

天妒

目录视图

摘要视图

RSS 订阅

C++读书笔记

在平时写程序的过程中遇到的一些问题，把所遇到的问题及解决方案记录下来，供日后查阅。

算法练习

算法的修炼，不是一朝一夕之事，日积月累吧。

个人资料



天妒WS

访问：41395次

积分：408

等级：BLOG > 2

异步赠书：Kotlin领衔10本好书 SDCC 2017之区块链技术实战线上峰会 程序员8月书讯 每周荐书：Java Web、Python极客编程（评论送书）

使用Caffe尝试DeepID

标签：DeepID Caffe FaceVerifi CNN LFW

2015-12-17 00:23

15144人阅读

评论(142)

分类：算法

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)

[+]

使用Caffe复现DeepID实验

本实验使用Casia-Webface part2的切图来复现DeepID实验结果。

DeepID网络配置文件

训练验证数据组织

实验结果

关闭

排名： 千里之外

原创： 7篇 转载： 0篇
译文： 0篇 评论： 164条

文章搜索

文章分类

[论文读书笔记](#) (1)
[C++学习笔记](#) (1)
[算法](#) (1)
[算法练习](#) (1)
[机器学习个人心得体会](#) (4)

文章存档

[2016年05月](#) (1)
[2015年12月](#) (1)
[2014年10月](#) (1)
[2014年09月](#) (3)
[2014年08月](#) (1)

阅读排行

[Caffe在Cifar10上复现Re](#)
(19432)
[使用Caffe尝试DeepID](#)
(15142)
[libsvm在python下的使用](#) (2224)

结果分析

DeepID网络配置文件

-下面给出deepId_train_test.prototxt的内容

```
1 name: "deepID_network"
2 layer {
3   name: "input_data"
4   top: "data"
5   top: "label"
6   type: "Data"
7   data_param {
8     source: "dataset/deepId_train_lmdb"
9     backend: LMDB
10    batch_size: 128
11  }
12  transform_param {
13    mean_file: "dataset/deepId_mean.proto"
14  }
15  include {
16    phase: TRAIN
17  }
18 }
19 layer {
20   name: "input_data"
21   top: "data"
22   top: "label"
23   type: "Data"
24   data_param {
25     source: "dataset/deepId_test_lmdb"
```

关闭

《Deep Learning Face F (1419)

C++ dll动态链接库头文件 (1238)

SVM , ROC读书报告 (951)

最长公共子串问题 (514)

评论排行

使用Caffe尝试DeepID (142)

Caffe在Cifar10上复现Re (20)

C++ dll动态链接库头文件 (1)

libsvm在python下的使用 (1)

最长公共子串问题 (0)

SVM , ROC读书报告 (0)

《Deep Learning Face F (0)

推荐文章

* CSDN日报20170828——《4个方法快速打造你的阅读清单》

* Android检查更新下载安装

* 动手打造史上最简单的Recycleview 侧滑菜单

* TCP网络通讯如何解决分包粘包问题

* SDCC 2017之区块链技术实战线上峰会

* 四大线程池详解

最新评论

使用Caffe尝试DeepID

```
26     backend: LMDB
27     batch_size: 128
28 }
29 transform_param {
30     mean_file: "dataset/deepId_mean.proto"
31 }
32 include {
33     phase: TEST
34 }
35 }
36 layer {
37     name: "conv1"
38     type: "Convolution"
39     bottom: "data"
40     top: "conv1"
41     param {
42         name: "conv1_w"
43         lr_mult: 1
44         decay_mult: 0
45     }
46     param {
47         name: "conv1_b"
48         lr_mult: 2
49         decay_mult: 0
50     }
51     convolution_param {
52         num_output: 20
53         kernel_size: 4
54         stride: 1
55         weight_filler {
56             type: "gaussian"
57             std: 0.01
58         }
59         bias_filler {
60             type: "constant"
```

关闭

qq_38426572: 博主，我有一个很简单的问题。。为什么你这不用import caffe的。。。

使用Caffe尝试DeepID

Richel-Li: @xx4383:我也不是本文的网络结构，我是采用Lightened CNN 网络结构

使用Caffe尝试DeepID

Richel-Li: @f413110095:你的问题解决了，精度还是0吗？

使用Caffe尝试DeepID

Richel-Li: @f413110095:我的图像大小是 128*128*3，精度几乎也为0

使用Caffe尝试DeepID

Richel-Li: 我训练5万个batch，训练数据和验证数据和楼主的方法差不多，但是 top1 = 0.02%，to...

使用Caffe尝试DeepID

qq_26320093:
@friendwang1989:请问你的问题解决了吗？我的问题和您差不多

使用Caffe尝试DeepID

_顺其_自然: 博主您好，问一个简单的问题，请问有什么人脸识别的代码可以将使用FaceTools对齐后的数据集分成训...

使用Caffe尝试DeepID

goodluckcwl: @u014230646:最近跑了一个Sphereface，LFW有99.18%，放在我的github...

使用Caffe尝试DeepID

goodluckcwl: @tdnyb:我觉得你还是换一个网络吧，deepid这个网络能力不行。

使用Caffe尝试DeepID

xx4383: @friendwang1989: 你好！请问你使用的是本篇文章的网络结构吗？我也是128*128的...

```

61     }
62   }
63 }
64
65 layer {
66   name: "relu1"
67   type: "ReLU"
68   bottom: "conv1"
69   top: "conv1"
70 }
71 layer {
72   name: "pool1"
73   type: "Pooling"
74   bottom: "conv1"
75   top: "pool1"
76   pooling_param {
77     pool: MAX
78     kernel_size: 2
79     stride: 2
80   }
81 }
82 layer {
83   name: "conv2"
84   type: "Convolution"
85   bottom: "pool1"
86   top: "conv2"
87   param {
88     name: "conv2_w"
89     lr_mult: 1
90     decay_mult: 0
91   }
92   param {
93     name: "conv2_b"
94     lr_mult: 2
95     decay_mult: 0

```

关闭

```
96     }
97     convolution_param {
98       num_output: 40
99       kernel_size: 3
100      stride: 1
101      weight_filler {
102        type: "gaussian"
103        std: 0.01
104      }
105      bias_filler {
106        type: "constant"
107      }
108    }
109  }
110
111  layer {
112    name: "relu2"
113    type: "ReLU"
114    bottom: "conv2"
115    top: "conv2"
116  }
117  layer {
118    name: "pool2"
119    type: "Pooling"
120    bottom: "conv2"
121    top: "pool2"
122    pooling_param {
123      pool: MAX
124      kernel_size: 2
125      stride: 1
126    }
127  }
128  layer {
129    name: "conv3"
130    type: "Convolution"
```

关闭

```
131 bottom: "pool2"
132 top: "conv3"
133 param {
134   name: "conv3_w"
135   lr_mult: 1
136   decay_mult: 0
137 }
138 param {
139   name: "conv3_b"
140   lr_mult: 2
141   decay_mult: 0
142 }
143 convolution_param {
144   num_output: 60
145   kernel_size: 3
146   stride: 1
147   weight_filler {
148     type: "gaussian"
149     std: 0.01
150   }
151   bias_filler {
152     type: "constant"
153   }
154 }
155 }
156
157 layer {
158   name: "relu3"
159   type: "ReLU"
160   bottom: "conv3"
161   top: "conv3"
162 }
163 layer {
164   name: "pool3"
165   type: "Pooling"
```

关闭

```
166 bottom: "conv3"
167 top: "pool3"
168 pooling_param {
169   pool: MAX
170   kernel_size: 2
171   stride: 2
172 }
173 }
174 layer {
175   name: "conv4"
176   type: "Convolution"
177   bottom: "pool3"
178   top: "conv4"
179   param {
180     name: "conv4_w"
181     lr_mult: 1
182     decay_mult: 0
183   }
184   param {
185     name: "conv4_b"
186     lr_mult: 2
187     decay_mult: 0
188   }
189   convolution_param {
190     num_output: 80
191     kernel_size: 2
192     stride: 1
193     weight_filler {
194       type: "gaussian"
195       std: 0.01
196     }
197     bias_filler {
198       type: "constant"
199     }
200   }
```

关闭

```
201 }
202 layer {
203     name: "relu4"
204     type: "ReLU"
205     bottom: "conv4"
206     top: "conv4"
207 }
208 layer {
209     name: "fc160_1"
210     type: "InnerProduct"
211     bottom: "pool3"
212     top: "fc160_1"
213     param {
214         name: "fc160_1_w"
215         lr_mult: 1
216         decay_mult: 1
217     }
218     param {
219         name: "fc160_1_b"
220         lr_mult: 2
221         decay_mult: 1
222     }
223     inner_product_param {
224         num_output: 160
225         weight_filler {
226             type: "gaussian"
227             std: 0.01
228         }
229         bias_filler {
230             type: "constant"
231         }
232     }
233 }
234
235 layer {
```

关闭


```
236 name: "fc160_2"
237 type: "InnerProduct"
238 bottom: "conv4"
239 top: "fc160_2"
240 param {
241   name: "fc160_2_w"
242   lr_mult: 1
243   decay_mult: 1
244 }
245 param {
246   name: "fc160_2_b"
247   lr_mult: 2
248   decay_mult: 1
249 }
250 inner_product_param {
251   num_output: 160
252   weight_filler {
253     type: "gaussian"
254     std: 0.01
255   }
256   bias_filler {
257     type: "constant"
258   }
259 }
260 }
261
262 layer {
263   name: "fc160"
264   type: "Eltwise"
265   bottom: "fc160_1"
266   bottom: "fc160_2"
267   top: "fc160"
268   eltwise_param {
269     operation: SUM
270 }
```

关闭

```
271 }
272 layer {
273   name: "dropout"
274   type: "Dropout"
275   bottom: "fc160"
276   top: "fc160"
277   dropout_param {
278     dropout_ratio: 0.4
279   }
280 }
281
282 layer {
283   name: "fc_class"
284   type: "InnerProduct"
285   bottom: "fc160"
286   top: "fc_class"
287   param {
288     name: "fc_class_w"
289     lr_mult: 1
290     decay_mult: 1
291   }
292   param {
293     name: "fc_class_b"
294     lr_mult: 2
295     decay_mult: 1
296   }
297   inner_product_param {
298     num_output: 10499
299     weight_filler {
300       type: "gaussian"
301       std: 0.01
302     }
303     bias_filler {
304       type: "constant"
305     }
```

关闭

```

306 }
307 }
308 layer {
309   name: "loss"
310   type: "SoftmaxWithLoss"
311   bottom: "fc_class"
312   bottom: "label"
313   top: "loss"
314 }
315 layer {
316   name: "accuracy"
317   type: "Accuracy"
318   bottom: "fc_class"
319   bottom: "label"
320   top: "accuracy"
321   include {
322     phase: TEST
323   }
324 }
325 #这里注意fc160维之后不接ReLU,个人在这里吃了亏,因为拿fc160做特征时,用了ReLU就将负

```

下面是deepId_solver.prototxt

```

1  net: "deepId_train_test.prototxt"
2  # conver the whole test set. 484 * 128 = 62006 images.
3  test_iter: 484
4  # Each 6805 is one epoch, test after each epoch
5  test_interval: 6805
6  base_lr: 0.01
7  momentum: 0.9
8  weight_decay: 0.005
9  lr_policy: "step"
10 # every 30 epochs, decrease the learning rate by factor 10.
11 stepsize: 204164

```

关闭

```
12 gamma: 0.1
13 # power: 0.75
14 display: 200
15 max_iter: 816659 # 120 epochs.
16 snapshot: 10000
17 snapshot_prefix: "trained_model/deepId"
18 solver_mode: GPU
```

训练数据组织

数据来自CASIA-Webface的切图（人脸对齐，缩放到一个固定的比例，比如55*55），CASIA-Web10575个人，每个人的图片数量从几十到几百不等。类别数量之间严重不平衡，这里，我试验了两种有用全部的10575类，如deepId_train_test.prototxt定义的那样，只使用了10499类，如果某类别的图片少，就不使用它）：

1. 让训练集的每一类数目完全一样，比如我的实验中为训练集每个人50张图片。
2. 图片数目最多的那一类于最少的哪一类的数目比例不超过3：1（个人愚见，不知是否有道理，请高手指点一二）。

验证集是从每一类别的都随机挑选一部分不在训练集中的图片来做验证。

-下面的prepare_deepId_data.py是我组织训练测试数据的python代码，仅供参考（高手请拍砖）。

```
1
2 import os
3 from random import shuffle
4 import cPickle
5 def check_parameter(param, param_type, create_new_if_missing=False):
6     assert param_type == 'file' or param_type == 'directory'
7     if param_type == 'file':
8         assert os.path.exists(param)
9         assert os.path.isfile(param)
10    else:
```

[关闭](#)

```
11     if create_new_if_missing is True:
12         if not os.path.exists(param):
13             os.makedirs(param)
14         else:
15             assert os.path.isdir(param)
16     else:
17         assert os.path.exists(param)
18         assert os.path.isdir(param)
19
20
21 def listdir(top_dir, type='image'):
22     # type_len = len(type)
23     tmp_file_lists = os.listdir(top_dir)
24     file_lists = []
25     if type == 'image':
26         for e in tmp_file_lists:
27             if e.endswith('.jpg') or e.endswith('.png') or e.endswith('.bmp'):
28                 file_lists.append(e)
29     elif type == 'dir':
30         for e in tmp_file_lists:
31             if os.path.isdir(top_dir + e):
32                 file_lists.append(e)
33     else:
34         raise Exception('Unknown type in listdir')
35     return file_lists
36
37
38 def prepare_deepId_data_eq(src_dir, tgt_dir, num_threshold=50):
39     check_parameter(src_dir, 'directory')
40     check_parameter(tgt_dir, 'directory', True)
41     if src_dir[-1] != '/':
42         src_dir += '/'
43     if tgt_dir[-1] != '/':
44         tgt_dir += '/'
45     class_lists = listdir(src_dir, 'dir')
```

关闭

```
46 print '# class is : %d' % len(class_lists)
47 class_table = {}
48
49 num = 0
50
51 for e in class_lists:
52     assert e not in class_table
53     class_table[e] = listdir(".").join([src_dir, e]), 'image')
54     if len(class_table[e]) > num_threshold:
55         num += 1
56 print 'There are %d people whose number of images is greater than %d.' % (num, num_threshold)
57 print 'Use %d num people to train the deepId net..' % num
58 train_set = []
59 test_set = []
60 label = 0
61 dirname2label = {}
62 for k, v in class_table.iteritems():
63     if len(v) >= num_threshold:
64         shuffle(v)
65         assert k not in dirname2label
66         dirname2label[k] = label
67         i = 0
68         for i in xrange(num_threshold):
69             train_set.append((k + '/' + v[i], label))
70             i += 1
71         num_test_images = min(num_threshold / 3, len(v) - num_threshold)
72         for j in xrange(num_test_images):
73             test_set.append((k + '/' + v[i + j], label))
74         label += 1
75 f = open(tgt_dir + 'dirname2label.pkl', 'wb')
76 cPickle.dump(dirname2label, f, 0)
77 f.close()
78 f = open(tgt_dir + 'deepId_train_lists.txt', 'w')
79 for e in train_set:
80     print >> f, e[0], ' ', e[1]
```

关闭

```
81     f.close()
82     f = open(tgt_dir + 'deepId_test_lists.txt', 'w')
83     for e in test_set:
84         print >> f, e[0], ' ', e[1]
85     f.close()
86
87
88 def prepare_deepId_data_dif(src_dir, tgt_dir, num_threshold=20, add_all=False):
89     check_parameter(src_dir, 'directory')
90     check_parameter(tgt_dir, 'directory', True)
91     if src_dir[-1] != '/':
92         src_dir += '/'
93     if tgt_dir[-1] != '/':
94         tgt_dir += '/'
95     class_lists = listdir(src_dir, 'dir')
96     print '# class is : %d' % len(class_lists)
97     class_table = {}
98
99     num = 0
100
101     for e in class_lists:
102         assert e not in class_table
103         class_table[e] = listdir(".".join([src_dir, e]), 'image')
104         if len(class_table[e]) > num_threshold:
105             num += 1
106     print 'There are %d people whose number of images is greater than %d.' % (num, num_threshold)
107     print 'Use %d num people to train the deepId net, we do not care the validation set result...' % num
108     train_set = []
109     test_set = []
110     label = 0
111     dirname2label = {}
112     for k, v in class_table.iteritems():
113         if len(v) >= num_threshold:
114             shuffle(v)
115             assert k not in dirname2label
```

关闭

```
116     dirname2label[k] = label
117     i = 0
118     for i in xrange(num_threshold):
119         train_set.append((k + '/' + v[i], label))
120         i += 1
121     j = 0
122     num_test_images = min(int(num_threshold / 3), len(v) - num_threshold)
123     for j in xrange(num_test_images):
124         test_set.append((k + '/' + v[i + j], label))
125
126     if len(v) > num_threshold + num_test_images:
127         offset = j + 1 + i
128         if add_all is False:
129             # add the rest of all images or 3 times the num_threshold images to the training
130             num_left = len(v) - num_threshold - num_test_images
131             num_left = min(num_left, num_threshold)
132
133             for ii in xrange(num_left):
134                 train_set.append((k + '/' + v[ii + offset], label))
135         else:
136             # print 'Adding the rest of all data into training set.'
137             while offset < len(v):
138                 train_set.append((k + '/' + v[offset], label))
139                 offset += 1
140         label += 1
141     f = open(tgt_dir + 'dirname2label.pkl', 'wb')
142     cPickle.dump(dirname2label, f, 0)
143     f.close()
144     f = open(tgt_dir + 'deepId_train_lists.txt', 'w')
145     for e in train_set:
146         print >> f, e[0], ' ', e[1]
147     f.close()
148     f = open(tgt_dir + 'deepId_test_lists.txt', 'w')
149     for e in test_set:
150         print >> f, e[0], ' ', e[1]
```

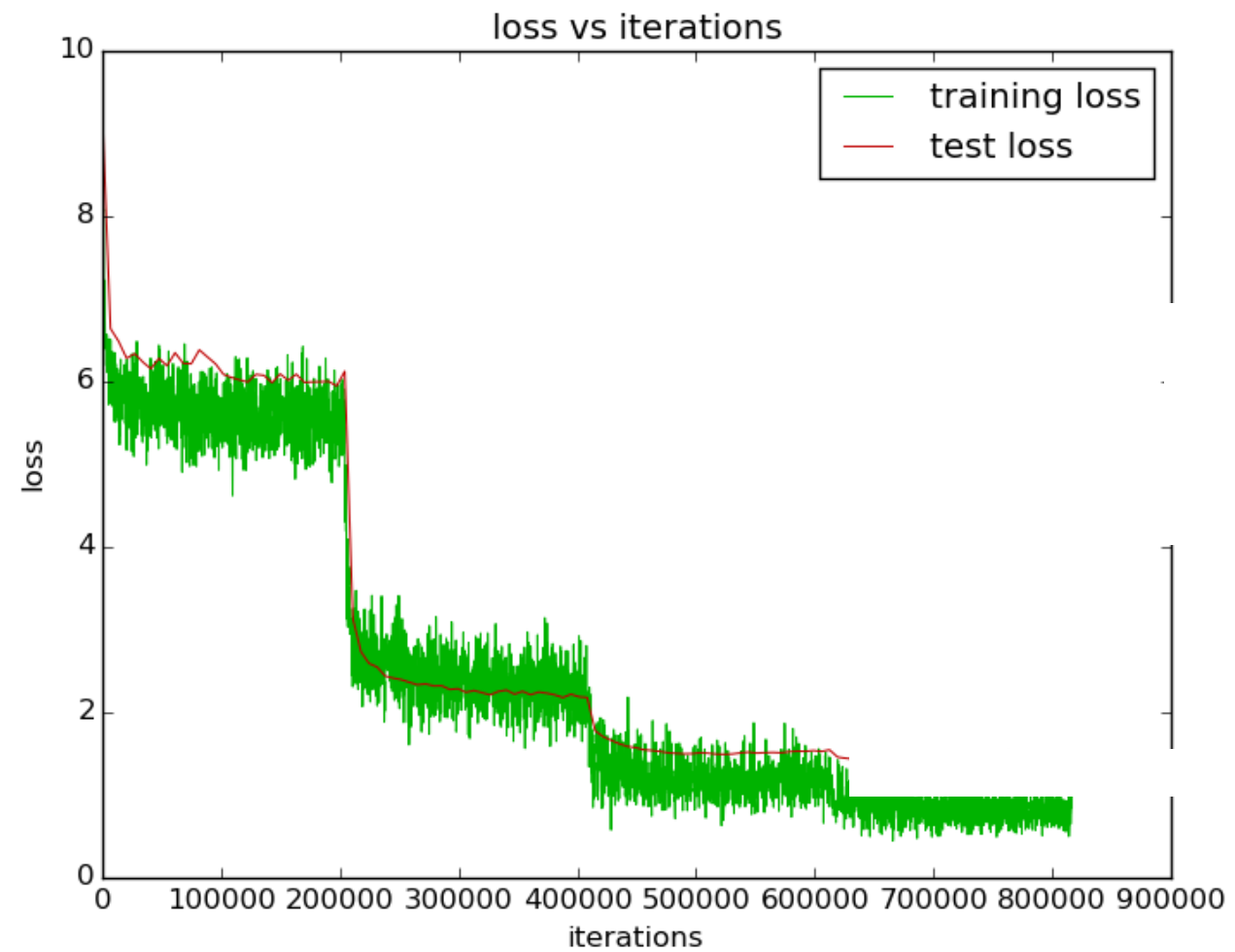
关闭


```
151     f.close()
152
153     if __name__ == '__main__':
154         prepare_deepId_data_eq('CASIA-Webface/', 'dataset', 50)
155         prepare_deepId_data_dif('CASIA-Webface/', 'dataset', 20, True)
156     #后缀eq表示每一类数目一样，50表示希望每一类都有50幅图片，dif每一类数目不一样。
```

实验结果

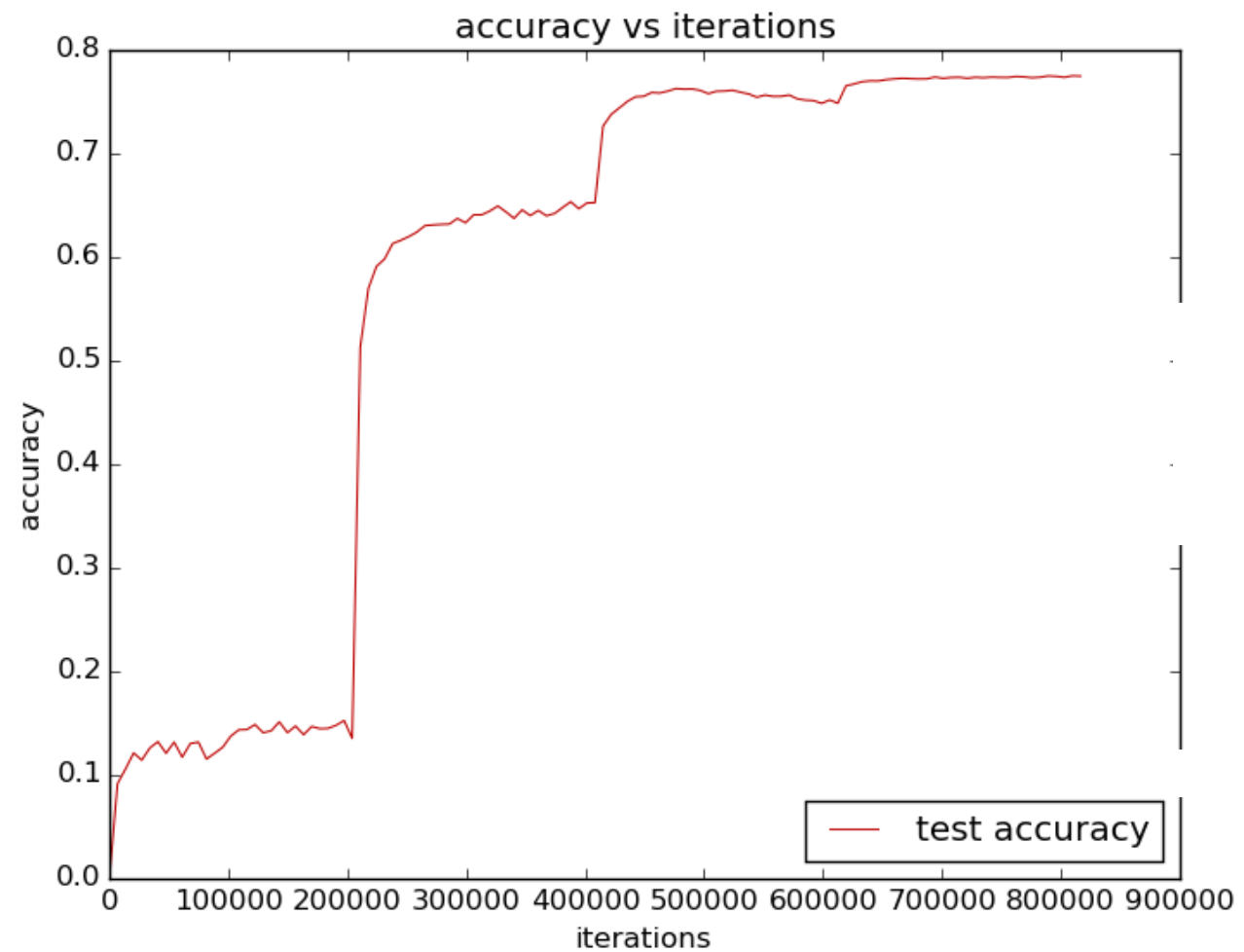
实验过程是抽取已经训练好了的模型，将lfw的测试数据抽取特征fc160维的特征，然后对特征用cc joint bayesian距离来做人脸验证。

-训练过程loss曲线如下:



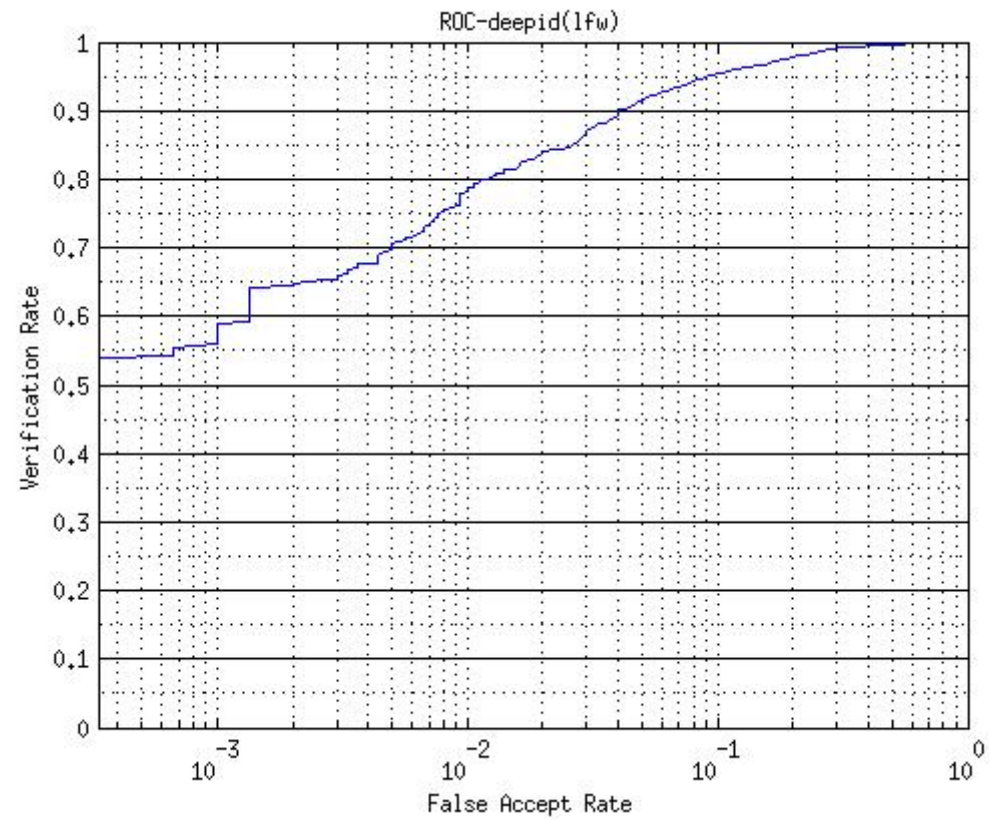
关闭

-训练过程的accuracy曲线如下:

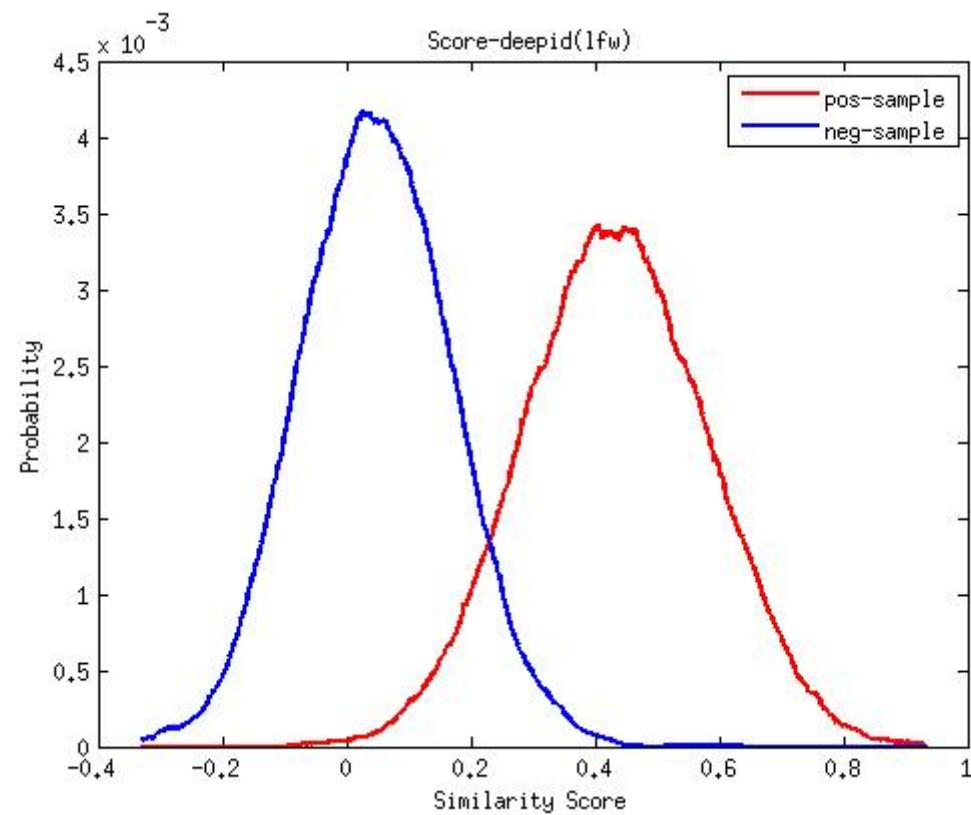


-使用cos距离度量在LFW上的roc曲线以及正负样本分布图

关闭

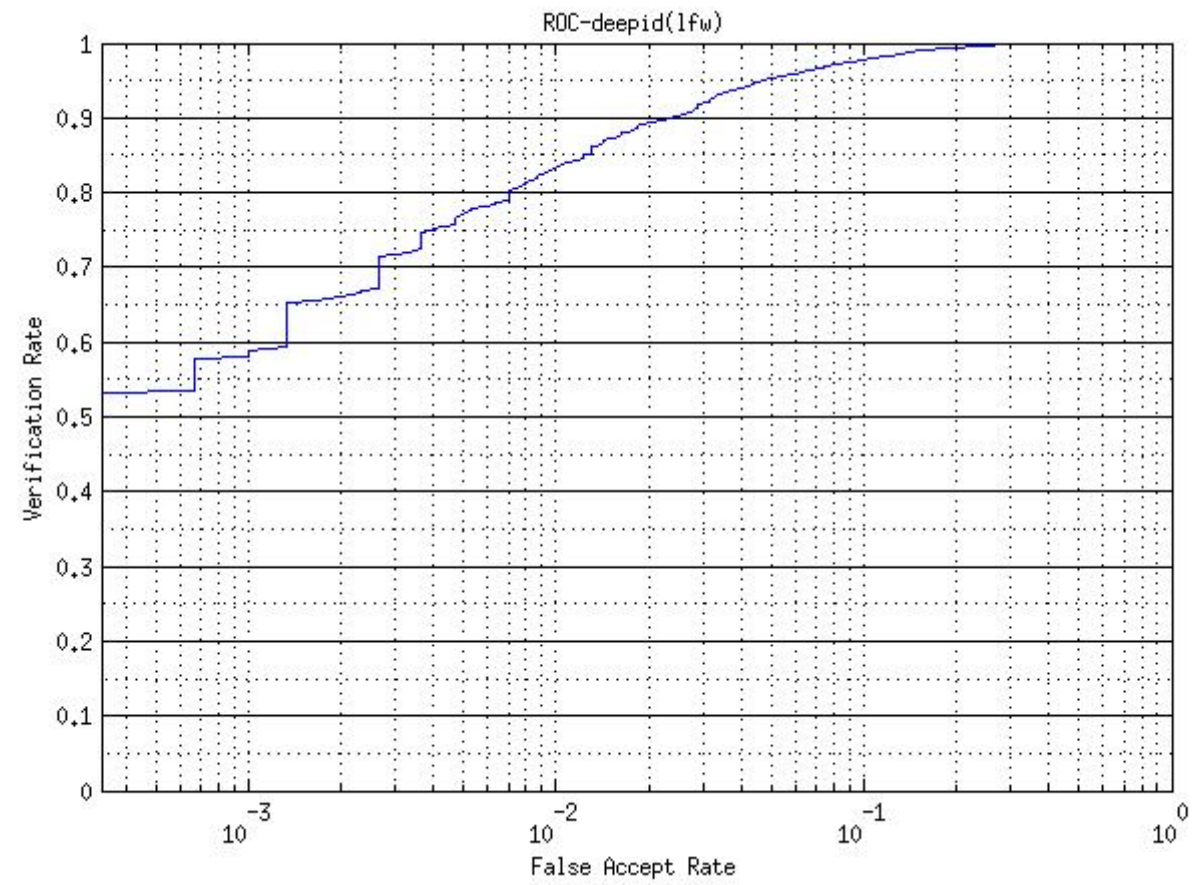


关闭

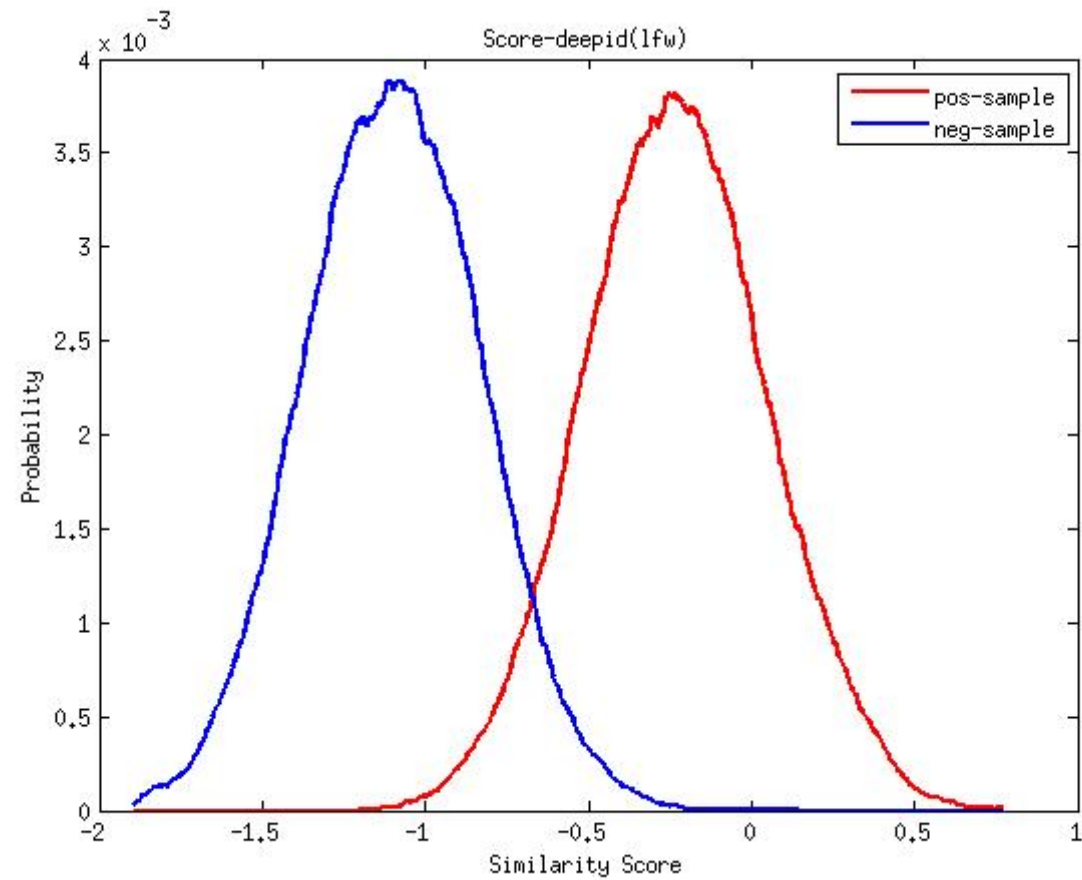


-使用joint bayesian在LFW上的roc曲线以及正负样本分布图

关闭



关闭



-在LFW上的单part模型结果如下：

metric	mean accuracy	std
cos	0.9395	0.0035
joint bayesian	0.9545	0.0045

实验结果分析说明

关闭

本实验单模型只有95.45%的准确率，没有到97%左右（实验室师兄用convnet复现得到的单part的准确率），存在如此大的差距。一方面还是参数没调好，训练的不够好，数据组织欠妥，另一方面也许是deepID的第3个conv层和第4个conv层用的local卷积，人脸不同区域用不同filter来提取特征能得到更加丰富的特征？

ps: caffe的local卷积太慢了，有点不能忍。话说happyneer大神deepID的LFW上了97.17，不得不佩服大神调参能力,还是自己太菜了。

顶 踩

6

2

上一篇 《Deep Learning Face Representaion from Predicting 10000 Classes》读书报告

下一篇 Caffe在Cifar10上复现ResNet

相关文章推荐

- 【Caffe实践】基于Caffe的人脸识别实现
- 30天系统掌握机器学习--唐宇迪
- 使用Caffe复现DeepID实验
- 【免费】XGBoost模型原理及其表现--卿来云
- Caffe —— Deep learning in Practice
- 全能项目经理训练视频
- VGG-16 prototxt
- 如何进行finetune
- Android实战基础知识
- DeepID Generation I
- 机器学习需要的掌握的数学知识汇总
- VGG 16 prototxt
- Caffe学习（十）protobuf及caffe.proto解析
- Faster-RCNN + OHEM + vgg16 + prototxt文件及...

关闭

[Python网络爬虫快速入门指导](#)[caffe+python 使用训练好的VGG16模型 对 单张图...](#)

查看评论

52楼 [qq_38426572](#) 2017-08-25 17:24发表



博主，我有一个很简单的问题。。为什么你这不用import caffe的。。。

51楼 [Richel-Li](#) 2017-08-14 19:09发表



我训练5万个batch，训练数据和验证数据和楼主的方法差不多，但是 top1 = 0.02%, top5=0.17%,精度很低，不知道什么原因？

50楼 [_顺其_自然](#) 2017-08-03 11:06发表



博主您好，问一个简单的问题，请问有什么人脸识别的代码可以将使用FaceTools对齐后的数据集分成训练和测试灰度处理的代码吗？我还是小白，感谢

49楼 [qq_31110769](#) 2017-07-19 09:49发表



没话说，仅想表示感谢！

48楼 [tdnyb](#) 2017-07-04 13:36发表



楼主，我用WebFace数据库和你的网络结构，基本上达到了跟你相同的效果，但是验证集的accuracy到60%多久」

ROC和LFW上的accuracy基本上跟楼主相差不多，在此基础上我在楼主的网络结

化，一处是在Relu1和Relu2后接了LRN层，从训练结果来看，验证accuracy达到72%，loss降到1左右，按这个结果应该是比我之前的训练结果要好一些，但是在LFW上测试时，发现所有数据提取出来的FC160基本上都是一样的，对于这个问题非常困惑，不知道楼主有没有遇到这个问题？

Re: [goodluckcwl](#) 2017-07-31 17:04发表



回复tdnyb：我觉得你还是换一个网络吧，deepid这个网络能力不行。

Re: [mikaël_hu](#) 2017-07-05 09:48发表



回复tdnyb：您好，我用了Webface数据集和楼主的网络，训练效果跟楼主差不多：验证：0.75，loss：1.75，训练：loss：0.2左右，但是在LFW上的accuracy只有0.8左右，很低了。不知您有没有做过什么特殊的处理？多谢！

47楼 [friendwang1989](#) 2017-06-28 12:04发表

关闭



楼主好，我用的webface数据集，采用了港中文的进行对齐，以及进行灰度处理，webface为144*144，在网络中被crop到128*128,而lfw也进行了对齐并弄成灰度的128*128。用了1万来个人，总共30万张webface图片进行训练，结果是训练效果确实train的identify accuracy大概85%，loss为0.7。validation的identify accuracy达到了75%,loss为1.5.应该算训练的比较好的网络了吧。。但是放到lfw上面，verification accuracy只有70%左右。。。可以帮分析一下么？谢谢

Re: [xx4383](#) 2017-07-21 11:07发表



回复friendwang1989：你好！请问你使用的是本篇文章的网络结构吗？我也是128*128的输入，效果不好

Re: [Richel-Li](#) 2017-08-14 20:17发表



回复xx4383：我也不是本文的网络结构，我是采用 Lightened CNN 网络结构

46楼 [friendwang1989](#) 2017-06-28 12:01发表



楼主好，我用的webface数据集，采用了港中文的进行对齐，以及进行灰度处理，webface为144*144，在网络中128*128,而lfw也进行了对齐并弄成灰度的128*128。用了1万来个人，总共30万张webface图片进行训练，结果是train的identify accuracy大概85%，loss为0.7。validation的identify accuracy达到了75%,loss为1.5.应该算训练的吧。。但是放到lfw上面，verification accuracy只有70%左右。。。可以帮分析一下么？谢谢

Re: [qq_26320093](#) 2017-08-04 10:17发表



回复friendwang1989：请问你的问题解决了吗？我的问题和差不多

45楼 [悼念者的天堂](#) 2017-06-20 16:37发表



你好，博主。

我使用你的方法组织数据时，发现超过50张的人，只有两千多，超过20张的只有六千多。

而你博文说使用了10499类，这是为什么？

我用的webface-maxpy-clean 版。

Re: [mikael_hu](#) 2017-06-21 10:56发表



回复悼念者的天堂：他用的是全部的，看2楼的回复

Re: [悼念者的天堂](#) 2017-06-22 15:11发表



回复mikael_hu：没注意到，谢谢了。

关闭

44楼 [mikael_hu](#) 2017-06-19 11:10发表



您好，根据您的网络和方法训练的，训练到100w，loss降到了0.1--0.2（在训练到50W次后，loss就降到了0.2左右），test accuracy，只有0.756，test loss 为1.75。不过，在LFW上测试的accuracy只有0.789，很低。LFW的图片预处理跟训练集（CASIA-Webface）的预处理一样（用的是港中大的人脸对齐工具：http://mmlab.ie.cuhk.edu.hk/archive/CNN_FacePoint.htm），都是经过人脸对齐，再crop到55x47大小的jpg图片。请问accuracy上不去的原因在哪呢？多谢！

43楼 [BABY](#) 2017-01-23 19:41发表



看原论文每张图分了很多patch，请问作者有分patch训练吗？

Re: [天妒WS](#) 2017-01-24 09:47发表



回复BABY：我把25个patch全部切了来训练过，lfw上最终到了98.99。

Re: [pervertsennin](#) 2017-03-03 11:53发表



回复天妒WS：你好，请问这个patch怎么分啊，我是直接用CAISA的数据将人类抠出来，然后效果不是很好

Re: [pervertsennin](#) 2017-03-03 11:52发表



回复天妒WS：你好，请问这个patch怎么分啊，我是直接用CAISA的数据将人类抠出来，然后效果不是很好

Re: [BABY](#) 2017-01-24 13:55发表



回复天妒WS：给力！也就是说本文的结果是单patch啦！

42楼 [AnMe90](#) 2017-01-13 16:08发表



您好，我用的casia-webface数据，并进行了对齐，只对数据进行了镜像增广，单模型cos在LFW上只有87%精度，在没有数据增加的情况下这样正常吗？谢谢。

Re: [天妒WS](#) 2017-01-13 18:45发表



回复AnMe90：非常不正常，没训练好

Re: [AnMe90](#) 2017-01-17 10:31发表

关闭



回复天妒WS：谢谢博主解答，我再尝试下修改下训练策略，也可能是数据清洗方面的问题。

Re: [天妒WS](#) 2017-01-17 15:50发表



回复AnMe90：虽然说CASIA有噪声，不过我做的实验结果表明，在这种小数据库上出来的结果都差不多，因此清洗CASIA并不是特别重要的事情，你大胆用它，注意预处理挺重要的。

Re: [AnMe90](#) 2017-01-17 16:58发表



回复天妒WS：这个最后的测试精度高，估计是我增广数据（模糊、变换等）后的原因？博主说的预处理，只是指的对齐？还是也包括滤波、标准化等？按照理论讲，DeepID的测试日志中最后的Test accuracy一般是0.8以下，我的却每次都到了0.9以上。谢谢释疑。

Re: [AnMe90](#) 2017-01-17 16:22发表



回复天妒WS：我是随机取了两张作为验证集，其它的作为训练集，转为55*47大小的网络，最后的Test accuracy=0.999，这感觉拟合的太好了吧。LFW上就只有87%。理，对齐用的Dlib自带的特征点对齐裁剪。谢谢

Re: [天妒WS](#) 2017-01-24 10:00发表



回复AnMe90：对齐指的是根据人脸关键点，进行一些旋转缩放等操作，将人脸校正到你预先定的位置。比如根据两眼，让两眼的位置在最终输出图片上水平位置，那么如果原图眼睛不是水平的，首先就得将原图旋转，使得两眼基本水平，然后再抠图。

Re: [AnMe90](#) 2017-02-08 15:54发表



回复天妒WS：谢谢您的帮助，现在的单模型已经达到0.96以上，只是自己用摄像头捕捉人脸测试了模型的性能，对正脸的比对结果还是很不错的，对侧脸与正脸，正脸和仰视等的比对结果就差了很多了（指的是相似度上，正脸之间或者正脸与小角度侧脸相似度都在75%以上，大角度侧脸与正脸进行比对就差了很多，有时候就不能判断是否是一个人了），请问这个DeepID的模型在现实中测试就是这样吗？谢谢

Re: [friendwang1989](#) 2017-06-28 11:17发表



回复AnMe90：请问你Lfw处理也跟webface一模一样么

Re: [friendwang1989](#) 2017-06-28 11:16发表

关闭



回复AnMe90：请问你Lfw做处理了么。跟webface处理的一模一样？

Re: 天妒WS 2017-02-15 17:44发表



回复AnMe90：你的训练集里面并没有这种大姿态的图片吧，训练集都小姿态图片，因此训练出来的模型对这种大姿态处理不会很高的。

41楼 f413110095 2016-12-27 16:06发表



楼主，你好 我用了你的网络参数和结构，我用的也是casia的人脸数据库，可是我训练的时候为何acc一直是0，loss也一直在8到9之间徘徊，不过我的人脸没缩放到55*55 而是227*227，不过这个和图片大小有关系吗

Re: Richel-Li 2017-08-14 19:13发表



回复f413110095：你的问题解决了么，精度还是0吗？

Re: Richel-Li 2017-08-14 19:12发表



回复f413110095：我的图像大小是 128*128*3，精度几乎也为0

Re: 天妒WS 2017-01-12 11:13发表



回复f413110095：那么大的图，这么浅的网络，不太合适。

40楼 xslora 2016-11-23 11:26发表



博主，威武，膜拜膜拜！！

39楼 paopaoxr 2016-11-16 10:13发表



楼主你好，人脸验证准确率95.45%是怎么计算出来的啊？有代码参考吗？

Re: 天妒WS 2016-11-23 11:18发表



回复paopaoxr：LFW的有个View2测试协议，有6000对测试比对，分10组，每组600对，300对正样本，300对负样本，然后分别在这10组上测试，共得到10组分数，然后取平均值就得出来了。博客中的结果应该是一个非常差的结果。

38楼 小_小_杨_ 2016-11-07 16:52发表

关闭



楼主，你好，请问一下您使用的人脸对齐是怎么做的，用的什么算法，还是说数据库有提供人脸关键点？

Re: 天妒WS 2016-11-23 11:20发表



回复小_小_杨_：用deepID论文中提到的sdm关键点对齐算法，这个港中文应该是公布源代码了的。

37楼 [goodluckcwl](#) 2016-10-22 20:16发表



楼主，我现在训练准确率只有65%左右，联合贝叶斯只有91.5%.训练如何优化？训练数据再做直方图均衡怎么样

Re: [goodluckcwl](#) 2017-07-31 17:08发表



回复goodluckcwl：最近跑了一个Sphereface，LFW有99.18%,放在我的github上了。

<https://github.com/goodluckcwl>

Re: [goodluckcwl](#) 2016-11-22 13:23发表



回复goodluckcwl：97.73%

Re: [pervertsennin](#) 2016-12-21 10:52发表



回复goodluckcwl：你好，我现在这个人了对齐弄不好，能赐教一下吗

Re: 天妒WS 2016-11-23 11:19发表



回复goodluckcwl：97.73是多模型融合的还是单模型，是deepID结构，正常来说，97.73在LFW上算是一个中规中矩的结果。不好不差。博客中的结果算是很差的结果了，我很久之前做的。

Re: [goodluckcwl](#) 2016-11-25 11:11发表



回复天妒WS：最近在看Ian Goodfellow的Deep learning这本书，有一些收获。然后之前的人脸识别是做项目，我现在在做face alignment。得赶紧想办法发论文，好毕业。

Re: [goodluckcwl](#) 2016-11-25 11:04发表



回复天妒WS：其实我也有很多不懂。我觉得DeepID这个网络结构太简单了。如果不用local conv的话。我最后用了两个model融合的。

Re: [goodluckcwl](#) 2016-11-20 14:09发表

关闭



回复goodluckcwl : 97.52%

Re: [goodluckcwl](#) 2016-11-18 10:15发表



回复goodluckcwl : 到了97.2%

Re: [friendwang1989](#) 2017-06-28 11:34发表



回复goodluckcwl : lfw和webface的操作一模一样？

Re: [friendwang1989](#) 2017-06-28 11:33发表



回复goodluckcwl : 请问哥们是分别怎么做webface和lfw的对齐的？

Re: [goodluckcwl](#) 2016-11-17 09:42发表



回复goodluckcwl : 到95%以上了，哈哈

36楼 [Aihahafox](#) 2016-09-29 10:30发表



你好，按照你的网络结构，我在mxnet上实现后，训练的分类准确率有80左右，但是提取出来的160-d特征在进行分类时也无法很好区分，只有55%左右的准确率，我想请问一下特征提取后关于Joint Bayesian的特征预处理时怎么做的

Re: [天妒WS](#) 2016-10-08 13:17发表



回复Aihahafox : 训练数据的预处理很重要，你对齐了吧。

Re: [Aihahafox](#) 2016-10-09 13:45发表



回复天妒WS : 现在cos可以达到90.6%，JB可以达到89.6%，单还是没有达到最好的效果，接下来可以在哪里提升？

Re: [天妒WS](#) 2016-10-11 18:38发表



回复Aihahafox : cos也很差啊，如果说是采用deepID结构的话，cos也能到95.x，说明你的训练还是有问题，可能就是训练数据的原因。训练数据一定要根据关键点校准对齐，这样任务就变得简单了，另外，你用的应该也是CASIA吧？

Re: [goodluckcwl](#) 2016-10-21 15:04发表

关闭



您好，请问数据对齐时是做相似变换，可是侧脸和正脸如何对齐呢？是都把眼睛放在55*47的人脸图像的固定位置吗？这样侧脸的图像是不是包含了更多的背景，怎么处理呢

35楼 [小娃娃妮](#) 2016-09-09 15:35发表



楼主你好，按照我在CASIA_WEB上，按照你说的数据组织方式人脸对齐后进行划分训练集和测试集，crop人脸框，使用灰度图按照你的网络进行训练，最后到100W次的时候，test accuracy也只到68%，后来调低了学习率反复训练到200W，test_accuracy准确率也没有提升，与你相差很多，loss率一直在2.0左右，请大神帮忙分析一下未能到达80%的原因，谢谢！

Re: [天妒WS](#) 2016-10-08 13:19发表



回复小娃娃妮：灰度图你做了什么操作呢，比如减均值或者是除了255？

34楼 [小娃娃妮](#) 2016-09-09 15:20发表



你好，我的数据是CASIA-WEBface人脸对齐过后的，输入的图像crop成80*80灰度图进行训练，每一类使用两个试，其余全部用来训练，网络结构和训练参数和提供的网络的一样，但是我的测试集合上的准确率只到68%，lo在2.0上下，到达65%以上保存了model,在调低了学习率之后，fixed学习20W次以上也只提高到68%，我想请问调整学习率一个参数，还是其他参数都要调整吗？谢谢！

33楼 [lyf5oo](#) 2016-09-03 20:53发表



我看图片集中的人脸都是对齐的啊（鼻子都在中间），还需要用什么工具来对齐吗？对齐指的是把人脸扣出来吗？

Re: [天妒WS](#) 2016-10-08 13:19发表



回复lyf5oo：对齐是要抠出来，根据关键点来做的，sdm算法。

32楼 [姜鱼](#) 2016-08-26 10:26发表



楼主CASIA-Webface样本库方便共享一下吗？Google下着下着就断了，毕竟需要用这个，可以的话加我qq呗，qq号：348132030。

Re: [天妒WS](#) 2016-09-01 18:15发表



回复姜鱼：这个数据库需要申请。不能私自传哦。你可以让你让老师申请吧。

31楼 [yueyuecsdn](#) 2016-08-22 14:32发表

关闭



楼主你好，我用的是caffe在训练数据，然后需要train文件夹和test文件夹及其levdel文件，那么由原始的数据集怎样生成这两个文件夹呢，求楼主

Re: [天妒WS](#) 2016-08-22 20:03发表



回复yueyuecsdn：说实话，不是特别能够理解你的问题。你是说如何划分训练测试还是指怎么用程序生成呢？

Re: [yueyuecsdn](#) 2016-08-23 14:53发表



回复天妒WS：问题已经解决，么么哒

30楼 [wjxzu](#) 2016-07-24 21:35发表



您好，很感谢这篇文章的指导，我在复现时，采用的是CASIA_WEBFACE的数据库，我对CASIA_WEBFACE进行处理，将图像归一化为55x47大小，利用您的网络训练时，test_accuracy一直不高，目前在60%-70%左右，跟您的10%，利用lfw数据库进行测试，效果也不是很好，我对您文中提到的让训练集每一类数目完全一样有点疑问，因相当一部分人的图片数目少于50张，我统计过大约有快一半的人数，那么怎么对这些人进行扩充满足条件呢，我并没有进行这种保证每人图片数均衡的操作，只是简单的划分训练集和验证集时采用了9:1的比例（另外我还去删少，<10张的人数），是不是每人图片均衡一点，效果会更好呢？还望您能解答一下，谢谢！

Re: [天妒WS](#) 2016-07-27 23:33发表



回复wjxzu：1、确保你的训练数据的人脸是对齐的；2、组织数据时，不必按照这篇博客的来，这个实验做的时间比较久了，后来又实验时，直接每一类取1~2张人脸作为验证集，其余的全部作为训练集。3、对齐后的，要按照分类任务那样crop做数据增强，仅仅做个水平 镜像即可。

29楼 [lyf5oo](#) 2016-07-21 22:39发表



感谢！这篇文章对我帮助很大！但当我用这个网络实际训练时却是如下的输出：

```
Solving deepID_network
Learning Rate Policy: step
Iteration 0, Testing net (#0)
Test net output #0: accuracy = 0
Test net output #1: loss = 9.26645 (* 1 = 9.26645 loss)
Iteration 0, loss = 9.26654
Train net output #0: loss = 9.26654 (* 1 = 9.26654 loss)
Iteration 0, lr = 0.001
```

关闭

Iteration 50, loss = -8.41308e-08
Train net output #0: loss = 0 (* 1 = 0 loss)
Iteration 50, lr = 0.001
Iteration 100, loss = -8.41308e-08
。 。 。
Iteration 2000, loss = -8.41308e-08
然后一直是这样，loss = -8.41308e-08

不知为啥这样，请指教，谢谢！

Re: [天妒WS](#) 2016-07-27 23:29发表



回复lyf500：CNN一般是需要大量数据才能训练好的，如果针对一个特定任务可用数据量太小的话，可一个较为相似的较大规模的数据库上预训练，然后Finetune这个小数据库。

Re: [天妒WS](#) 2016-07-22 13:30发表



回复lyf500：lfw人脸太少了，CNN需要大数据量的，建议使用CASIA-Webface

Re: [lyf500](#) 2016-07-23 07:17发表



回复天妒WS：就是CASIA-Webface下不到，看页面上说需要大学申请，不对个人:(

Re: [lyf500](#) 2016-07-21 22:40发表



回复lyf500：我用的是lfw的人脸库训练的。

28楼 [cxc1992](#) 2016-07-16 14:42发表



请问博主使用的是什么型号的GPU 迭代训练一共花了多长时间 我用Titan X迭代30万次 预估时间要200多天。。。。。。请问调整哪些系数能加快训练 谢谢

Re: [天妒WS](#) 2016-07-17 11:08发表



回复cxc1992：你一定是哪里搞错了吧，我用的也是titanx，训练3个多小时就完成了。

27楼 [纪阴阳](#) 2016-07-14 13:49发表

关闭



楼主你好，我又继续来烦你了，不好意思。。

关于LFW的测试过程，博文里提的不多，楼主简要说一下可以么？主要是测试数据怎么组织的？谢谢！

Re: [天妒WS](#) 2016-07-15 15:52发表



回复纪阴阳：测试数据就是LFW的View2协议吧，10折交叉验证，评估算法效果。

Re: [yueyuecsdn](#) 2016-08-23 21:35发表



回复天妒WS：楼主，有关这个view2协议，有什么相关文档和实现代码可以参考吗？

Re: [纪阴阳](#) 2016-07-18 12:38发表



回复天妒WS：多谢楼主哈，我还想再请教一下，lfw的pair comparison测试，楼主是用pycaffe matcaffe自己写的么？如果是的话，图片的读取和预处理应该也是自己做的吧？请问除了减去还需要什么预处理操作呢。。如果不是用pycaffe自己写的，那是怎么做的呢？我之前train是月不过看来它应该没法搞pair comparison。楼主是又自己写了一个test的prototxt吗？多谢了！

26楼 [bomberggggg](#) 2016-07-11 10:05发表



楼主你好，你说的55*47不太明确，是高55，宽47吗？另外，我用webface对齐加载剪后的图片训练，lfw测试精度.....，训练的精度只有50%多，和你的结果差别很大，我估计是样本有问题，我想楼主能不能把你用于训练的样本（对齐打包成百度云发给大家，多谢了！

Re: [纪阴阳](#) 2016-07-14 10:17发表



回复bomberggggg：层主你好，我情况跟你差不多啊，我是912个类，每个类60-100张不等，尺寸250*250（filter size跟着相应调整），但是调了很久参数准确率也只有62%.....

而且我非常不能理解博主的loss/accuracy曲线为什么会有那种突降，是个step down的学习率相关吗还是为什么

25楼 [bomberggggg](#) 2016-07-11 10:05发表



楼主你好，你说的55*47不太明确，是高55，宽47吗？另外，我用webface对齐加载剪后的图片训练，lfw测试精度只有82%，训练的精度只有50%多，和你的结果差别很大，我估计是样本有问题，我想楼主能不能把你用于训练的样本（对齐裁剪后的）打包成百度云发给大家，多谢了！

24楼 [cxc1992](#) 2016-07-07 15:27发表



你好，请问训练好model后如何计算余弦距离或者贝叶斯距离来测试？

关闭

Re: [天妒WS](#) 2016-07-09 23:09发表



回复cxc1992：你训练好了模型，然后将两幅待比对的人脸图片进行特征提取，比如deepID的160维特征向量，你就得到了两个160维特征向量，这两个特征向量就可以计算cos距离了。

23楼 [纪阴阳](#) 2016-07-01 11:28发表



楼主你好，我对你的文章的理解是这样的，不知道对不对

你的95%的正确率是verification的正确率，也就是说任务是给两张一个人或者不同人的照片然后判断是否是一个人。但是之前训练模型时是按照分类任务进行训练的，这个训练的准确率应该没有95%这么高吧？请问大概是多少？

Re: [天妒WS](#) 2016-07-04 13:10发表



回复纪阴阳：训练的准确率也只有70~80%

22楼 [qq_32196403](#) 2016-06-27 10:43发表



请问博主，你的卷积核大小和stride以及pooling的size和stride是怎么选择的？

Re: [天妒WS](#) 2016-06-28 10:30发表



回复qq_32196403：pooling的size和大小一般都是按照将feature map的大小减半来设计的吧，参考了vgg的这个实验除了pooling2以外，基本与deepID1的论文结构保持一致。

21楼 [小古东阿杜](#) 2016-06-23 09:45发表



你好，请问最后用于分类的160特征保存在哪边，如何提取出来进行分类呢，非常.....

Re: [天妒WS](#) 2016-06-28 10:32发表



回复小古东阿杜：那个你看看tools/extract_features.cpp的源代码就清楚了吧。

Re: [天妒WS](#) 2016-06-24 10:02发表



回复小古东阿杜：提取特征你可以用tools文件夹下面的extract_features.cpp或者自己写一个。特征提取过程就是网络在测试状态下的一次前向传播过程。

Re: [小古东阿杜](#) 2016-06-24 16:07发表

关闭



回复天妒WS：能说的再详细一点吗，刚接触caffe，还不是太熟悉，怎么利用您说的.cpp文件提取网络的特征向量呢，谢谢！

20楼 qq_32196403 2016-05-27 15:56发表



就是楼主有像DeepID论文里将脸按照不同特征点对齐裁剪出来再训练吗？

Re: 天妒WS 2016-05-30 13:48发表



回复qq_32196403：这个数据库得向CASIA申请，对齐是用SDM检测关键点然后根据相似变换做的。具体的SDM算法可以参见网址<http://www.cnblogs.com/cv-pr/p/4797823.html>

19楼 qq_32196403 2016-05-27 15:55发表



或者楼主能给个数据链接吗？这个数据还得交申请，网上下载的就是一堆z01,z02什么的

18楼 qq_32196403 2016-05-27 15:51发表



请问楼主，那个casia_webface数据库里面webface.z01,z02是啥意思？跟那个casia_webface.zip有啥关系？还有对齐的人脸，预处理有同时对lfw和casia_webface裁剪并对齐脸吗？

17楼 刘春水 2016-05-14 18:16发表



博主您好，我最近也在复现deepid，目前还没训练完，但是发现准确率在相当长的时间停在0.5左右，我发现你给的loss和accuracy线图有一个骤升的过程，这是为什么呢？

Re: 天妒WS 2016-05-16 13:25发表



回复刘春水：曲线突变处是因为学习补偿降低了。

Re: friendwang1989 2017-06-28 12:05发表



回复天妒WS：请问学习补偿降低是值learning_rate降低么

Re: 纪阴阳 2016-07-14 10:19发表



回复天妒WS：这点不太明白，google了一下也没有找到相关解释，能烦请博主多解释一下吗？

16楼 gongxuchao001 2016-05-07 10:42发表

关闭



感谢您的回复，是这样的，我直接作cos距离的时候统计了一下，相同人脸的相似度有很多比不同人脸的还低，整体准确率也就在50%左右，我也试过别人训练的模型也都差不多是这个准确率，这是啥原因呢？是不是简单的特征比对有问题？多谢了！

Re: [天妒WS](#) 2016-05-09 18:31发表



回复gongxuchao001：我猜可能的原因：1、模型没训练好，以LFW为测试基准，看看你的模型在LFW上的准确率；2、测试的时候，训练数据是怎么预处理的，测试数据也怎么预处理；3、画出ROC曲线，选取一个最好的阈值。

15楼 [gongxuchao001](#) 2016-05-06 17:33发表



您好 我最近在复现deepid，已经训练完成，比对的时候我按照您博客里的方式提取了160维特征，直接用cos距离做的匹配，不是同一张人脸的相似度也都很高，请问是我度量的方式有问题么？

Re: [天妒WS](#) 2016-05-06 20:01发表



回复gongxuchao001：不是同一个人的分数也很高这很正常，本身这个deepID网络训练好后在LFW上的96左右，也就是说还是有那么一些不是同一个人的人脸图片被当成同一个人脸的图片，度量方式并没有而且你还得选择一个阈值，使得在LFW上的识别率最高，也就是那个ROC曲线中你选取的阈值。当两引于这个阈值为同一个人，小于时为不同的人。这个结果告诉你你大概有96%的信心相信预测结果是正确的。

14楼 [zeromike](#) 2016-05-03 19:28发表



楼主你好，我最近第一次接触caffe，图片的预处理，lmdb转换，caffe的一些细节都不太熟练，不知如何下手，楼：我一份完整code让我参考一下

13楼 [Sunshine_in_Moon](#) 2016-05-03 10:33发表



楼主你好，我看到你的test精度没有达到80%，为什么在IFW上测试能达到90%以上呢？差别为什么这么大？

Re: [天妒WS](#) 2016-05-03 10:40发表



回复Sunshine_in_Moon：那个test是在CASIA上的准确率，其实就是用来指示什么时候训练可以结束的标识而已。跟LFW上的准确率并无任何关系。

Re: [Sunshine_in_Moon](#) 2016-05-04 10:20发表



回复天妒WS：谢谢！

12楼 [xubokun1992](#) 2016-04-11 09:48发表

关闭

[\[html\]](#) [view plain](#) [copy](#) [print](#) ?

01.

楼主你好，我最近也在学习研究deepid。我目前也在caffe的平台上做这个，关于deepid1的版本论文：

Deep learning face representation from predicting 10000
这个应该是你复现的版本。我想请教一下，论文中提到的在第三层卷积层的权值局部共享(locally shared in every 2*2 regions)，然后在第四层全部不共享(totally unshared)，您是怎么实现的？我看您在con3、con4的定义中没有什么特别的地方，还是说您在python代码中实习的？

02.

还是说您没有加这两个特性，发现跑出来的结果也还不错？

Re: [天妒WS](#) 2016-04-11 10:25发表



回复xubokun1992：第3，4个卷积层我用的就是普通的卷积，没有用局部权值不共享的那种卷积层，ca种局部权重不共享的那种卷积，叫做local卷积，但是local卷积实现得不好，速度非常慢，而且大大增加所以我没有用，我发现用普通的卷积效果就不错额

11楼 [xubokun1992](#) 2016-04-11 09:46发表



楼主你好，我最近也在学习研究deepid。我目前也在caffe的平台上做这个，关于deepid1的版本论文：Deep learning face representation from predicting 10000 class，这个应该是你复现的版本。我想请教一下，论文中提到的在第三层卷积层的权值局部共享(locally shared in every 2*2 regions)，然后在第四层全部不共享(totally unshared)，您是怎么实现的？我看您在con3、con4的定义中没有什么特别的地方，还是说您在python代码中实习的？还是说您没有加这两个特性，发现跑出来的结果也还不错？

10楼 [csuwujiyang](#) 2016-04-05 15:25发表



博主您好，看评论中您说输入的图像是55x47，那这个尺寸是如何计算得到的呢？谢谢！

Re: [天妒WS](#) 2016-04-06 09:58发表



回复csuwujiyang：CASIA-Webface的人脸尺寸大约在55像素左右，考虑到人脸是类似矩形的，而且DeepID论文中也是55*47的，所以这里就用的55*47，这种尺寸的人脸包含的背景较少哦。

Re: [csuwujiyang](#) 2016-04-06 14:30发表

[关闭](#)

回复天妒WS：还有一个问题就是，我看了一下您的网络结构定义，应该就是deepid2论文中的定义结构吧，那么第二个池化层pool2的stride不应该也是2吗，为什么你的是1呢？

Re: 天妒WS 2016-04-06 14:36发表



回复csuwujiyang：原始论文中,pooling层的stride的确全部是2，这里是因为当时一时疏忽，而且，我自己做实验发现，pool2得stride设为1要比stride2的识别率高，这应该是因为增加了网络参数所致的吧。你也可以直接设为2呢。

9楼 qq_31557779 2016-03-29 14:44发表



您好我做完了对齐，在将图像数据转换为lmdb的时候我有10575类人的图像，难道要针对每一个人都写一条语句吗？有没有批量转换的方法？

Re: 天妒WS 2016-03-30 21:09发表



回复qq_31557779：转化lmdb有现成的转化程序啊/

Re: qq_31557779 2016-04-06 09:10发表



回复天妒WS：您好，能加一下您的QQ吗？有些问题还想向您请教，麻烦您了。

Re: 天妒WS 2016-04-06 14:38发表



回复qq_31557779：什么问题，请说吧，我晓得的一定及时回复。

Re: 天妒WS 2016-04-06 14:39发表



回复天妒WS：1207991086

8楼 Anti-Ichigo 2016-03-25 20:51发表



请问博主的prepare_deepid_data_dif('CASIA-Webface/', 'dataset', 20, True)中的20指什么

Re: 天妒WS 2016-03-28 13:23发表



回复Anti-Ichigo：20指的是确保每类图片不得少于20张

7楼 qq_31557779 2016-03-24 10:30发表



您好，我是一名本科生，本科毕业设计要做这个东西，我现在caffe配置好了，也申请到了李子青团队的人脸数据，但是现在很迷茫，不知道接下来该怎么做，能给些意见吗？

Re: [天妒WS](#) 2016-03-24 11:24发表



回复qq_31557779：先按照deepID论文用sdm进行人脸对齐，然后可以参考本文给出的网络结构进行训练。然后就可以得出一个还算过得去的识别率。

6楼 [qq_31557779](#) 2016-03-24 10:29发表



您好，我是一名本科生，本科毕业设计要做这个东西，我现在caffe配置好了，也申请到了李子青团队的人脸数据，但是现在很迷茫，不知道接下来该怎么做，能给些意见吗？

5楼 [qq_31557779](#) 2016-03-24 10:28发表



您好，我是一名本科生，本科毕业设计要做这个东西，我现在caffe配置好了，也申请到了李子青团队的人脸数据迷茫，不知道接下来该怎么做，能给些意见吗？

4楼 [Anti-lchigo](#) 2016-03-21 00:17发表



请问博主的prepare_deepId_data.py是干什么的？就是生成train.txt和val.txt的吗？
第2个问题是，我一直不明白DeepID产生的特征后训练分类器的是哪部分代码？

Re: [天妒WS](#) 2016-03-23 16:37发表



回复Anti-lchigo：prepare_deepId_data.py就是用来划分训练集和验证集的，deepID产生的特征你可以直接计算余弦距离或者用该特征训练一个joint Bayesian分类器。我没有贴出joint Bayesian的代码，网上有相关的python代码。

Re: [Anti-lchigo](#) 2016-03-24 23:35发表



回复天妒WS：那么deepid得到的特征保存在哪里？

3楼 [Nan_cy](#) 2016-03-07 16:53发表



你好，我用的是CASIA-Webface，这个数据库共10575类，人脸数据是自己对齐成55x47大小的RGB图像，按照楼主的网络结构训练，但不收敛，不知道为什么，望赐教,在此谢谢了！

Re: [qq_31557779](#) 2016-03-29 14:52发表



回复Nan_cy：您好我也在做这方面的东西，有些不懂的可以向您请教一下吗？我做完图像对齐后，要将图像数据转换为lmdb的时候我有10575类人的图像，难道要针对每一个人都写一条语句吗？有没有批量转换的方法？

关闭

Re: [天妒WS](#) 2016-03-09 09:56发表



回复Nan_cy：你试着确保你减了均值，然后，你试着将初始的学习率降低，比如base_lr=0.001,一般这样会收敛。

Re: [Nan_cy](#) 2016-03-16 10:31发表



回复天妒WS：好的，我试试，非常感谢

2楼 [xgbhk2](#) 2016-02-26 09:10发表



你好，文中说到：

让训练集的每一类数目完全一样，比如我的实验中为训练集每个人50张图片。

怎么我拿到的数据集里只有2500个人左右有超过50张图片？你用的是webface里面的normalize face中扩展加入n分吗？即使用那个翻倍了之后我用你的程序跑也没有10499类，是我哪里搞错了吗？

Re: [天妒WS](#) 2016-03-01 18:24发表



回复xgbhk2：我用的是CASIA-Webface，这个数据库共10575类，每一类很不均衡，当时我训练的时候经验，所以训练的时候训练集的分类不至于数量太不均衡。其实你可以忽略这篇博客的数据组织方式。我后来的数据组织方式是，将10575类中每一类随机拿出2张图片作为验证集，然后其余的全部作为训练集，然后训练。lfw上的结果是（joint bayesian）96.22%。

1楼 [nuohanfengyun](#) 2016-01-21 09:29发表



你好，我使用你搭建的网络进行训练，为什么训练不收敛呢？麻烦大神不吝赐教，谢谢！

Re: [天妒WS](#) 2016-01-25 10:52发表



回复nuohanfengyun：你的数据是人脸数据是对齐了的么，输入图像是55x47大小的RGB

Re: [落目白](#) 2016-09-19 19:55发表



回复天妒WS：你好，请问你的切图是按照论文上的还是每个人脸切割一张的呢？

您还没有登录,请[登录](#)或[注册](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

关闭

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

网站客服

杂志客服

微博客服

webmaster@csdn.net

400-660-0108

| 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 | 江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved



关闭