代码如下：

xmin = (stride\*j/scale)

ymin = (stride\*i/scale)

xmax = ((stride\*j+cellsize-1)/scale)

ymax = ((stride\*j+cellsize-1)/scale)

其中i，j分别为候选框的坐标，stride=2是因为P-Net中有一步Pooling对输入进行了2倍下采样，scale为图片的缩放比例。于是生成了原图上的候选框。这里选取的矩形框实际上是候选框选择的操作，相当于用的候选框遍历整张图片(缩放后的图片)，结合前面的图片金字塔的操作，可以得到一组不同尺度的候选框，再将可能是人脸的候选框缩放到原图片的大小。

*非极大抑制*。NMS是为了选取一组重叠区域较小的框。这里采用的是贪婪算法，即先选取得分最高的候选框，在剩下的候选框中，重叠区域大于阈值的候选框被移除，并选取重叠区域较小的候选框中得分第二高的候选框。以此类推，直到遍历所有的候选框。

*人脸框回归*。根据前面的候选框和P-Net的第二个输出，得到准确的人脸候选框的坐标值(左上和右下)。其代码如下：

box[0] += bbox\_reg[0] \* w

box[1] += bbox\_reg[1] \* h

box[2] += bbox\_reg[2] \* w

box[3] += bbox\_reg[3] \* h

其中w和h分别任人脸候选框的宽和高，bbox\_reg为P-Net输出的人脸框的回归值，box为人脸框的坐标值。

*人脸框方形化和人脸框padding*。方形化是人脸的矩形框扩大为正方形，padding是判断在图片边缘人脸候选框是否超过了图片的大小。若超过了图片大小，则需限定候选框的坐标值。

* 1. 第二层R-Net

从P-Net层输出的候选框是R-Net层的输入，候选框仍然要进行去均值和归一化的预处理，然后将候选框resize成的维度输入到R-Net中。R-Net也有两个输出，第一个为候选框是人脸/非人脸的概率，维度为2；第二个是人脸的候选框的回归值，维度为4。第一个输出同样先经过阈值判断(代码中阈值为0.7)，淘汰掉人脸概率低的候选框，然后经过NMS，人脸框回归，方形化和padding，得到R-Net层的输出。

* 1. 第三层O-Net

从R-Net层输出的候选框进入最后的O-Net，同样经过去均值和归一化的预处理，并将候选框resize成的维度输入到O-Net中。O-Net有三个输出，第一个为候选框是人脸/非人脸的概率，维度为2；第二个是人脸的候选框的回归值，维度为4；第三个为人脸特征点的坐标(包括左右眼睛，鼻子和左右嘴角)，维度为10。同样，第一个输出经过阈值判断(代码中阈值为0.7)，淘汰掉人脸概率低的候选框，然后进行人脸框回归，NMS和padding，得到最终的人脸框和人脸特征点坐标。

参考文献

1. Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, *et al*. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. IEEE Signal Processing Letters, 2016.