

FaceProcessing.cpp :

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/article/details/52183263)
👤 9674

```
1  #include"FaceProcessing.h"
2
3  int gauss(float x[], float y[], int length, float sigma);
4  Mat gaussianfilter(Mat img, double sigma0, double sigma1, double shift1, double shift2);
5  Mat FaceProcessing(const Mat &img_, double gamma = 0.2, double sigma0 = 1, double sigma1 = -2, double mask = 0, dou
6
7  //找出矩阵中的最大值或最小值，输入MAX，或MIN
8  double MatMaxMin(Mat im, String flag = "MAX") .
9  {
10     double value = im.ptr<float>(0)[0];
11     if (flag == "MAX")
12     {
13         for (int i = 0; i<im.rows; i++)
14             for (int j = 0; j<im.cols; j++)
15                 if (im.ptr<float>(i)[j]>value)
16                     value = im.ptr<float>(i)[j];
17         return value;
18     }
19     else if (flag == "MIN")
20     {
21         for (int i = 0; i<im.rows; i++)
22             for (int j = 0; j<im.cols; j++)
23                 if (im.ptr<float>(i)[j]<value)
24                     value = im.ptr<float>(i)[j];
25         return value;
26     }
27     return -1;
28 }
29 //高斯滤波
30 Mat gaussianfilter(Mat img, double sigma0, double sigma1, double shift1 = 0, double shift2 = 0)
31 {
32     int i, j;
33     sigma0 = (float)sigma0;
34     sigma1 = (float)sigma1;
35     shift1 = (float)shift1;
```

```
35 snr1 = (float)snr1;
36 shift2 = (float)shift2;
37 Mat img2 = img;
38 Mat img3 = img;
39 Mat imgResult;
40
41 //将数据存入横向高斯模板中
42
43 int rowLength = (int)(floor(3.0*sigma0 + 0.5 - shift1) - ceil(-3.0*sigma0 - 0.5 - shift1) + 1);
44 int rowBegin = (int)ceil(-3.0*sigma0 - 0.5 - shift1);
45 float rowArray[30], Gx[30];
46 for (i = 0; i < rowLength; i++)
47 {
48     rowArray[i] = rowBegin + i;
49 }
50 gauss(rowArray, Gx, rowLength, sigma0);
51 Mat kx = Mat(1, rowLength, CV_32F); //转换成mat类型
52 float *pData1 = kx.ptr<float>(0);
53 for (i = 0; i < rowLength; i++)
54 {
55     pData1[i] = Gx[i];
56 }
57 //将数据存入纵向高斯模板中
58 int colLength = (int)(floor(3.0*sigma1 + 0.5 - shift2) - ceil(-3.0*sigma1 - 0.5 - shift2) + 1);
59 int colBegin = (int)ceil(-3.0*sigma1 - 0.5 - shift2);
60 float colArray[30], Gy[30];
61 for (i = 0; i < colLength; i++)
62 {
63     colArray[i] = colBegin + i;
64 }
65 gauss(colArray, Gy, colLength, sigma1);
66 Mat ky = Mat(colLength, 1, CV_32F);
67 float *pData2;
68 for (i = 0; i < colLength; i++)
69 {
70     pData2 = ky.ptr<float>(i);
71     pData2[0] = Gy[i];
72 }
73 filter2D(img, img2, img.depth(), kx, Point(-1, -1));
74 filter2D(img2, imgResult, img2.depth(), ky, Point(-1, -1));
75 return imgResult;
76 }
77 //行列卷积
78 int gauss(float x[], float y[], int length, float sigma)
79 {
80     int i;
81     float sum = 0.0;
82     for (i = 0; i < length; i++)
83     {
84         x[i] = exp(-pow(x[i], 2) / (2 * pow(sigma, 2)));
85         sum += x[i];
86     }
87 }
```



6



6



```
87     for (i = 0; i<length; i++)
88     {
89         y[i] = x[i] / sum;
90     }
91     return 1;
92 }
93
94
95 Mat FaceProcessing(const Mat &img_, double gamma, double sigma0, double sigma1, double mask, double do_norm)
96 {
97     Mat img;
98     img_.convertTo(img, CV_32F);
99     Mat imT1, imT2;
100     int rows = img.rows;
101     int cols = img.cols;
102     Mat im = img;
103     int b = floor(3 * abs(sigma1)); //左右扩充边缘的距离
104     Mat imtemp(Size(cols + 2 * b, rows + 2 * b), CV_32F, Scalar(0)); //保存扩充的图形
105     Mat imtemp2(Size(cols, rows), CV_32F, Scalar(0));
106     float s = 0.0;
107     //Gamma correct input image to increase local contrast in shadowed regions.
108     if (gamma == 0)
109     {
110         double impixeltemp = 0;
111         double Max = MatMaxMin(im, "MAX"); //等价于max(1,max(max(im)))
112         for (int i = 0; i<rows; i++)
113             for (int j = 0; j<cols; j++)
114             {
115                 impixeltemp = log(im.ptr<float>()(i)[j] + Max / 256);
116                 im.ptr<float>()(i)[j] = impixeltemp;
117             }
118     }
119     else
120     {
121         for (int i = 0; i<rows; i++)
122             for (int j = 0; j<cols; j++)
123                 im.ptr<float>()(i)[j] = pow(im.ptr<float>()(i)[j], gamma);
124     }
125     float *pData1;
126     //run prefilter, if any
127     if (sigma1)
128     {
129         double border = 1;
130         if (border) //add extend-as-constant image border to reduce
131             //boundary effects
132         {
133             for (int i = 0; i<rows + 2 * b - 1; i++)
134             {
135                 pData1 = imtemp.ptr<float>(i);
136                 for (int j = 0; j<cols + 2 * b - 1; j++){
137                     //中间
138                     if (i >= b&&i<im.rows + b&&j >= b&&j<im.cols + b)
139                         pData1[j] = im.ptr<float>(i - b)[j - b];
```



6



```
140 //左上
141 else if (i<b&&j<b)
142     pData1[j] = im.ptr<float>(0)[0];
143 //右上
144 else if (i<b&&j >= im.cols + b&&j<cols + 2 * b)
145     pData1[j] = im.ptr<float>(0)[cols - 1];
146 //左下
147 else if (i >= im.rows + b&&i<rows + 2 * b&&j<b)
148     pData1[j] = im.ptr<float>(rows - 1)[0];
149 //右下
150 else if (i >= im.rows + b&&j >= im.cols + b)
151     pData1[j] = im.ptr<float>(im.rows - 1)[im.cols - 1];
152 //上方
153 else if (i<b&&j >= b&&j<im.cols + b)
154     pData1[j] = im.ptr<float>(0)[j - b];
155 //下方
156 else if (i >= im.rows + b&&j >= b&&j<im.cols + b)
157     pData1[j] = im.ptr<float>(im.rows - 1)[j - b];
158 //左方
159 else if (j<b&&i >= b&&i<im.rows + b)
160     pData1[j] = im.ptr<float>(i - b)[0];
161 //右方
162 else if (j >= im.cols + b&&i >= b&&i<im.rows + b)
163     pData1[j] = im.ptr<float>(i - b)[im.cols - 1];/**/
164 }
165 }
166 }
167
168 else
169 {
170     if (sigma0>0)
171     {
172         imT1 = gaussianfilter(imtemp, sigma0, sigma0);
173         imT2 = gaussianfilter(imtemp, -sigma1, -sigma1);
174         imtemp = imT1 - imT2;
175         //imtemp=gaussianfilter(imtemp,sigma0,sigma0)-gaussianfilter(imtemp,-sigma1,-sigma1);
176     }
177     else
178         imtemp = imtemp - gaussianfilter(imtemp, -sigma1, -sigma1);
179 }
180
181 if (border)
182 {
183     //再取回中间部分
184     for (int i = 0; i<rows; i++)
185     {
186         pData1 = im.ptr<float>(i);
187         for (int j = 0; j<cols; j++)
188             pData1[j] = imtemp.ptr<float>(i + b)[j + b];
189     }
190 }
191 // test=im.ptr<float>(19)[19];
192 }
```

```
192     }
193
194     /*
195     % Global contrast normalization. Normalizes the spread of output
196     % values. The mean is near 0 so we don't bother to subtract
197     % it. We use a trimmed robust scatter measure for resistance to
198     % outliers such as specularities and image borders that have
199     % different values from the main image. Usually trim is about
200     % 10.
201     */
202
203     if (do_norm)
204     {
205         double a = 0.1;
206
207         double trim = abs(do_norm);
208
209         //im = im./mean(mean(abs(im).^a)/(1/a);
210         imtemp2 = abs(im);
211
212         //cvPow(&im,&im,a)//为每个元素求pow
213         for (int i = 0; i<rows; i++)
214         {
215             pData1 = imtemp2.ptr<float>(i);//imtemp2为零矩阵
216             for (int j = 0; j<cols; j++)
217                 pData1[j] = pow(imtemp2.ptr<float>(i)[j], a);
218         }
219
220         //求平均值s
221         s = 0.0;
222         for (int i = 0; i<rows; i++)
223         {
224             pData1 = imtemp2.ptr<float>(i);
225             for (int j = 0; j<cols; j++)
226                 s += imtemp2.ptr<float>(i)[j];
227         }
228         s /= (im.rows*im.cols);
229         double temp = pow(s, 1 / a);
230         for (int i = 0; i<rows; i++)
231         {
232             pData1 = im.ptr<float>(i);
233             for (int j = 0; j<cols; j++)
234                 pData1[j] = pData1[j] / temp;//点除
235         }
236
237         //im = im./mean(mean(min(trim,abs(im)).^a)/(1/a);
238         imtemp2 = abs(im);
239         for (int i = 0; i<rows; i++)
240         {
241             pData1 = imtemp2.ptr<float>(i);
242             for (int j = 0; j<cols; j++)
243                 if (pData1[j]>trim)
244                     pData1[j] = trim;//min(trim,abs(im))
```



6



```
245 //cvPow(&im,&im,a);///为每个元素求pow
246 for (int i = 0; i<rows; i++)
247 {
248     pData1 = imtemp2.ptr<float>(i);
249     for (int j = 0; j<cols; j++)
250         pData1[j] = pow(pData1[j], a);
251 }
252 //求平均值
253 s = 0.0;
254 for (int i = 0; i<rows; i++)
255 {
256     pData1 = imtemp2.ptr<float>(i);
257     for (int j = 0; j<cols; j++)
258         s += pData1[j];
259 }
260 s /= (im.rows*im.cols);
261 temp = pow(s, 1 / a);//
262 for (int i = 0; i<rows; i++)
263 {
264     pData1 = im.ptr<float>(i);
265     for (int j = 0; j<cols; j++)
266         pData1[j] = pData1[j] / temp;//点除
267 }
268
269 if (do_norm>0)
270 {
271     //im = trim*tanh(im/trim);
272     for (int i = 0; i<rows; i++)
273     {
274         pData1 = im.ptr<float>(i);
275         for (int j = 0; j<cols; j++)
276             pData1[j] = trim*tanh(pData1[j] / trim);
277     }
278 }
279 }
280 //归一化处理
281 double Min;
282 Min = MatMaxMin(im, "MIN");//找到矩阵的最小值
283 for (int i = 0; i<rows; i++)
284 {
285     pData1 = im.ptr<float>(i);
286     for (int j = 0; j<cols; j++)
287         pData1[j] += Min;
288 }
289 //im.convertTo(im, CV_32F, 1.0/255.0);
290
291 normalize(im, im, 0, 255, NORM_MINMAX);
292 //normalize(im,im,0,255,NORM_MINMAX);
293 /* for(int i=0;i<rows;i++)
294 {
295     pData1=im.ptr<float>(i);
296     for(int i=0;i<cols;i++)
```


6

≡

🔖

💬

🔗

```
297     pData1[j]*=255;
298 }*/
299 im.convertTo(im, CV_8UC1);
300 return im;
301 }
```

FaceDetect.cpp :

```
1  #include <FaceDetect.h>
2  #include <FaceRotate.h>
3  #include <FaceProcessing.h>
4  void Dlib_Prodefine()
5  {
6      deserialize("shape_predictor_68_face_landmarks.dat") >> sp;//读入标记点文件
7  }
8  Mat Facedetect(Mat frame)//脸是否存在
9  {
10     Mat gray,error;
11     cvtColor(frame, gray, CV_BGR2GRAY);
12     int * pResults = NULL;
13     pResults = facedetect_frontal_tmp((unsigned char*)(gray.ptr(0)), gray.cols, gray.rows, gray.step, 1.2f, 5, 24);
14     int peopleNUM = (pResults ? *pResults : 0);
15
16     for (int i = 0; i < peopleNUM; i++)//代表有几张人脸(pResults ? *pResults : 0)
17     {
18         short * p = ((short*)(pResults + 1)) + 6 * i;
19         Rect opencvRect(p[0], p[1], p[2], p[3]);
20         //gray = gray(opencvRect);
```

👍

6

≡

🔖

💬

🔗

```
21     dlib::rectangle dlibRect((long)opencvRect.tl().x, (long)opencvRect.tl().y, (long)opencvRect.br().x - 1, (long)opencvRect.b  
22     dlib::full_object_detection shape = sp(dlib::cv_image<uchar>(gray), dlibRect);//标记点  
23     std::vector<full_object_detection> shapes;  
24     shapes.push_back(shape);//把点保存在了shape中  
25     dlib::array<array2d<rgb_pixel>> face_chips;  
26     extract_image_chips(dlib::cv_image<uchar>(gray), get_face_chip_details(shapes), face_chips);  
27     Mat pic = toMat(face_chips[0]);  
28     cvtColor(pic, pic, CV_BGR2GRAY);  
29     resize(pic, pic, Size(224, 224));  
30     return FaceProcessing(pic);  
31 }  
32 return error;//如果没有检测出人脸 将返回一个空矩阵  
33 }
```

在上述代码中，关于dlib的array2d< rgb_pixel >类型与Mat类型的转换可以在这里进行体现：

```
1  dlib::array<array2d<rgb_pixel>> face_chips;  
2      extract_image_chips(dlib::cv_image<uchar>(gray), get_face_chip_details(shapes), face_chips);  
3      Mat pic = toMat(face_chips[0]);
```

其中face_chips[0]即为一个array2d< rgb_pixel >的类型，可以通过toMat函数进行转换。

将Mat类型转换为array2d< rgb_pixel >则可以用：

```
1  Mat gray;  
2  dlib::cv_image<uchar>(gray);
```

在这个地方，我们特别需要注意，还要转换一次灰度：

```
1  cvtColor(pic, pic, CV_BGR2GRAY);  
2      resize(pic, pic, Size(224, 224));  
3      return FaceProcessing(pic);
```

为什么？因为在测试过程中我发现，dlib函数中的toMat函数返回的不是CV_BGR2GRAY（OpenCV中的灰度图像类型），如果这里你不加，那么这个预处理将会只卷积左半部分脸。

我们可以看看识别的效果。调用FaceDetect()函数接口：

```
1  Dlib_Prodefine();  
2      //Caffe_Prodefine();  
3      Mat lena = imread("lena.jpg");  
4      imshow("Face Detect", Facedetect(lena));  
5      waitKey(0);
```

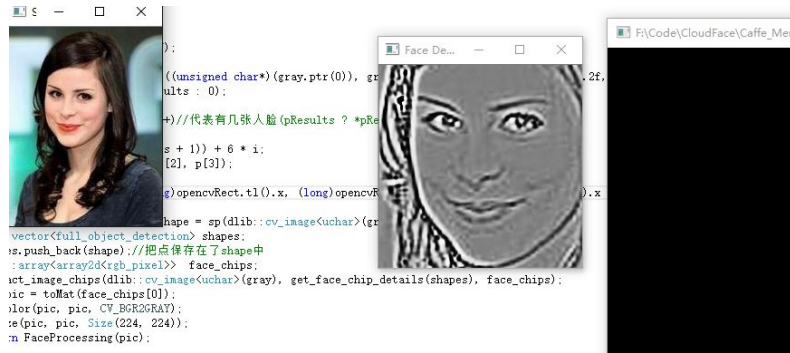


6



当然，这个地方，在进行检测之前，我们最好还是先判断FaceDetect(lena)是否为空，再进行Imshow.

检测与处理图片效果显示：



为什么要把图片最后转换为224*224的尺寸？因为：Vgg网络模型接收的就是224*224的尺寸，后面还会讲这个东西。

基于深度学习的人脸识别系统系列（Caffe+OpenCV+Dlib）
——【二】人脸检测与预处理接口的设计 完结，如果在代码过程中出现了任何问题，直接在博客下留言即可，共同交流学习。

版权声明：本文为博主原创文章，未经博主允许不得转载。



发表你的评论

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




yinsu123789 (yinsu123789) 2017-12-01 17:37

16楼

(/yinsu20170925/faceProcessing.cpp中第4,5行注释掉,否则会报错;在我这运行using namespace cv容易和winnt冲突,这个倒是好解决,不过怎么结果只是人脸变黑白了,并没有楼主的效果,估计是某段代码没执行。caffe还没配置,就先弄这个检测,应该没问题吧

回复

 dawudayudaxue (/dawudayudaxue) 2017-09-25 11:01 15楼
(/dawudayudaxue)博主,使用extract_image_chips函数提取的chips并不是原图,有没有可以修改哪里使这个函数提取的chips为原图的?

回复

 VV4yne (/VV4yne) 2017-06-06 13:00 14楼
(/VV4yne)博主,使用extract_image_chips函数提取的chips并不是原图,有没有可以修改哪里使这个函数提取的chips为原图的?


回复

查看 24 条热评

相关文章推荐


Opencv与dlib联合进行人脸关键点检测与识别 (http://blog.csdn.net/Mr_Curry/article/details...

前言依赖库: opencv 2.4.9 /dlib 19.0/libfacedetection 本篇不记录如何配置,重点在算法实现上。使用libfacedetection实现人脸区域检测,联合dlib...

 Mr_Curry (http://blog.csdn.net/Mr_Curry) 2016年07月22日 14:28 16444

使用Dlib库进行人脸检测,人脸对齐和人脸识别 (http://blog.csdn.net/u013078356/article/det...

简介在之前的博客中,我已经介绍了如何使用dlib-18.17进行人脸检测和人脸对齐。 Windows10+VS2013环境下Dlib库的编译与使用-邬小阳 使用Dlib库进行人脸检测与对齐-邬小阳...

 u013078356 (http://blog.csdn.net/u013078356) 2017年04月20日 09:12 4899




月薪40k的前端程序员都避开了哪些坑？

程序员薪水有高有低,同样工作5年的程序员,有的人月薪30K、50K,有的人可能只有5K、8K。是什么因素导致了这种差异？

(http://www.baidu.com/cb.php?c=lgF_pyfqhHmknj0dP1f0lZ0qnfK9ujYzP1nYPH0k0Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1dWuAc1nvN9njczmHI9rjD40AwY5HDdnHc3rjb4P1c0lgF_5y9YIZ0IQzq-uZR8mLPbUB48ugfElAqspynElvNBnHqdlAdxTvqdThP-5yF_UvTkn0KzujYk0AFV5H00TZcq0KdpyfqhHRLPjnvnfKEpyfqhHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqPjDdnHD)

Dlib提取人脸特征点（68点，opencv画图） (http://blog.csdn.net/zmdsjtu/article/details/53...



Dlib+opencv 68点特征点的使用以及绘图。

 zmdsjtu (http://blog.csdn.net/zmdsjtu) 2016年12月04日 14:31 17949

Dlib+OpenCV深度学习人脸识别 (http://blog.csdn.net/jcx0315/article/details/73449315)





Dlib+OpenCV深度学习人脸识别 前言 人脸识别在LWF(Labeled Faces in the Wild)数据集上人脸识别率现在已经99.7%以上，这个识别率确实非常高了，但是真实的环境...

 jcjx0315 (<http://blog.csdn.net/jcjx0315>) 2017年06月19日 01:10  3271

人脸识别(4)--Python3.6+dlib19.4识别实例 (<http://blog.csdn.net/u012842255/article/details/...>)

生成方形框识别人脸 关键线识别人脸前提条件： 确保python+dlib环境已经搭建成功。搭建步骤可以参考上一篇博客：<http://blog.csdn.net/u012842255/article/>...

 u012842255 (<http://blog.csdn.net/u012842255>) 2017年04月21日 23:06  1220





程序员跨越式成长指南

完成第一次跨越，你会成为具有一技之长的开发者，月薪可能翻上几番；完成第二次跨越，你将成为拥有局部优势或行业优势的专业人士，获得个人内在价值的有效提升和外在收入的大幅跃迁.....

(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjtzrjD0IZ0qnfK9ujYzP1f4PjnY0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1YLn1TYm1cYnAmLmW99mhwW0AwY5HDdnHc3rjb4P1c0lgF_5y9Y1Z0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq_lyd8Q1R4uWc4uHf3uAckPHRkPWN9PhcsmW9huWqdlAdxTvqdThP-5HDknWFBmhkEusKzujYk0AFV5H00TZcqn0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqnWbLnH0)



Dlib人脸特征点检测（速度优化） (http://blog.csdn.net/Leo_812/article/details/51945743)

Dlib

 Leo_812 (http://blog.csdn.net/Leo_812) 2016年07月18日 21:25  13932

使用dlib人脸识别的例子 (<http://blog.csdn.net/liukang325/article/details/55211814>)

来自官方的例子：http://dlib.net/face_detection_ex.cpp.html 做了一些修改：`#include #include #include #include u...`

 liukang325 (<http://blog.csdn.net/liukang325>) 2017年02月15日 16:47  2078



dlib人脸检测功能介绍 (<http://blog.csdn.net/xiamentingtao/article/details/50968514>)

本文主要介绍三个点： 1. 如何单独建立一个工程，使用dlib的人脸检测功能。 2. 提高人脸检测率的两个方法 3. 加速人脸检测的方法 下面围绕这几个点展开叙述。建人脸检测工程1. 首先...

 xiamentingtao (<http://blog.csdn.net/xiamentingtao>) 2016年03月24日 09:02  13472

OpenCV实践之路——用dlib库进行人脸检测与人脸标记（Python） (<http://blog.csdn.net/xin...>)

看人脸方面的资料的时候，会发现很多人都会提到dlib这个库，于是就安装尝试下这个库看看它到底有多神奇。今天只是初次尝试一下dlib到底怎么用。 安装dlib： 我的操作系统是window 7，安装...

 xingchenbingbuyu (<http://blog.csdn.net/xingchenbingbuyu>) 2016年04月11日 00:13  12067

利用Dlib进行人脸特征局部定位 (http://blog.csdn.net/wahaha1_/article/details/53114783)

```
void CDlib_MFCDlg::OnBnClickedFace() { // TODO: 在此添加控件通知处理程序代码 char img_file[]="G:\\Source\\111.jp...
```



wahaha1_ (http://blog.csdn.net/wahaha1_)

2016年11月10日 13:17

1145

DLIB 人脸识别 python代码 (<http://blog.csdn.net/zmschy2128/article/details/78546873>)

下面是python代码，基于dlib-19.4.0，python3.5 x64，个人感觉官方给定门限值0.6偏大，改为0.5左右更好。#-*- coding: utf-8 -*- import...

zmschy2128 (<http://blog.csdn.net/zmschy2128>)

2017年11月16日 08:38

45

dlib人脸检测 (http://blog.csdn.net/qq_18854309/article/details/78094225)

一、dlib Dlib是一个机器学习的C++库，包含了许多机器学习常用的算法。下载:<http://www.dlib.net/ml.html> 二、安转qt,设置环境变量，用qt自带的...

qq_18854309 (http://blog.csdn.net/qq_18854309)

2017年09月26日 13:28

198

Dlib+opencv实时提取人脸轮廓（windows环境下//Dlib配置入门） (<http://blog.csdn.net/zmdsytu>)

Windows环境下使用Dlib进行人脸特征点提取以及轮廓勾画//示例程序 可以商用的人脸特征点检测里不错的选择...

zmdsytu (<http://blog.csdn.net/zmdsytu>)

2016年09月03日 13:49

11553



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](#)

【dlib代码解读】人脸检测器的训练 (http://blog.csdn.net/elaine_bao/article/details/5304654...)

基于dlib训练自己的人脸检测器

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

2016年11月05日 20:22

6780

【深度学习】基于深度学习的人脸识别系统系列（Caffe+OpenCV+Dlib） (http://blog.csdn.net/Taily_Duan)

基于深度学习的人脸识别系统系列（Caffe+OpenCV+Dlib）——【二】人脸检测与预处理接口的设计 前言 基于深度学习的人脸识别系统，一共用到了5个开源库：OpenC...

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

2016年12月28日 16:41

2365



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](#)

Dlib机器学习库学习系列三----人脸对齐（特征点检测） (http://blog.csdn.net/Sunshine_in_Moon)

本篇博客是Dlib库学习的第三篇---人脸对齐。人脸对齐与人脸检测工程建立与配置基本相同，在此不再赘述。可参照我上一篇博客。闲话少说，来点干货。 步骤一：建立并配置工程，参照上一篇博客。 ...

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

2015年12月02日 16:51

15570



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](#)

dlib人脸特征点对齐 (http://blog.csdn.net/xiamentingtao/article/details/51396948)

dlib人脸特征点对齐前面我们介绍了使用dlib进行人脸检测，下面我们给出如何使用dlib进行人脸特征点检测。我们直接贴出代码。我们的代码包括如下几部分功能： 检测单张图片 检测一个视频 检测一个ca...

 xiamentingtao (<http://blog.csdn.net/xiamentingtao>) 2016年05月13日 17:15  5785

-  6
- 
- 
- 
- 