# 中兴人工智能比赛-人脸认证

## 算法说明

具体算法说明详见算法代码（有注释）

### 图像预处理

对包含人脸的图像进行人脸框识别，人脸对齐和人脸剪裁。

### 原理

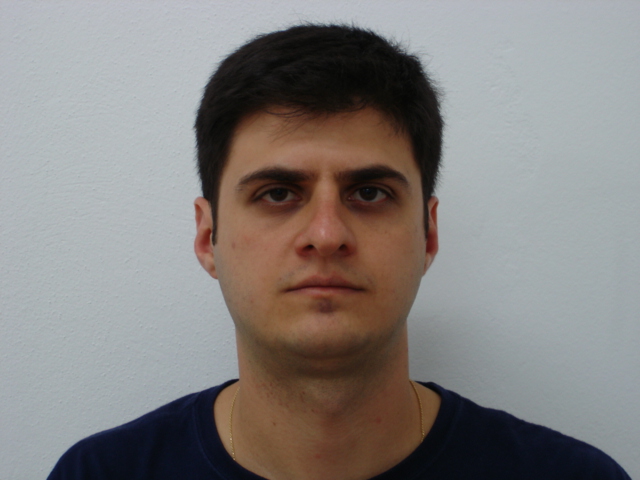
人脸框识别

人脸对齐

人脸剪裁

**这三个步骤可以用我做的一个小工具：**[**FaceTools**](https://github.com/RiweiChen/FaceTools) **来一键完成。**

具体来说，需要选择一个标准的人脸图像作为对齐的基准，如图：

****

**训练数据通过对齐后是这样的：**

****

**LFW测试数据通过对齐后是这样的：**

### 数据转换

图像处理好之后，需要将其转化为Caffe 可以接受的格式。虽然Caffe支持直接读图像文件的格式进行训练，但是这种方式磁盘IO会比较的大，所以我这里不采用图像列表的方式，而是将训练和验证图片都转化为LMDB的格式处理。

### caffe训练

根据DeepID的网络使用caffe训练得到模型参数。

### 原理

对原始数据分离训练集和测试集

转换为caffe可以处理的lmdb格式

根据设定的Net网络和Solver配置文件进行训练

得到训练的模型

### 实现

修改DeepID.py中demo(num)方法中的人脸对齐后的文件夹以及最后一行的中训练的人数（1-10575）

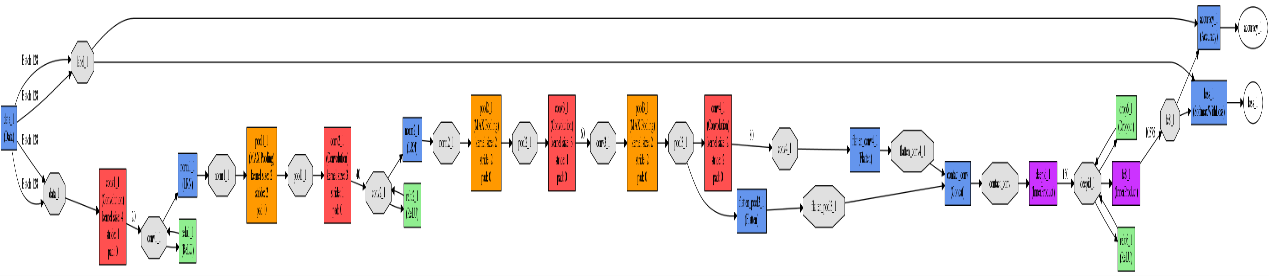
### 参考论文

1. [deepID](http://mmlab.ie.cuhk.edu.hk/pdf/YiSun_CVPR14.pdf)《Deep learning face representation from predicting 10,000 classes》
2. [deepID2](http://mmlab.ie.cuhk.edu.hk/pdf/YiSun_CVPR14.pdf)《Deep Learning Face Representation by Joint Identification-Verification》
3. [deepID2+](http://www.ee.cuhk.edu.hk/~xgwang/papers/shaoKLWcvpr15.pdf)《Deeply Learned Attributes for Crowded Scene Understanding", IEEE Conf. on Computer Vision and Pattern Recognition, June 2015 (Oral)》

[deepID3](http://arxiv.org/pdf/1502.00873.pdf)《Face Recognition with Very Deep Neural Networks》

### 详细说明

DeepID是深度学习方法进行人脸识别中的一个简单，却高效的一个网络模型，其结构的特点可以概括为两句话：1、训练一个多个人脸的分类器，当训练好之后，就可以把待测试图像放入网络中进行提取特征，2对于提取到的特征，然后就是利用其它的比较方法进行度量。



通过深度学习来进行图像高级特征表示（DeepID），进而进行人脸的分类。

优点：在人脸验证上面做，可以很好的扩展到其他的应用，并且夸数据库有效性；在数据库中的类别越多时，其泛化能力越强，特征比较少，不像其他特征好几K甚至上M，好的泛化能力+不过拟合于小的子训练集。

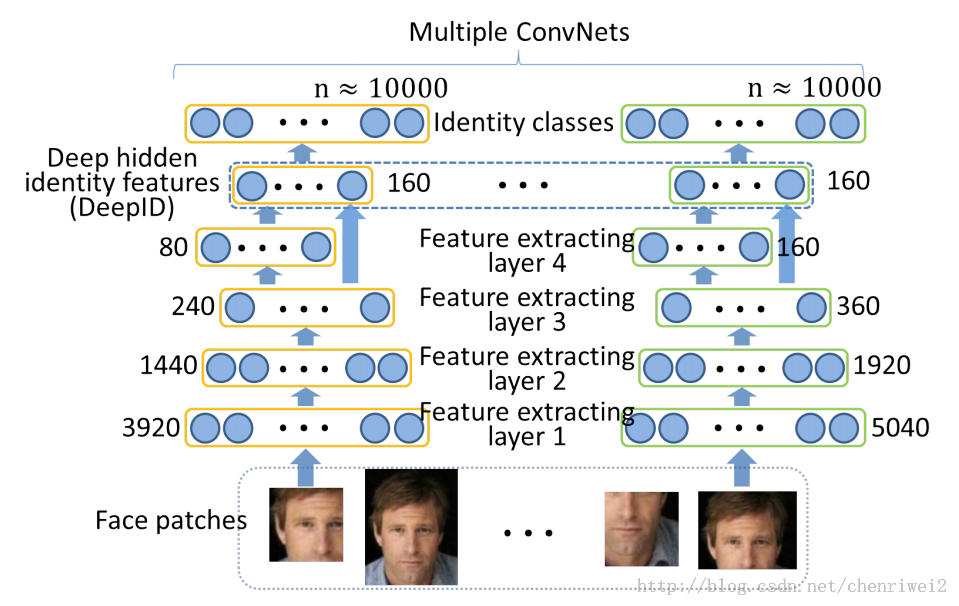
主要过程：采用卷积神经网络（CNN）方法，并且采用CNN最后一层的激活值输出作为features，不同的人脸区域放入CNN中提取特征，形成 了互补、过完全的特征表示。（form complementary and over-complete representations）。

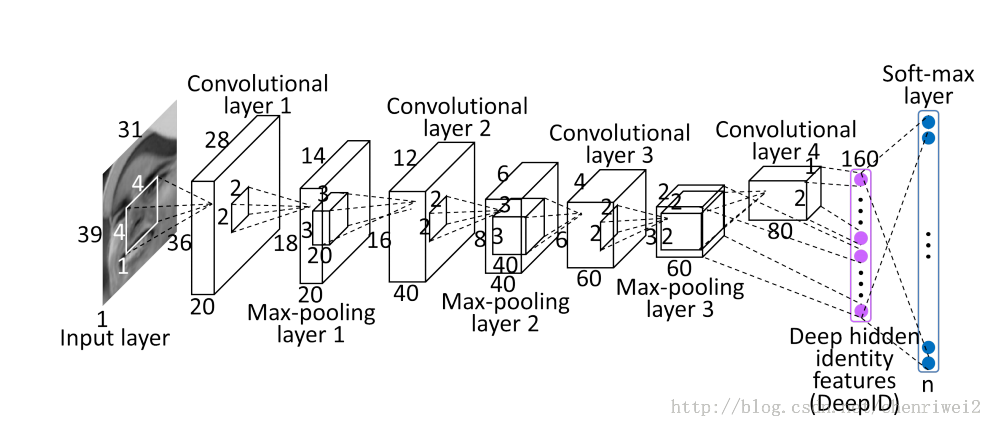
通过深度卷积网络来学习高级的过完全特征（有监督），CNN的最后一层激活值作为输出，

具体细节：

采用3个尺度，10个人脸region，60个patch，训练60个CNN网络，每个提取两个160维的特征（两个是因为： extracts two 160-dimensional DeepID vectors from a particular patch and its horizontally flipped counterpart.），所以最后一张人脸图像的特征的维度是：160\*2\*60=19200维。

CNN的结构如下：



****

**说明**：共5层网络，越往上的神经元的个数就越少，到最后就剩下160个神经元的输出，上面的Face patches 是进过对齐过后的的人脸块，也就是说已左（右）眼为中心的人脸区域块，嘴角为中心的人脸区域块等等，这样就有多个不同的输入块输入到CNN中，文章采用了 把倒数第二层的输出+倒数第一层的输出作为特征（这应该是采用12年的Le Cun 那篇文章的track）。最后再把不同的块所输出的特征连接起来，就形成了一个最终一张人脸的特征。然后再用各种分类器对其特征进行分类。

采用Max-Pooling，softmax；

输入图像：39\*31\*k 个人矩形脸图像块+31\*31\*k（这里k在彩色图像时为3，灰度时k为1）个人脸正方形块（因为后面要考虑到是全局图像还是局部图像，且需要考虑到尺度问题），使用ReLU非线性处理；

不同的输入图像：



其中局部图像是关键点（每个图像一个关键点）居中，不同的区域大小和不同的尺度图像输入到CNN中，其CNN的结构可能会不相同，但是最后的特征的都是160维度，最后将所有的特征级联起来。

最后一层的特征是第三层和第四层全相连（比较特殊的地方），因为这样可以加入尺度特征，因为第三层和第四层学习到的特征的尺度是不一样的。

特点：提取的特征很Compact，只有160\*k，k不大。自然就具有判别力了。

在训练CNN中，训练数据的类别越多，其性能越好，但是会在训练模型中出现问题，也就是太慢。

CNN的输出是特征，而不是输出类别，

**分类**：

采用Joint Bayesian 来进行人脸的verification；也采用了神经网络来比较，但是联合贝叶斯的效果比较好；

方法比较：

当前的人脸识别方法：过完全的低级别特征+浅层模型。

ConvNet 能够有效地提取高级视觉特征。

已有的DL方法：

1. Huang【CVPR2012】的生成模型+非监督；

2. Cai 【2012】的深度非线性度量学习；

3 Sun【CVPR2013】的监督学习+二类分类（人脸校验 verfication），是作者去年写的。而这一篇文章是多类分类问题（identification），而且这篇文章中，有10000类的人脸类别。

## 训练数据集

### 开放数据集：

VGG: <http://www.robots.ox.ac.uk/~vgg/software/vgg_face/>

CMU-OpenFace: <http://cmusatyalab.github.io/openface/>

### Datasets

1. [CASIA WebFace Database](http://www.cbsr.ia.ac.cn/english/CASIA-WebFace-Database.html). 10,575 subjects and 494,414 images
2. [Labeled Faces in the Wild](http://vis-www.cs.umass.edu/lfw/).13,000 images and 5749 subjects
3. [Large-scale CelebFaces Attributes (CelebA) Dataset](http://mmlab.ie.cuhk.edu.hk/projects/) 202,599 images and 10,177 subjects. 5 landmark locations, 40 binary attributes.
4. [MSRA-CFW](http://research.microsoft.com/en-us/projects/msra-cfw/). 202,792 images and 1,583 subjects.
5. [MegaFace Dataset](http://megaface.cs.washington.edu/) 1 Million Faces for Recognition at Scale 690,572 unique people
6. [FaceScrub](http://vintage.winklerbros.net/facescrub.html). A Dataset With Over 100,000 Face Images of 530 People.
7. [FDDB](http://vis-www.cs.umass.edu/fddb/).Face Detection and Data Set Benchmark. 5k images.
8. [AFLW](https://lrs.icg.tugraz.at/research/aflw/).Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization. 25k images.
9. [AFW](http://www.ics.uci.edu/%7Exzhu/face/). Annotated Faces in the Wild. ~1k images. 10.[3D Mask Attack Dataset](https://www.idiap.ch/dataset/3dmad). 76500 frames of 17 persons using Kinect RGBD with eye positions (Sebastien Marcel)
10. [Audio-visual database for face and speaker recognition](https://www.idiap.ch/dataset/mobio).Mobile Biometry MOBIO <http://www.mobioproject.org/>
11. [BANCA face and voice database](http://www.ee.surrey.ac.uk/CVSSP/banca/). Univ of Surrey
12. [Binghampton Univ 3D static and dynamic facial expression database](http://www.cs.binghamton.edu/%7Elijun/Research/3DFE/3DFE_Analysis.html). (Lijun Yin, Peter Gerhardstein and teammates)
13. [The BioID Face Database](https://www.bioid.com/About/BioID-Face-Database). BioID group
14. [Biwi 3D Audiovisual Corpus of Affective Communication](http://www.vision.ee.ethz.ch/datasets/b3dac2.en.html). 1000 high quality, dynamic 3D scans of faces, recorded while pronouncing a set of English sentences.
15. [Cohn-Kanade AU-Coded Expression Database](http://www.pitt.edu/%7Eemotion/ck-spread.htm). 500+ expression sequences of 100+ subjects, coded by activated Action Units (Affect Analysis Group, Univ. of Pittsburgh.
16. [CMU/MIT Frontal Faces](http://cbcl.mit.edu/software-datasets/FaceData2.html) . Training set: 2,429 faces, 4,548 non-faces; Test set: 472 faces, 23,573 non-faces.

### Trained Model

1. [openface](https://github.com/cmusatyalab/openface). Face recognition with Google's FaceNet deep neural network using Torch.
2. [VGG-Face](http://www.robots.ox.ac.uk/%7Evgg/software/vgg_face/). VGG-Face CNN descriptor. Impressed embedding loss.

## 运行环境

|  |  |
| --- | --- |
| 电脑软硬件说明 | |
| CPU | Intel 酷睿i7 4790K |
| RAM | 16G DDR3-1600 |
| GPU | GTX980 4GRAM |
| OS | Ubuntu 14.04 LTS desktop |
| 深度学习框架1 | Caffe |
| 深度学习框架2 | Torch |
| 深度学习框架3 | Tensorflow |
| 深度学习框架4 | Mxnet |
| 深度学习框架5 | Theano |
| 辅助工具1 | Opencv(3.1.0/2.4.13) |
| 辅助工具2 | Anaconda（python2.7） |
| 辅助工具3 | CUDA7.5 |
| 辅助工具4 | Cudnn 5.0.5 |
|  |  |

## ROC

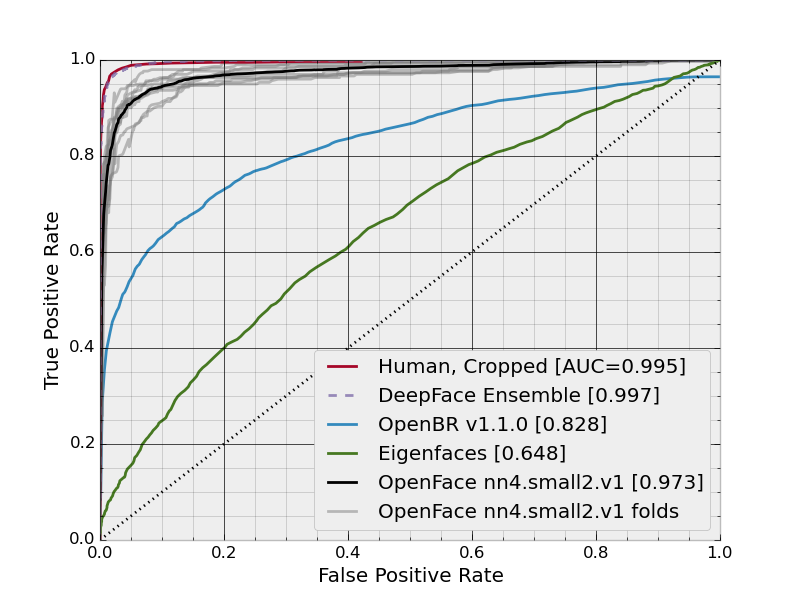
### Performance

| Model | Runtime (CPU) | Runtime (GPU) |
| --- | --- | --- |
| nn4.v1 | 75.67 ms ± 19.97 ms | 21.96 ms ± 6.71 ms |
| nn4.v2 | 82.74 ms ± 19.96 ms | 20.82 ms ± 6.03 ms |
| nn4.small1.v1 | 69.58 ms ± 16.17 ms | 15.90 ms ± 5.18 ms |
| nn4.small2.v1 | 58.9 ms ± 15.36 ms | 13.72 ms ± 4.64 ms |

### Accuracy on the LFW Benchmark

| Model | Accuracy | AUC |
| --- | --- | --- |
| **nn4.small2.v1** (Default) | 0.9292 ± 0.0134 | 0.973 |
| nn4.small1.v1 | 0.9210 ± 0.0160 | 0.973 |
| nn4.v2 | 0.9157 ± 0.0152 | 0.966 |
| nn4.v1 | 0.7612 ± 0.0189 | 0.853 |
| FaceNet Paper (Reference) | 0.9963 ± 0.009 | not provided |

### ROC Curves



#### 

