

AI Demo



认识你的脸

汤志远

文件夹结构

- ❑ AIDemo 虚拟机桌面上点击右键，选择“打开终端”。
- ❑ 用linux命令进入pca-face-recognition，并查看文件夹内容。

- doc: 样例程序文档说明
- code: 样例程序代码
- code/run.sh: 样例程序入口

```
[tutorial@localhost ~]$ ls
aibook  下载  公共  图片  文档  桌面  模板  视频  音乐
[tutorial@localhost ~]$ cd aibook/
[tutorial@localhost aibook]$ ls
demo
[tutorial@localhost aibook]$ cd demo/
[tutorial@localhost demo]$ ls
data  env  image  lang  mind  robot  speech
[tutorial@localhost demo]$ cd image/
[tutorial@localhost image]$ ls
dnn-face-recognition  face-detection  pca-face-recognition  tf-neural-style  tf-neural-texture
[tutorial@localhost image]$ cd pca-face-recognition/
[tutorial@localhost pca-face-recognition]$
[tutorial@localhost pca-face-recognition]$
[tutorial@localhost pca-face-recognition]$
```

绝大多数情况，运行如下命令持行样例程序：sh run.sh

人脸识别简介

- 人脸识别，是基于人的脸部特征信息进行身份识别的一种生物识别技术。用摄像机或摄像头采集含有人脸的图像或视频流，并自动在图像中检测和跟踪人脸，进而对检测到的人脸进行脸部识别的一系列相关技术，通常也叫做人像识别、面部识别。
- 一组特征脸可以通过在一大组描述不同人脸的图像上进行主成分分析（PCA）获得。任意一张人脸图像都可以被认为是这些标准脸的组合。例如，一张人脸图像可能是特征脸1的10%，加上特征脸2的55%，在减去特征脸3的3%。值得注意的是，它不需要太多的特征脸来获得大多数脸的近似组合。



FACE RECOGNITION USING PCA

特征	0	1
特征1.眼皮	单眼皮	双眼皮
特征2.鼻子	塌鼻梁	挺鼻梁
特征3.眼球	蓝色	灰色
特征4.肤色	黄色	白色
特征5.发色	棕黄	黑色

实验一：运行缺省配置

对测试集中的数据预测每张照片的身份（即训练集中出现过的人物）

```
[tutorial@localhost code]$ sh run.sh
/home/tutorial/aibook/demo/env/py2.7/lib/python2.7/site-packages/sklearn/cross_vali
nterface of the new CV iterators are different from that of this module. This module
"This module will be removed in 0.20.", DeprecationWarning)
/home/tutorial/aibook/demo/env/py2.7/lib/python2.7/site-packages/sklearn/grid_search.
ed in 0.20.
DeprecationWarning)
Total dataset size:
n_samples: 1288
n_features: 1850
n_classes: 7
Extracting the top 150 eigenfaces from 966 faces
/home/tutorial/aibook/demo/env/py2.7/lib/python2.7/site-packages/sklearn/utills/depre
n DOES NOT store whitened components. Apply transform to get them.
warnings.warn(msg, category=DeprecationWarning)
done in 0.218s
Projecting the input data on the eigenfaces orthonormal basis
done in 0.085s
Fitting the classifier to the training set
done in 21.142s
Best estimator found by grid search:
SVC(C=1000.0, cache_size=200, class_weight='balanced', coef0=0.0,
decision_function_shape='ovr', degree=3, gamma=0.005, kernel='rbf',
max_iter=1, probability=False, random_state=None, shrinking=True,
tol=0.001, verbose=False)
Predicting people's names on the test set
done in 0.084s
```

该示例程序提取的 PC 个数为 150

	precision	recall	F1-score	support
Ariel Sharon	0.77	0.53	0.62	19
Colin Powell	0.80	0.78	0.79	55
Donald Rumsfeld	0.80	0.64	0.71	25
George W Bush	0.81	0.97	0.88	142
Gerhard Schroeder	1.00	0.77	0.87	31
Hugo Chavez	1.00	0.75	0.86	16
Tony Blair	0.79	0.68	0.73	34
avg / total	0.83	0.83	0.82	322

七位测试者的识别性能，其中precision为预测结果中标记为某人的照片里确实为该人照片的比例；recall为某人所有照片中被正确标记出来的比例；F1-score为上述两种性能指标的综合值。

10	4	1	4	0	0	0]
2	43	1	8	0	0	1]
0	0	16	8	0	0	1]
1	3	0	138	0	0	0]
0	0	1	2	24	0	4]
0	2	0	2	0	12	0]
0	2	1	8	0	0	23]

这七位测试之间的混淆度，其中第i行第j列的数字为识别系统将第i人识别为第j人的次数。

实验一：运行缺省配置



重组



George W Bush



实验二：修改运行参数

Pca-face-recognition 系统的主要参数是n-components，即EigenFace的个数，定义在recog.py中。这一参数值越小，提取的PCA特征越精减，计算速度越快，但识别性能越低。修改这一参数，可以观察EigenFace个数和识别性能的关系。

实验三：识别自己的照片

上传一张自己的照片，将其放在my_home下，并修改recog-myphoto.py中的photo_fn变量，指向该照片的文件名。运行run-myphoto.sh，即可得到对该照片的预测。注意，该预测的结果只能是在原LFW数据集中的人名，因此可以理解为你和该LFW中谁更相似。AIDemo中上传了一张作者的照片，该结果显示作者和Colin Powell最相似。

The end