

CSDN新首页上线啦，邀请你来立即体验！(http://blog.csdn.net/)



博客 (//blog.csdn.net/shakevincent) 学院 (//edu.csdn.net?ref=toolbar)
下载 (//download.csdn.net?ref=toolbar) GitChat (//gitbook.cn/?ref=csdn)
更多 ▾

opencv-dnn人脸识别

原创 2016年10月27日 15:50:15

标签：opencv (http://so.csdn.net/so/search/s.do?q=opencv&t=blog) /
深度学习 (http://so.csdn.net/so/search/s.do?q=深度学习&t=blog) /
脸部识别 (http://so.csdn.net/so/search/s.do?q=脸部识别&t=blog)

3306

随着深度学习的发展，opencv3.1也可以直接调用caffe或者torch。下面是使用opencv的dnn模块来进行人脸识别：

1：编译opencv3.1

首先下载opencv源码https://github.com/opencv/opencv (https://github.com/opencv/opencv)

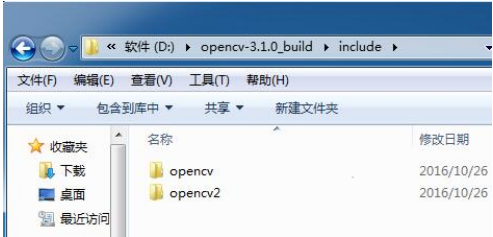
下载Cmake https://cmake.org/download/ (https://cmake.org/download/)

下载opencv的

具体的cmake过程可以参考这篇博客：

http://www.cnblogs.com/jliangqiu2016/p/5597501.html (http://www.cnblogs.com/jliangqiu2016/p/5597501.html)

编译完成后可以把不需要的文件删除仅保留include bin lib 文件即可。



低价处理二手车



+ 关注
(http://blog.csdn.net/shakevincent)

原创	粉丝	喜欢	未开通
19	40	0	(https://github.com/shakevincent)

他的最新文章
更多文章 (http://blog.csdn.net/shakevincent)

微软libcaffe封装成dll和lib !!! (http://blog.csdn.net/shakevincent/article/details/76098626)

caffe 提取LMDB时可能出现的错误 (http://blog.csdn.net/shakevincent/article/details/53905981)

windows下使用训练好的caffemodel做分类 (http://blog.csdn.net/shakevincent/article/details/52995253)

相关推荐

OpenCV dnn模块支持Caffe (http://blog.csdn.net/BBZZ2/article/details/52535840)

OpenCV3.3出炉，DNN为最大亮点 (http://blog.csdn.net/xingchenbingbuyu/article/details/76759445)

opencv3.3出炉，与深度学习结合更密切



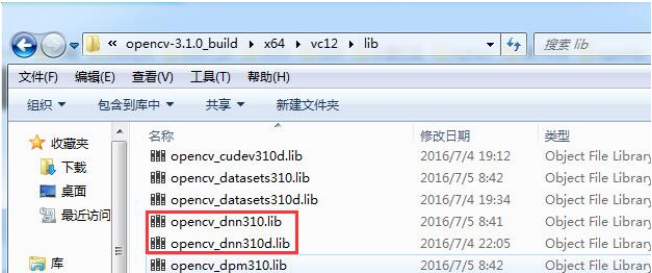
低价处理二手车



立即体验

内容举报

返回顶部



编译好的opencv3.1和普通opencv的配置过程一样：

1	opencv_aruco310.lib
2	opencv_bgsegm310.lib
3	opencv_bioinspired310.lib
4	opencv_calib3d310.lib
5	opencv_ccalib310.lib
6	opencv_core310.lib
7	opencv_cudaarithm310.lib
8	opencv_cudabgsegm310.lib
9	opencv_cudacodec310.lib
10	opencv_cudafeatures2d310.lib
11	opencv_cudafilters310.lib
12	opencv_cudaimproc310.lib
13	opencv_cudalegacy310.lib
14	opencv_cudaobjdetect310.lib
15	opencv_cudaoptflow310.lib
16	opencv_cudastereo310.lib
17	opencv_cudawarping310.lib
18	opencv_cudev310.lib
19	opencv_datasets310.lib
20	opencv_dnn310.lib
21	opencv_dpm310.lib
22	opencv_face310.lib
23	opencv_features2d310.lib
24	opencv_flann310.lib



在线课程



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-T0TAq15H)



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-T0TAq15H)



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-T0TAq15H)



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-T0TAq15H)



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-T0TAq15H)



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-T0TAq15H)



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-T0TAq15H)



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-T0TAq15H)



windows下使用训练好的caffemodel做分类 (http://blog.csdn.net/shakevincent/article/details/52995253)

opencv -dnn人脸识别 (http://blog.csdn.net/shakevincent/article/details/52946499)

```

24 opencv_houghlines310.lib
25 opencv_fuzzy310.lib
26 opencv_highgui310.lib
27 opencv_imgcodecs310.lib
28 opencv_imgproc310.lib
29 opencv_line_descriptor310.lib
30 opencv_ml310.lib
31 opencv_objdetect310.lib
32 opencv_optflow310.lib
33 opencv_photo310.lib
34 opencv_plot310.lib
35 opencv_reg310.lib
36 opencv_rgbd310.lib
37 opencv_saliency310.lib
38 opencv_shape310.lib
39 opencv_stereo310.lib
40 opencv_stitching310.lib
41 opencv_structured_light310.lib

42 opencv_superres310.lib
43 opencv_surface_matching310.lib
44 opencv_text310.lib
45 opencv_tracking310.lib
46 opencv_ts310.lib
47 opencv_video310.lib
48 opencv_videoio310.lib
49 opencv_videostab310.lib
50 opencv_viz310.lib
51 opencv_xfeatures2d310.lib
52 opencv_ximgproc310.lib
53 opencv_xobjdetect310.lib
54 opencv_xphoto310.lib

```

在opencv的源码中提供了dnn的test.cpp

下面具体分析代码：

/* Find best class for the blob (i. e. class with maximal probability) */

获取prob层的输出：实际意义为测试图片所对应与标签的概率值。resize成一个列向量，然后排序，输出最大值和最大值所对应的位置。

```

layers {
  bottom: "fc8"
  top: "prob"
  name: "prob"
  type: SOFTMAX
}

```

```

1 void getMaxClass(dnn::Blob &probBlob, int *classId, double *classProb)
2 {
3     Mat probMat = probBlob.matRefConst().reshape(1, 1); //reshape the blob to 1x1000 matrix
4     Point classNumber;
5
6     minMaxLoc(probMat, NULL, classProb, NULL, &classNumber);
7     *classId = classNumber.x;
8 }

```

内容举报

返回顶部



内容举报

相关系数函数：一种相似性度量用于判断两个人的相似性距离。

返回顶部

```
1 float mean(const std::vector<float>& v)
2 {
3     assert(v.size() != 0);
4     float ret = 0.0;
5     for (std::vector<float>::size_type i = 0; i != v.size(); ++i)
6     {
7         ret += v[i];
8     }
9     return ret / v.size();
10 }
11
12 float cov(const std::vector<float>& v1, const std::vector<float>& v2)
13 {
14     assert(v1.size() == v2.size() && v1.size() > 1);
15     float ret = 0.0;
16     float v1a = mean(v1), v2a = mean(v2);
17
18     for (std::vector<float>::size_type i = 0; i != v1.size(); ++i)
19     {
20         ret += (v1[i] - v1a) * (v2[i] - v2a);
21     }
22
23     return ret / (v1.size() - 1);
24 }
25
26 // 相关系数
27 float coefficient(const std::vector<float>& v1, const std::vector<float>& v2)
28 {
29     assert(v1.size() == v2.size());
30     return cov(v1, v2) / sqrt(cov(v1, v1) * cov(v2, v2));
31 }
```

cos相似性距离函数：

内容举报
返回顶部

```
1 //cos 相似性度量
2 float cos_distance(const std::vector<float>& vecfeature1, vector<float>& vecfeature2)
3 {
4     float cos_dis=0;
5     float dotmal=0, norm1=0, norm2=0;
6     for (int i = 0; i < vecfeature1.size(); i++)
```





1



```

6   for (int i = 0; i < vecfeature1.size(); i++)
7   {
8       dotmal += vecfeature1[i] * vecfeature2[i];
9       norm1 += vecfeature1[i] * vecfeature1[i];
10      norm2 += vecfeature2[i] * vecfeature2[i];
11  }
12  }
13  norm1 = sqrt(norm1);
14  norm2 = sqrt(norm2);
15  cos_dis = dotmal / (norm1*norm2);
16  return cos_dis;
17  }

```

下面是主函数：



内容举报



返回顶部



1



```

1  /**/
2  //
3  //
4  #include <opencv2/dnn.hpp>
5  #include <opencv2/imgproc.hpp>
6  #include <opencv2/highgui.hpp>
7  using namespace cv;
8  using namespace cv::dnn;
9
10 #include <fstream>
11 #include <iostream>
12 #include <cstdlib>
13 #include <time.h>
14 #include <math.h>
15 using namespace std;
16
17 /* Find best class for the blob (i. e. class with maximal probability) */

```



```

18 void getMaxClass(dnn::Blob &probBlob, int *classId, double *classProb)
19 {
20     Mat probMat = probBlob.matRefConst().reshape(1, 1); //reshape the blob to 1x1000 matrix
21     Point classNumber;
22
23     minMaxLoc(probMat, NULL, classProb, NULL, &classNumber);
24     *classId = classNumber.x;
25 }
26
27 std::vector<String> readClassNames(const char *filename = "synset_words.txt")
28 {
29     std::vector<String> classNames;
30
31     std::ifstream fp(filename);
32     if (!fp.is_open())
33     {
34         std::cerr << "File with classes labels not found: " << filename << std::endl;
35         exit(-1);
36     }
37
38     std::string name;
39     while (!fp.eof())
40     {
41         std::getline(fp, name);
42
43         if (name.length())
44             classNames.push_back(name.substr(name.find(' ') + 1));
45     }
46     fp.close();
47     return classNames;
48 }
49 string Int_String(int a)
50 {
51     std::stringstream ss;
52     std::string str;
53     ss << a;
54     ss >> str;
55     return str;
56 }
57 float mean(const std::vector<float> &v)
58 {
59     assert(v.size() != 0);
60     float ret = 0.0;
61     for (std::vector<float>::size_type i = 0; i != v.size(); ++i)
62     {
63         ret += v[i];
64     }
65     return ret / v.size();
66 }
67
68 float cov(const std::vector<float> &v1, const std::vector<float> &v2)
69 {
70     assert(v1.size() == v2.size() && v1.size() > 1);
71     float ret = 0.0;

```



内容举报



返回顶部



1



```

71     float ret = 0.0;
72     float v1a = mean(v1), v2a = mean(v2);
73
74     for (std::vector<float>::size_type i = 0; i != v1.size(); ++i)
75     {
76         ret += (v1[i] - v1a) * (v2[i] - v2a);
77     }
78
79     return ret / (v1.size() - 1);
80 }
81
82 // 相关系数
83 float coefficient(const std::vector<float>& v1, const std::vector<float>& v2)
84 {
85     assert(v1.size() == v2.size());
86     return cov(v1, v2) / sqrt(cov(v1, v1) * cov(v2, v2));
87 }
88 //cos 相似性度量
89 float cos_distance(const std::vector<float>& vecfeature1, vector<float>& vecfeature2)
90 {
91     float cos_dis=0;
92     float dotmal=0, norm1=0, norm2=0;
93     for (int i = 0; i < vecfeature1.size(); i++)
94     {
95         dotmal += vecfeature1[i] * vecfeature2[i];
96         norm1 += vecfeature1[i] * vecfeature1[i];
97         norm2 += vecfeature2[i] * vecfeature2[i];
98     }
99
100     norm1 = sqrt(norm1);
101     norm2 = sqrt(norm2);
102     cos_dis = dotmal / (norm1*norm2);
103     return cos_dis;
104 }
105 int main()
106 {
107     String modelTxt = "VGG_FACE_deploy.prototxt";
108     String modelBin = "VGG_FACE.caffemodel";
109     //String imageFile = (argc > 1) ? argv[1] : "ak.png";
110
111     /*String modelTxt = "bvlc_googlenet.prototxt";
112     String modelBin = "bvlc_googlenet.caffemodel";
113     String imageFile = (argc > 1) ? argv[1] : "1.jpg";*/
114
115     //! [Create the importer of Caffe model]
116     Ptr<dnn::Importer> importer;
117     try
118     {
119         importer = dnn::createCaffeImporter(modelTxt, modelBin);
120     }
121     catch (const cv::Exception &err)    //Importer can throw errors, we will catch them
122     {
123         std::cerr << err.msg << std::endl;

```



内容举报

返回顶部

内容举报

返回顶部



1



```

124 }
125 //! [Create the importer of Caffe model]
126
127 if (!importer)
128 {
129     std::cerr << "Can't load network by using the following files: " << std::endl;
130     std::cerr << "prototxt: " << modelTxt << std::endl;
131     std::cerr << "caffemodel: " << modelBin << std::endl;
132     std::cerr << "bvlc_googlenet.caffemodel can be downloaded here:" << std::endl;
133     std::cerr << "http://dl.caffe.berkeleyvision.org/bvlc_googlenet.caffemodel" << std::endl;
134     exit(-1);
135 }
136
137 //! [Initialize network]
138 dnn::Net net;
139 importer->populateNet(net);
140 importer.release(); //We don't need importer anymore
141 //! [Initialize network]
142
143 //! [Prepare blob]
144 //=====进行训练样本提取=====可修改=====
145 //=====五个人，每人一张照片=====
146 std::vector<Mat> train;
147 std::vector<int> train_label;
148 int train_man = 1, train_num = 1; //训练的人的种类、人的个数
149
150 for (train_man = 1; train_man <= 4; train_man++)
151 {
152     for (train_num = 1; train_num <= 1; train_num++)
153     {
154         string train_road = "VGG_train/" + Int_String(train_man) + "-" + Int_String(train_num) + ".jpg";
155         cv::Mat train_Sample = imread(train_road);
156         // cv::imshow("train_1", train_Sample);
157         // waitKey(1);
158         if (!train_Sample.empty())
159         {
160             train.push_back(train_Sample);
161             train_label.push_back(train_man);
162             cout << "There is train pic!!" << train_man << "" << train_num << endl;
163         }
164
165         else
166         {
167             cout << "There is no pic!!" << train_man << "" << train_num;
168             getchar();
169             exit(-1);
170         }
171     }
172 }
173
174 clock_t start, finish;
175 double totaltime;
176
177 //=====训练=====

```



1



内容举报

返回顶部




```
1/b
177 dnn::Blob train_blob = dnn::Blob(train);
178 net.setBlob("data", train_blob);
179 cout << "Please wait..." << endl;
180 net.forward();
181 dnn::Blob prob = net.getBlob("fc7");//提取哪一层
182 Mat probMat = prob.matRefConst().reshape(1, 1); //reshape the blob to 1x4096 matrix
183
184 finish = clock();
185 totaltime = (double)(finish - start) / CLOCKS_PER_SEC;
186 totaltime = totaltime / 4;
187 std::cout << "extract feature the train image is : " << totaltime << "sec" << std::endl;
188
189 vector< vector<float> > > feature_vector;
190 feature_vector.clear();
191 int train_man_num = 0;//第几个人
192 clock_t start2, finish2;
193 double totaltime2;
194 start2 = clock();
195 for (train_man_num = 0; train_man_num <= 3; train_man_num++)
196 {
197     vector<float> feature_one;//单个人的feature
198     int channel = 0;
199     while (channel < 4096)//看网络相应层的output
200     {
201         feature_one.push_back(*prob.ptrf(train_man_num, channel, 1, 1));
202         channel++;
203         string train_txt = Int_String(train_man_num) + ".txt";
204         ofstream myfile(train_txt, ios::app); //example.txt是你输出的文件的名字,这里把向量都分开保存为txt, 以便于后
205         myfile << *prob.ptrf(train_man_num, channel, 1, 1) << endl;
206     }
207     feature_vector.push_back(feature_one);//把它赋给二维数组
208     feature_one.clear();
209 }
210 finish2 = clock();
211 totaltime2 = (double)(finish2 - start2) / CLOCKS_PER_SEC;
212 totaltime2 = totaltime2 / 4;
213 std::cout << "save the train image feature is : " << totaltime2 << "sec" << std::endl;
214 cout << "Successful extract!!!" << endl;
215 train_blob.offset();
216 //=====//
217 //                                     //
218 //             Test                     //
219 //                                     //
220 //=====//
221 //string test_fileroad = "C:/wamp/www/pic/" + Int_String(x) + ".jpg";//图片的地方, 改成摄像头也可以。
222 Mat testSample = imread("C:\\Users\\naslab\\Desktop\\opencv_dnn_face_train\\opencv_dnn\\WGG_test\\1.jpg");
223
224 if (testSample.empty())
225     cout << "There is no testSample ..." << endl;
226 else
227 {
228     //testSample = Facedetect(testSample);
229     ...
230 }
```



内容举报

返回顶部

```
229 vector<Mat> test;
230 vector<int> test_label;
231 test.push_back(testSample);
232 test_label.push_back(0);
233 //then
234 dnn::Blob test_blob = dnn::Blob(test);//如果用原来的似乎会报错。。。
235 net.setBlob("data", test_blob);
236 cout << "extracting features..." << endl;
237
238 clock_t start1, finish1;
239 double totaltime1;
240 start1 = clock();
241 net.forward();
242
243 dnn::Blob prob_test = net.getBlob("fc7");
244 vector<float> test_feature;//第8层的特征
245
246
247
248 int channel = 0;
249 while (channel < 4096)
250 {
251     test_feature.push_back(*prob_test.ptrf(0, channel, 1, 1));
252     channel++;
253 }
254 finish1 = clock();
255 totaltime1 = (double)(finish1 - start1) / CLOCKS_PER_SEC;
256 std::cout << "extract feature the train image is :" << totaltime1 << "sec" << std::endl;
257
258 cout << "we got it." << endl;
259 float higher_score = 0;//相似度
260 int T_number = 0;
261 for (int test_num_vector = 0; test_num_vector <= 3; test_num_vector++)
262 {
263     float score1 = coefficient(feature_vector[test_num_vector], test_feature);
264     float score = cos_distance(feature_vector[test_num_vector], test_feature);
265     cout << "The coefficient" << test_num_vector << "-----to-----" << score1 << endl;
266     cout << "The cos_distance" << test_num_vector << "-----to-----" << score << endl;
267     if (score > higher_score)
268     {
269         higher_score = score;
270         T_number = test_num_vector;
271     }
272 }
273
274 cv::imshow("trainSample", train[T_number]);//可以直接把和测试样本最相近的一张图亮出来
275 cv::waitKey(1);
276 }
277 cv::imshow("testSample", testSample);
278 cv::waitKey(0);
279 } //main
```



内容举报

返回顶部

内容举报

返回顶部

里面我有所修改，本来提取的是fc8层的，后来改成fc7层4096维特征。

```
Net Outputs(1):
prob
extract feature the train image is :8.60375sec
save the train image feature is :15.4968sec
Successful extract!!!
extracting features...
extract feature the train image is :33.596sec
we got it..
The coefficient0-----to-----1
The cos_distance0-----to-----1
The coefficient1-----to-----0.492546
The cos_distance1-----to-----0.518373
The coefficient2-----to-----0.187122
The cos_distance2-----to-----0.221112
The coefficient3-----to-----0.316476
The cos_distance3-----to-----0.345675
```



这速度真喜人！！！！！！提取个特征就要8秒！！！！！！

1：程序的改进方向：

1：保存提取的特征为dat文件，这样可以预先训练，直接测试即可

2：程序输出的是Bolb格式的数据，保存数据占用的时间比较长，可以修改一下。

3：还是使用caffe for windows吧！

下面是一些参考链接：

http://blog.csdn.net/mr_curry/article/details/52183263 (http://blog.csdn.net/mr_curry/article/details/52183263)

http://docs.opencv.org/trunk/d5/de7/tutorial_dnn_googlenet.html

(http://docs.opencv.org/trunk/d5/de7/tutorial_dnn_googlenet.html)

http://docs.opencv.org/trunk/de/d25/tutorial_dnn_build.html

(http://docs.opencv.org/trunk/de/d25/tutorial_dnn_build.html)


版权声明：本文为博主原创文章，未经博主允许不得转载。

内容举报

返回顶部

 发表你的评论

(http://my.csdn.net/weixin_35068028)

 lqhu23 (/lqhu23) 2017-11-25 22:01
(/lqhu23)

2楼

你好，博主，不知道你用的是opencv是哪个版本的？为什么我的dnn类中没有Blob变量。

回复 1条回复





1



u014511048 (/u014511048) 2016-11-04 10:45

1楼

(/u014511048)G_FACE_deploy.prototxt这个文件有修改吗，我在vgg_face官网下了，程序运行报错。

回复

查看 7 条热评 ∨

相关文章推荐

OpenCV dnn模块支持Caffe (<http://blog.csdn.net/BBZZ2/article/details/52535840>)

```
#include #include #include using namespace cv; using namespace cv::dnn; #include #include #incl...
```

BBZZ2 (<http://blog.csdn.net/BBZZ2>) 2016年09月14日 10:51 1142

OpenCV3.3出炉，DNN为最大亮点 (<http://blog.csdn.net/xingchenbingbuyu/article/details/...>)

OpenCV3.3在8月3号正式出炉，想要体验最新特性的朋友可以去官网下载了，反正配置一下只需要几分钟。这次最主要的更新就是，终于把DNN模块从contrib里面提到主仓库里面，放到了官方发布版中。虽...

xingchenbingbuyu (<http://blog.csdn.net/xingchenbingbuyu>)
2017年08月06日 11:08
 2758



程序员该学Python吗？90%的码农这么说！

日前我们被一条消息刷屏：我国的程序员更爱Python。一石激起千层浪，身边的程序员对待 Python 究竟是何种态度？他们是怎么说...

http://www.baidu.com/cb.php?c=lgF_pyfqhHmkjnvPj0cI0Z0qnFK9UyZp1f4PjDs0Aw-Hc3rHnYnHb0TAq15HfLPWRznjb0T1dBP1Tzuhm1PH-WrAn4mvD30AwY5HDdnHc3rj03rjb0lgF_5y9YIz0IQzq-ZR8mLPbUB48ugfElAqspynETZ-YpAq8nWQdlAdxTvqdThP-yF_UvTkn0KzujYk0AFV5H00T2cqn0KdpfqhHRLPjnvnfKEpyfqhHc4rj6kP0KWpyfqP1civrHnz0AqLUWYs0ZK45HcsP6KWThnqn1TzP1b)

opencv3.3出炉，与深度学习结合更密切 (<http://blog.csdn.net/HelloHaibo/article/details/777...>)

opencv3.3在八月3号发布，日志中说到：The main news is that we promoted DNN module from opencv_contrib to the main line.

 HelloHaibo (<http://blog.csdn.net/HelloHaibo>) 2017年08月30日 13:16 1295

【OpenCV3.3】编译源码并搭建VS2017+Windows开发环境 (<http://blog.csdn.net/rrrfff/article...>)

OpenCV 3.3在8月3号正式Release，带来了许多优化改进和新特性，包括备受关注的深度神经网络(DNN)模块被正式引入主仓库，标志着OpenCV对DNN有了更深层次的优化与支持；支持通过宏...

rrrfff (<http://blog.csdn.net/rrrfff>) 2017年08月07日 16:10 18451

opencv的dnn解析 (<http://blog.csdn.net/langb2014/article/details/51286828>)

在学习过caffeemodel加载之后，回头看看这个dnn里面都编译了哪些函数？先看blob头文件：`#ifndef __OPENCV_DNN_DNN_BLOB_HPP__` `#define __OPE...`



低价处理二手车





1



低价处理二手车



 langb2014 (<http://blog.csdn.net/langb2014>) 2016年05月06日 22:09  5193





一学就会的 WordPress 实战课

学习完本课程可以掌握基本的 WordPress 的开发能力，后续可以根据需要开发适合自己的主题、插件，打造最个性的 WordPress 站点。

(http://www.baidu.com/cb.php?c=lgF_pyfqnhmknjfvP1m0lZ0qnfK9ujYzP1f4Pjnz0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1YvuHmvP1u-rjIWuW-bn1bY0AwY5HDdnHc3rj03rjb0lgF_5y9YlZ0IQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq_lyd8Q1N9nHmvnj7hnHPWnjFhPAD1Pyn4uW99ujqdlAdxTvqdThP-5HDknWw9mhhEusKzujYk0AFV5H00TZcqn0KdpyfqnhRLPjnvnfKEpyfqnhnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqn1mLnWm)



【C++】基于 OpenCV 的人脸识别(强烈推荐) (<http://blog.csdn.net/u013346007/article/detai...>)

一点背景知识 OpenCV 是一个开源的计算机视觉和机器学习库。它包含成千上万优化过的算法，为各种计算机视觉应用提供了一个通用工具包。根据这个项目的关于页面，OpenCV 已被广泛运用在各种项目上，从...

 u013346007 (<http://blog.csdn.net/u013346007>) 2016年01月23日 14:18  39447



基于OpenCV读取摄像头进行人脸检测和人脸识别 (<http://blog.csdn.net/chary8088/article/de...>)

前段时间使用OpenCV的库函数实现了人脸检测和人脸识别，笔者的实验环境为VS2010+OpenCV2.4.4，OpenCV的环境配置网上有很多，不再赘述。检测的代码网上很多，记不清楚从哪儿copy的...

 chary8088 (<http://blog.csdn.net/chary8088>) 2014年05月03日 21:39  49302



使用Opencv的dnn模块进行深度学习人脸识别（速度较慢） (http://blog.csdn.net/Mr_Curry/a...)

这里给出了一个样例，如何使用Opencv的dnn模块进行人脸识别，因为在编译Opencv时似乎没有加上WITH_CUDA，所以导致forward()的传播速度巨慢，但是不影响我们进行实验。准备工作1、...

 Mr_Curry (http://blog.csdn.net/Mr_Curry) 2016年08月11日 16:03  9713

OpenCV实践之路——人脸识别之三识别自己的脸 (<http://blog.csdn.net/xingchenbingbuyu/a...>)

在之前OpenCV实践之路——人脸识别之一数据收集和预处理和OpenCV实践之路——人脸识别之二模型训练两篇博客中，已经把人脸识别的整个流程全部交代清楚了。包括今天这篇人脸识别方面的内容都已经在上述...

 xingchenbingbuyu (<http://blog.csdn.net/xingchenbingbuyu>) 2016年05月21日 21:18  29078

opencv 人脸识别 （一）训练样本的处理 (<http://blog.csdn.net/abcjennifer/article/details/20...>)

本文实现基于eigenface的人脸检测与识别。给定一个图像数据库，进行以下步骤：进行人脸检测，将检测出的人脸存入数据库2 对数据库2进行人脸建模 在测试集上进行recognition 本篇实现第...

 abcjennifer (<http://blog.csdn.net/abcjennifer>) 2014年03月04日 13:15  66427

基于OpenCV读取摄像头进行人脸检测和人脸识别 (<http://blog.csdn.net/u010944555/article/d...>)

前段时间使用OpenCV的库函数实现了人脸检测和人脸识别，检测的代码网上很多，记不清楚从哪儿copy的了，识别的代码是从OpenCV官网上找到的，<http://docs.opencv.org/trunk>

 内容举报

 返回顶部

 1   





Delphi7高级应用开发随书源码 (http://download.csdn.net/download/chenx...)

http://download.csdn.net/download/chenx...

2003年04月30日 00:00

676KB

下载

OpenCV实践之路——人脸识别之一数据收集和预处理 (http://blog.csdn.net/xingchenbingbu...)

前段时间对人脸检测和人脸标记进行了一些尝试：人脸检测（C++/Python）和用Dlib库进行人脸检测和人脸标记。但是检测和识别是不同的，检测解决的问题是图片中有没有人脸；而识别解决的问题是，如果一张...

 xingchenbingbuyu (http://blog.csdn.net/xingchenbingbuyu)

2016年05月12日 21:27

19575

使用OpenCV进行人脸识别的三种方法 (http://blog.csdn.net/Real_Myth/article/details/52771...)

OpenCV从版本2.4开始，加入了一个类FaceRecognizer，使用它可以方便地进行人脸识别(源代码，在OpenCV的opencv\modules\contrib\doc\face_recognition\...

 Real_Myth (http://blog.csdn.net/Real_Myth)

2016年10月09日 22:18

7892

OpenCV — 人脸识别 (http://blog.csdn.net/jscese/article/details/54409627)

前段时间弄过一下人脸识别相关的东西，记录一下撰写不易,转载需注明出处：http://blog.csdn.net/jscese/article/details/54409627本文来自【jscese】...

 jscese (http://blog.csdn.net/jscese)

2017年01月13日 16:22

4387

OpenCV中使用Eigenfaces 或 Fisherfaces进行人脸识别 (http://blog.csdn.net/dawn_chen1...)

OpenCV中使用Eigenfaces 或 Fisherfaces进行人脸识别 Translate by Dawn ~~~~~~翻译的好辛苦~~~~~...

 dawn_chen121 (http://blog.csdn.net/dawn_chen121)

2016年08月16日 08:21

3722

Delphi7高级应用开发随书源码 (http://download.csdn.net/download/chenx...)

http://download.csdn.net/download/chenx...

2003年04月30日 00:00

676KB

下载

【OpenCV人脸识别入门教程之二】人脸检测 (http://blog.csdn.net/lsq2902101015/article/de...)

本篇文章主要介绍了如何使用OpenCV实现人脸检测的功能。要实现人脸识别功能，首先要进行人脸检测，判断出图片中人脸的位置，才能进行下一步的操作。人脸检测的方法介绍 OpenCV中的方法 函数 ...

 lsq2902101015 (http://blog.csdn.net/lsq2902101015)

2015年09月28日 18:27

10691

opencv 人脸识别（二）训练和识别 (http://blog.csdn.net/abcjennifer/article/details/204460...)

上一篇中我们对训练数据做了一些预处理，检测出人脸并保存在pic\color\文件夹下（x=1,2,3,...类别号），本文做训练和识别。为了识别，首先将人脸训练数据 转为灰度、对齐、归一化，再放入...

1

1

1

1

1

内容举报

返回顶部

内容举报

返回顶部

http://blog.csdn.net/shakevincent/article/details/52946499

14/15

abcjennifer (http://blog.csdn.net/abcjennifer) 2014年03月04日 13:21 35665



Delphi7高级应用开发随书源码 (http://download.csdn.net/download/chenx...

2003年04月30日 00:00 676KB 下载



低价处理二手车



1



内容举报

返回顶部