### 卡通人脸检测技术报告

姓名: 陆放明

学号: 171250660

代码链接:

###### 任务描述

针对已有的卡通人脸进行人脸检测，模型需要能够基于输入的卡通人脸，给出具体的人脸bbox描述。最终的输出文件 test.txt，作为模型预测结果。

###### 技术选型

本实验中，采用非深度学习的方式。借鉴opencv[1]中的基于AbaBoost进行级联分类器的训练，采用集成学习的方式，针对每一个弱学习器进行级联，最终形成基于弱学习器的一个Boosting 加性模型。每一个弱学习器提取图像中的LBP特征[3]进行训练。

###### 实验步骤说明

数据预处理

1. 灰度图生成

首先输入的图像均为彩色RGB图像，为了加快特征提取的速度，首先针对每一张图片生成相同尺度的灰度图，具体的代码位于gen\_cartoon\_middle.py中的 gen\_gray\_img 方法。最终在 dataset/cartoon\_dataset/gray\_train 目录下生成所有的训练集灰度图。具体的样例如下：

许多照片放在一起

描述已自动生成

于此同时，为了生成便于opencv训练对应的训练bbox训练真值文件，实验中基于 ./ dataset/cartoon\_dataset/train.txt 文件，生成 ./dataset/cartoon\_dataset/train\_bbox\_gt.txt文件。具体的文件生成格式如下：



图表示 000001.jpg 图片中有 1 个人脸bbox，它的位置是 (xmin, ymin, w, h) = (246 , 141 , 155, 155)。它由原文本第一行 000001.jpg,246,141,401,296 内容转化而来

1. 正样本 / 负样本生成

实验接下来针对已有的灰度图数据集，进行数据扩充 (augmentation) ，主要分为正样本生成和负样本生成两部分 (代码逻辑位于gen\_cartoon\_middle.py中的gen\_middle\_img 方法)

原数据集的照片数量为**8000** 张，bbox数目为 **14701**。在这里首先基于每一张图片的每一个bbox（分别为当前bbox的宽度与长度像素值），选取一个新的bbox框。生成策略如下:

根据当前图像的 w, h 大小，给出一个新的bbox尺寸 size，这个size随机进行获取，介于 之间，得到x , y偏移量 ，基于此给出新的bbox四元组：

在获取得到新的bbox之后，针对所有的bbox进行归一化，将尺寸resize到target\_size大小。实验中，target\_size被设置为了。针对归一化之后的bbox，对原图片进行裁剪，得到四元组。将生成的正样本图片放置在 ./middle\_data\_set / positive 目录下，并且将bbox四元组追加写入 ./middle\_data\_set/positives.txt 文件中。

针对如上的裁剪过程，每一个bbox都重复进行5次。故最终的正样本数目被扩充至原有的5倍(63752个正样本)

为了提取得到opencv训练所需的background数据，进行负样本生成。这里具体的生成逻辑如下，并且会将crop\_box归一化至target\_size尺度(与正样本生成类似)，并且计算和当前图片内各个bbox的IoU大小。只有当最大的IoU值小于 0.01 时，才认定为负样本。针对每一张图片，如上过程进行40次。最终生成319999个负样本

手机屏幕的截图

描述已自动生成

具体的正样本、负样本生成示例如下图：

图片包含 照片, 看着, 活, 房间

描述已自动生成

###### 级联模型训练

这里的模型训练中，opencv版本为 3.4.7。首先基于正样本集合生成63752个样本，其尺寸归一化为。此外，opencv\_traincascade的具体参数如下。所有训练的参数说明参考于[1][4]

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 值 |
| numPos | 每一级联提取的正样本个数 | 8000 |
| numNeg | 每一级联提取的负样本个数 | 24000 |
| minHitRate | 分类器的每一级希望得到的最小检测率 | 0.995 |
| maxFalseAlarmRate | 分类器的每一级希望得到的最大误检率 | 0.4 |
| maxWeakCount | 每一级的最大弱分类器数 | 500 |
| numStages | 训练级数 | 14 |

最终生成的级联分类器描述xml文件位于 ./data/cascade.xml

###### 模型预测与评估

预测部分：

主要的代码逻辑位于 evaluater.py 文件内。首先对于测试集的每一个图片进行灰度图处理，而后输入给级联分类器，输出为 n 个 (x1,y1, w, h) 四元组 , n为预测的卡通人脸个数

针对每一张图片生成的n个四元组，追加写入到最终的 test.txt 文件中

评估部分：

我们令所有预测检索到的脸总数为N\_predict, 所有系统的真值脸总数 N\_grondtruth, 检索正确的脸个数为TP

和实验要求一致，针对每一个预测的bbox，考察所有的ground bbox，如果存在一个ground bbox和预测的边框IoU > 0.7 , 那么令 TP 数加一

最终的P (Precision) 和 R(Recall) 计算如下

而后计算F1 score:

###### 实验结果描述

针对训练集全集下，实验结果为: F1 score = 0.395

###### 引用

* [1] <https://docs.opencv.org/4.0.0-alpha/d6/d00/tutorial_py_root.html>
* [2] Jha, Saurav & Agarwal, Nikhil & S., Agarwal. (2018). Bringing Cartoons to Life: Towards Improved Cartoon Face Detection and Recognition Systems.
* [3] Ahonen T, Hadid A. and Pietikäinen M. “Face description with local binary patterns: Application to face recognition.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(12):2037-2041.
* [4] https://blog.csdn.net/qq\_26898461/article/details/49514787