|  |  |
| --- | --- |
| dlib库人脸对齐算法解构报告 | 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：2018-09-13 |
| 版本：1.0V |

dlib库人脸对齐算法解构报告



目 录

[1 dlib库人脸对齐算法 3](#_Toc527038146)

[1.1 算法 3](#_Toc527038147)

[1.2 回归树的实现 4](#_Toc527038148)

[2 算法实现细节 6](#_Toc527038149)

[参考文献 1](#_Toc527038150)

# dlib库人脸对齐算法

* 1. 算法

人脸对齐的任务是对输入的人脸图像，标示出人脸特征点的坐标(如眼睛，鼻子，嘴等)。dlib库里的人脸对齐采用文献[1]中的算法，算法的核心思想是梯度提升树(Gradient Tree Boosting)[2]。

设为图片上第i个人脸标志(landmark)的坐标，则矢量



定义为人脸的**shape**，其中p为人脸标志的个数。定义为**S**当前的估计。是由一组级联的回归器(regressor)生成的：



其中为图片，为第t级回归器。

回归器的训练算法采用的是梯度提升树[2]，并且应用平方误差之和(sum of square error)作为损失函数。设训练数据为，第一个回归器为，定义初始化的三元组为：



其中，，这里的*R*表示每幅图片*I*被初始化的次数。为初始化的shape，从训练数据的shape中随机产生(除了自己本来的shape)，为初始shape的残差。shape的迭代公式为：



回归器的学习算法如图1所示。其中，为学习率，一共运行T次，得到T个回归器。对第t个回归器的输出三元组，是作为第t+1个回归器的训练数据，其中包括图片本身、初始化的shape 和回归目标(也是残差)。对齐结果如图2所示。



图1：学习回归器



图2：人脸对齐结果

* 1. 回归树的实现

上面的算法中，核心步骤是用回归树拟合残差目标。这一小节对其中的几个实现细节加以说明。

**Shape invariant split tests**

对于回归树的节点，我们是用两个像素点强度之差和阈值来做决策的。这两个像素点**u**和**v**是平均shape(mean shape)定义的坐标体系下的两个像素的坐标，对于一张图片和任意一个shape，我们可以标注出和**u，v**相对于mean shape的对应的坐标点。

具体而言，设为mean shape的人脸特征中与像素点**u**最近的特征的标号，则其相对于**u**的偏差为：



对图片和shape ，图片中与**u**最接近的点的坐标为：



其中，分别为将shape 转换到mean shape 的尺度和旋转矩阵，其值可以通过最小化下式得到：



同理可以得到。设三元组，则每个节点的决策为：



**Choosing the node splits**

我们将回归树近似成一个分段常函数(piecewise constant function)，每个叶节点对应一个常数矢量。具体而言，我们随机生成一组参数，并贪婪地选择最小化平方和误差的最优的。优化目标为：



其中，分别表示左节点和右节点，为根据参数选择左或者右的样本索引，为图片根据梯度提升算出来的残差，且：



根据可以得到最优参数为：



**Feature selection**

用两个像素点之差作为决策依据比用单个像素本身的强度更好，因为可以排除全局光照变化带来的影响。但相应的，其决策空间也会大大增减。因此我们采用一个指数分布作为选取像素点的先验：



# 算法实现细节

人脸对齐算法的输入和参数如表1所示：

表1：算法的输入和参数列表

|  |  |
| --- | --- |
| 名称 | 维度 |
| img |  |
| dets |  |
| anchor\_idx |  |
| delta |  |
| init\_shape |  |
| splits |  |
| leaf |  |

其中img和dets为输入，其他的为预先训练好的参数。img为输入的图像，*W、H*和*C*分别为图片的宽、高和通道数(RGB)，*W*对应图片的列，同理，*H*对应图片的行。dets为人脸检测框的坐标，其中为图片img中检测出来的人脸个数，每个人脸框由四个参数描述，即人脸框的。参数anchor\_idx为式中的下标，其中和分别为强分类器和弱分类器的个数，在代码中，，。参数delta为式中的。参数init\_shape为初始的人脸shape，为人脸shape的维度，在代码中68个人脸特征的shape维度为。参数splits和leaf为回归树的参数，回归树深度为4。

人脸对齐的伪代码如表2所示。算法分为三个循环，分别对应人脸个数，强分类器个数和弱分类器个数。对第一层循环，初始化curr\_shape为参考的init\_shape；在强分类器的循环中，计算坐标转换和特征像素亮度。在弱分类器的循环中，进行回归树计算，并更新curr\_shape。最后进行去归一化，得到人脸shape。

人脸坐标转换采用的是文献[3]的公式(34)到(43)。输入curr\_shape和init\_shape，输出两者的转换矩阵，即公式中的。

特征像素亮度先根据公式计算对应特征像素的坐标，然后计算图片在该像素点的亮度值，对于RGB的图像，其亮度为(R+G+B)/3。

回归树的计算入公式所示，根据splits和特征像素亮度做决策，选择对应的leaf上的回归值更新curr\_shape。

去归一化映射是将人脸框和单位框，即由(0,0)和(1,1)确定的框，进行映射，因为curr\_shape是定义在(0,1)之间的，需要映射到图片的人脸框上。

表2：人脸对齐伪代码

|  |
| --- |
| 输入：img, dets  **for** (idets=1:dets)  curr\_shape = init\_shape;  **for** (ist=1:)  ***计算当前人脸坐标系到参考坐标系的转换关系***  ***计算特征像素的亮度值***  **for** (iweak=1:)  ***计算回归树，更新curr\_shape***  end  end  ***去归一化映射***  end  输出： |

# 仿真结果

图3为人脸对齐仿真结果，其中左边为dlib库人脸对齐算法的结果，右边为重构算法的仿真结果。可以看到两者差别不大，可以算作人脸对齐成功。只是实际的结果数据仍有少量偏差(大概在正负1个像素之间)。

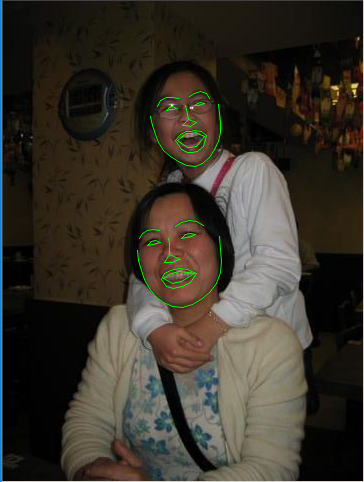


图3：人脸对齐仿真结果

参考文献

1. Vahid Kazemi and Josephine Sullivan. One Millisecond Face Alignment with an Ensemble of Regression Trees. CVPR 2014.
2. T. Hastie, R. Tibshirani, and J. H. Friedman. The elements of statistical learning: data mining, inference, and prediction. New York: Springer-Verlag, 2001.
3. Shinji Umeyama. Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991.