

人脸识别系统设计

一、 设计内容

本设计欲实现一个完整的人脸识别系统，最后达到从 USB 相机采集图像流进行实时的人脸检测并识别的功能。系统在图像采集后转换颜色为灰度，使用方向梯度直方图^[1]（Histogram of Oriented Gradients, HOG）的方法进行人脸的检测；检测到人脸后，利用面部特征点估计^[2]（face landmark estimation）的算法将人脸对齐；最后使用 FaceNet^[3]提取脸部特征点，识别过程则采用支持向量机^[4]（Support vector machine, SVM）分类器进行人脸识别。系统流程图如图 1 所示。

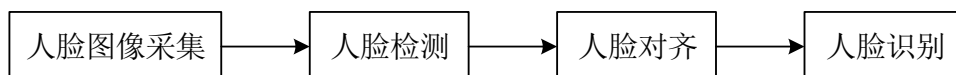


图 1.人脸检测流程图

二、 设计方法

2.1 HOG 人脸检测

主要思想：在一副图像中，局部目标的表象和形状（appearance and shape）能够被梯度或边缘的方向密度分布很好地描述。（本质：梯度的统计信息，而梯度主要存在于边缘的地方）。首先将图像分成小的连通区域，称为细胞单元。然后采集细胞单元中各像素点的梯度的或边缘的方向直方图。最后把这些直方图组合起来就可以构成特征描述器。

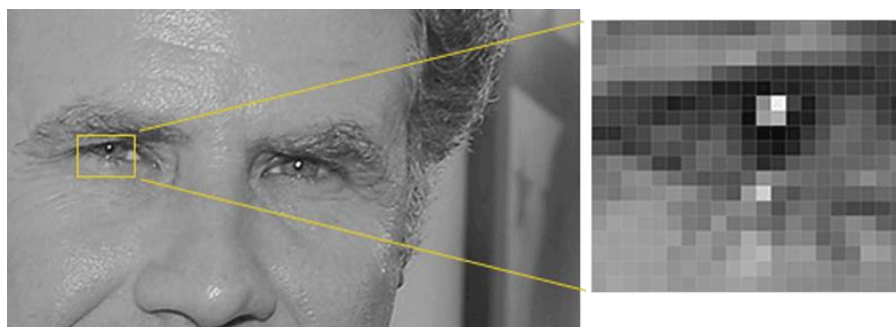


图 2.逐像素遍历卷积

计算人脸图像横坐标和纵坐标方向的梯度，并据此计算每个像素位置的梯度方向值；首先用 $[-1,0,1]$ 梯度算子对原图像做卷积运算，得到 x 方向（水平方向，以向右为正方向）的梯度分量 gradscalx ，然后用 $[1,0,-1]^T$ 梯度算子对原图像做卷

积运算，得到 y 方向（竖直方向，以向上为正方向）的梯度分量 $grad_{scaly}$ 。然后再用以上公式计算该像素点的梯度大小和方向。将图像划分成小 cells（例如 6×6 像素/cell）；统计每个 cell 的梯度直方图（不同梯度的个数），即可形成每个 cell 的 descriptor；将每几个 cell 组成一个 block（例如 3×3 个 cell/block），一个 block 内所有 cell 的特征 descriptor 串联起来便得到该 block 的 HOG 特征 descriptor。将图像内的所有 block 的 HOG 特征 descriptor 串联起来就可以得到图像的 HOG 特征描述器

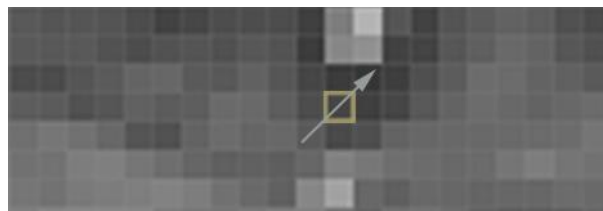


图 3.绘制明暗变化方向

2.2 面部关键点提取

使用 Vahid Kazemi 和 Josephine Sullivan 提出的面部对齐方法^[2]。对每个人脸提取 68 个关键点（landmark）：下巴，两只眼睛的外边缘，眉毛的内边缘等等。如图 4 所示。

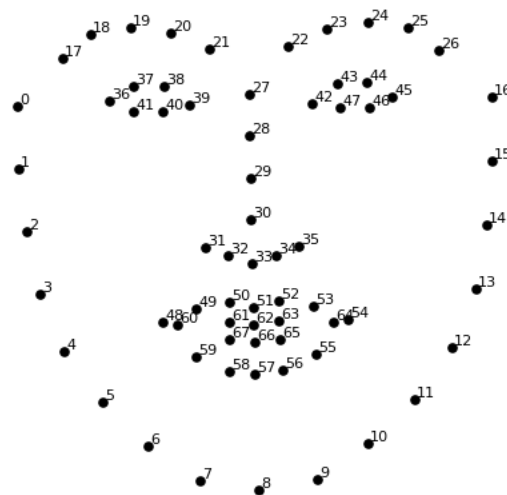


图 4.人脸的 68 个关键点示意图

提取出关键点后使用图像的仿射变换对脸部进行校正和裁剪处理，得到人脸的便于后续的特征提取，人脸对齐实验结果如图 5 所示，将 68 个关键点连成线后的效果，其中实验过程为实时对视频流进行处理。



图 5.人脸对齐实验

2.3 面部特征点估计及识别

参考文献^[4]，将校正裁剪居中的人脸图片传入训练好的 FaceNet 网络模型进行编码得到 128 个特征测量值并保存。最后，识别过程中将每个人脸表示为 128 维空间中的点，即是一个人脸为一个点。要检查两个人脸是否匹配，直接检查这两个点之间的距离是否小于 0.6（经验阈值）。

2.4 程序运行条件和步骤

- [1] 系统平台：Ubuntu16.04 + cuda_10.0.130 + cudnn_v7.4.1.5(GTx 750Ti)
- [2] 软件平台：Python3.5.2 + OpenCV3.3.1 + PyTorch1.0.1(自带 caffe)
- [3] 将预识别的包含人脸的图片以人名命名放入 face_recognition/examples/文件夹中；
- [4] 插入 USB 摄像头运行程序：python facerec_from_webcam_faster.py。

三、 运行结果

SVM 分类过程中阈值取为 0.6，若低于 0.6 的阈值会使人脸更接近准确率更高！最终系统运行结果如图 6 所示。

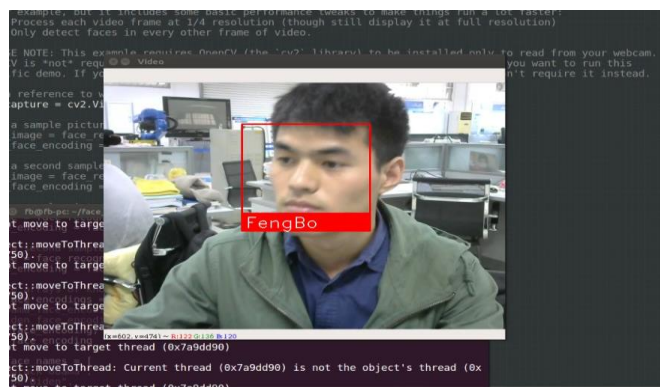


图 6.系运行截图

图 6 为在线测试运行截图，该系统能实时的进行人脸检测与识别。更重要的是本系统能够同时在线进行多张人脸的检测识别。

四、 讨论

在实验过程中可以看到，当人脸离摄像头较远时，人脸检测的效果不是很好，这一点可以通过程序来设定将

```
face_locations = face_recognition.face_locations(rgb_frame)
```

更改为：

```
face_locations=face_recognition.face_locations(rgb_frame,number_of_times_to_
upsample = 2)
```

这样能够检测更小的脸，但是代价就是运行帧率降低以及内存消耗更大。

总之，人脸识别作为一项非常普遍的应用目前已经有很多成熟的方案，同时也有很多更好更优秀的方案在涌现出来。十年前还停留在想象阶段的事情，现如今已经成熟投入使用，科技的进步与发展速度令人咂舌，就算作为一名平凡的普通人也唯有不断地学习，紧跟时代的步伐，才能享受科技带来的便利，才能享受当代生活的极致体验。

参考文献

- [1] N.Dalal, B.Triggs.Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR' 05)
- [2] V.Kazemi, J.Sullivan.One Millisecond Face Alignment with an Ensemble of Regression Trees. Computer Vision and Pattern Recognition, At Columbus, Ohio, USA
- [3] F.Schroff ,D.Kalenichenko,J.Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2015
- [4] https://en.wikipedia.org/wiki/Support-vector_machine
- [5] <http://dlib.net/>
- [6] <https://cmusatyalab.github.io/openface/>