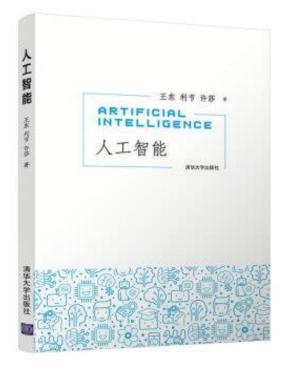


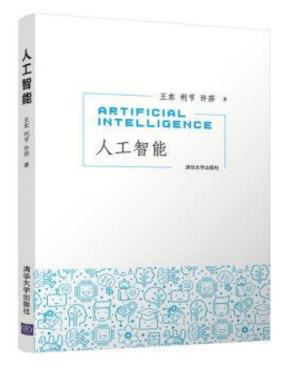
认识你的脸

利节



AIDemo 示例

http://aibook.cslt.org



实验准备:

AIDemo 提供了一个基于PCA+SVM 的人脸识别示例程序, 安装在image/pca-face-recognition 目录下。该示例程序事 实上是机器学习开源工具包sklearn 的一个演示程序。该程序 利用LFW 数据库训练PCA 模型和SVM 模型,训练过程需要 sklearn 工具包。因此,需要下载LFW 数据库并安装sklearn。 为了方便读者,AIDemo 已经将这些数据库和工具包安装完毕, 读者只需依doc/README 下的步骤进行实验。



实验一: 运行缺省配置

在终端窗口中运行image/pca-face-recognition/code 目录下的run.sh,即可观察到该人脸识别系统的缺省运行过程。该过程分为如下几步:

- 1. 从LFW 数据库读入全部人脸照片,并将这些照片分为训练集和测试集;
- 2. 基于训练集中的人脸照片数据,训练PCA 模型,每个PC 对应一个 Eigen Face;将训练集中的人脸照片在PC 上做映射,得到的映射结果 即为原始照片的特征向量;
- 3. 基于上一步提取的PCA 特征训练SVM 模型;
- 4. 基于SVM 模型,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)。



实验二:修改运行参数

Pca-face-recognition 系统的主要参数是n-components,即 Eigen Face

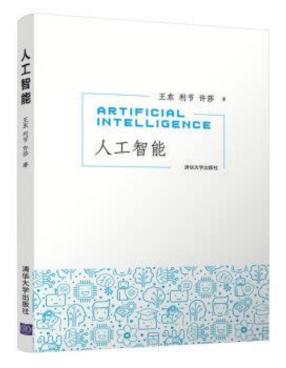
的个数,定义在recog.py 中。这一参数值越小,提取的PCA 特征越精减,

计算速度越快,但识别性能越低。修改这一参数,可以观察**Eigen** Face 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。AlDemo 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!