





```
#include <opencv.hpp>
using namespace cv;
// define
Mat Facedetect(Mat frame);
//dlib的配置函数 后面几章会讲
void Dlib_Prodefine();

1FaceRotate.h：（用于关键点矫正）
2
3#include <dlib/image_processing/frontal_face_detector.h>
4#include <dlib/image_processing/render_face_detections.h>
5#include <dlib/image_processing.h>
6#include <dlib/gui_widgets.h>
7#include <dlib/image_io.h>
8#include <dlib/opencv/cv_image.h>
9#include <dlib/opencv.h>
10
11using namespace dlib;
12
13frontal_face_detector detector = get_frontal_face_detector();
14shape_predictor sp;//Already get

1 FaceProcessing.h：（用于对人脸进行预处理）
2#include <opencv2/opencv.hpp>
3
4using namespace std;
5using namespace cv;
6Mat FaceProcessing(const Mat &img_, double gamma = 0.2, double sigma0 = 1, double sigma1 = -2, double mask = 0, double do_norm = 10);
7
81FaceProcessing.cpp：
9
102
113
124
135
146
157
168
179
20#include "FaceProcessing.h"
21
22int gauss(float x[], float y[], int length, float sigma);
23Mat gaussianfilter(Mat img, double sigma0, double sigma1, double shift1, double shift2);
24Mat FaceProcessing(const Mat &img_, double gamma = 0.2, double sigma0 = 1, double sigma1 = -2, double mask = 0, double do_norm = 10);
25
26//找出矩阵中的最大值或最小值，输入MAX，或MIN
27double MatMaxMin(Mat im, String flag = "MAX")
28{
29    double value = im.ptr<float>(0)[0];
30    if (flag == "MAX")
31    {
32        for (int i = 0; i<im.rows; i++)
33            for (int j = 0; j<im.cols; j++)
34                if (im.ptr<float>(i)[j]>value)
35                    value = im.ptr<float>(i)[j];
36        return value;
37    }
38    else if (flag == "MIN")
39    {
40        for (int i = 0; i<im.rows; i++)
41            for (int j = 0; j<im.cols; j++)
42                if (im.ptr<float>(i)[j]<value)
43                    value = im.ptr<float>(i)[j];
44        return value;
45    }
46    return -1;
47}
```



在线课程



([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-)



Python 空战工程师 5HR1jrfkn100T1Y4m1fzryR4nW7BuymzmWIB0AwY5HDdnHc3rj0Lnjn0lgF\_5y9Y1Z0IQzq-  
([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-)



Python 空战工程师 5HR1jrfkn100T1Y4m1fzryR4nW7BuymzmWIB0AwY5HDdnHc3rj0Lnjn0lgF\_5y9Y1Z0IQzq-  
([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-)



Python 空战工程师 5HR1jrfkn100T1Y4m1fzryR4nW7BuymzmWIB0AwY5HDdnHc3rj0Lnjn0lgF\_5y9Y1Z0IQzq-  
([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-)



Python 空战工程师 5HR1jrfkn100T1Y4m1fzryR4nW7BuymzmWIB0AwY5HDdnHc3rj0Lnjn0lgF\_5y9Y1Z0IQzq-  
([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-)



Python 空战工程师 5HR1jrfkn100T1Y4m1fzryR4nW7BuymzmWIB0AwY5HDdnHc3rj0Lnjn0lgF\_5y9Y1Z0IQzq-  
([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-)



Python 空战工程师 5HR1jrfkn100T1Y4m1fzryR4nW7BuymzmWIB0AwY5HDdnHc3rj0Lnjn0lgF\_5y9Y1Z0IQzq-  
([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-)



Python 空战工程师 5HR1jrfkn100T1Y4m1fzryR4nW7BuymzmWIB0AwY5HDdnHc3rj0Lnjn0lgF\_5y9Y1Z0IQzq-  
([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-)



Python 空战工程师 5HR1jrfkn100T1Y4m1fzryR4nW7BuymzmWIB0AwY5HDdnHc3rj0Lnjn0lgF\_5y9Y1Z0IQzq-  
([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-)



Python 空战工程师 5HR1jrfkn100T1Y4m1fzryR4nW7BuymzmWIB0AwY5HDdnHc3rj0Lnjn0lgF\_5y9Y1Z0IQzq-  
([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-)



Python 空战工程师 5HR1jrfkn100T1Y4m1fzryR4nW7BuymzmWIB0AwY5HDdnHc3rj0Lnjn0lgF\_5y9Y1Z0IQzq-  
([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0IZ0qnK9ujYzP1mznWR10Aw-)



app开发报价单



15327

【Halcon】Halcon与OpenCV介绍、比较  
([http://blog.csdn.net/taily\\_duan/article/details/51499769](http://blog.csdn.net/taily_duan/article/details/51499769))

13783

【OpenCV】立体匹配算法 StereoBM/StereoSGM/StereoVar  
([http://blog.csdn.net/taily\\_duan/article/details/52165458](http://blog.csdn.net/taily_duan/article/details/52165458))

9125

```
//高斯滤波
Mat gaussianfilter(Mat img, double sigma0, double sigma1, double shift1 = 0, double shift2 = 0)
{
    int i, j;
    sigma0 = (float)sigma0;
    sigma1 = (float)sigma1;
    shift1 = (float)shift1;
    shift2 = (float)shift2;
    Mat img2 = img;
    Mat img3 = img;
    Mat imgResult;

    //将数据存入横向高斯模板中
    int rowLength = (int)(floor(3.0*sigma0 + 0.5 - shift1) - ceil(-3.0*sigma0 - 0.5 - shift1) + 1);
    int rowBegin = (int)ceil(-3.0*sigma0 - 0.5 - shift1);
    float rowArray[30], Gx[30];
    for (i = 0; i < rowLength; i++)
    {
        rowArray[i] = rowBegin + i;
    }
    gauss(rowArray, Gx, rowLength, sigma0);
    Mat kx = Mat(1, rowLength, CV_32F); //转换成mat类型

    float *pData1 = kx.ptr<float>(0);
    for (i = 0; i < rowLength; i++)
    {
        pData1[i] = Gx[i];
    }
    //将数据存入纵向高斯模板中
    int colLength = (int)(floor(3.0*sigma1 + 0.5 - shift2) - ceil(-3.0*sigma1 - 0.5 - shift2) + 1);
    int colBegin = (int)ceil(-3.0*sigma1 - 0.5 - shift2);
    float colArray[30], Gy[30];
    for (i = 0; i < colLength; i++)
    {
        colArray[i] = colBegin + i;
    }
    gauss(colArray, Gy, colLength, sigma1);
    Mat ky = Mat(colLength, 1, CV_32F);
    float *pData2;
    for (i = 0; i < colLength; i++)
    {
        pData2 = ky.ptr<float>(i);
        pData2[0] = Gy[i];
    }
    filter2D(img, img2, img.depth(), kx, Point(-1, -1));
    filter2D(img2, imgResult, img2.depth(), ky, Point(-1, -1));
    return imgResult;
}

//行列卷积
int gauss(float x[], float y[], int length, float sigma)
{
    int i;
    float sum = 0.0;
    for (i = 0; i < length; i++)
    {
        x[i] = exp(-pow(x[i], 2) / (2 * pow(sigma, 2)));
        sum += x[i];
    }
    for (i = 0; i < length; i++)
    {
        y[i] = x[i] / sum;
    }
    return 1;
}
```



内容举报



返回顶部



内容举报

```
Mat FaceProcessing(const Mat &img_, double gamma, double sigma0, double sigma1, double mask, double do_norm)
{
    Mat img;
    img_.convertTo(img, CV_32F);
    Mat imT1, imT2;
    int rows = img.rows;
    int cols = img.cols;
    Mat im = img;
    int b = floor(3 * abs(sigma1)); //左右扩充边缘的距离
    Mat imtemp(Size(cols + 2 * b, rows + 2 * b, CV_32F, Scalar(0))); //保存扩充的图形
    Mat imtemp2(Size(cols, rows), CV_32F, Scalar(0));
    float s = 0.0;
    //Gamma correct input image to increase local contrast in shadowed regions.
    if (gamma == 0)
    {
        double imixeltemp = 0;
        double Max = MatMaxMin(im, "MAX"); //等价于max(1,max(max(im)))
        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++)
            {
                imixeltemp = log(im.ptr<float>(i)[j] + Max / 256);
                im.ptr<float>(i)[j] = imixeltemp;
            }
    }
    else
    {
        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++)
                im.ptr<float>(i)[j] = pow(im.ptr<float>(i)[j], gamma);
    }
    float *pData1;
    //run prefilter, if any
    if (sigma1)
    {
        double border = 1;
        if (border) //add extend-as-constant image border to reduce
            //boundary effects
        {
            for (int i = 0; i < rows + 2 * b - 1; i++)
            {
                pData1 = imtemp.ptr<float>(i);
                for (int j = 0; j < cols + 2 * b - 1; j++){
                    //中间
                    if (i >= b && i < rows + b && j >= b && j < cols + b)
                        pData1[j] = im.ptr<float>(i - b)[j - b];
                    //左上
                    else if (i < b && j < b)
                        pData1[j] = im.ptr<float>(0)[0];
                    //右上
                    else if (i < b && j >= cols + b && j < cols + 2 * b)
                        pData1[j] = im.ptr<float>(0)[cols - 1];
                    //左下
                    else if (i >= rows + b && i < rows + 2 * b && j < b)
                        pData1[j] = im.ptr<float>(rows - 1)[0];
                    //右下
                    else if (i >= rows + b && j >= cols + b)
                        pData1[j] = im.ptr<float>(rows - 1)[cols - 1];
                    //上方
                    else if (i < b && j >= b && j < cols + b)
                        pData1[j] = im.ptr<float>(0)[j - b];
                    //下方
                    else if (i >= rows + b && j >= b && j < cols + b)
                        pData1[j] = im.ptr<float>(rows - 1)[j - b];
                    //左方
                }
            }
        }
    }
}
```

 [返回顶部](#)



app开发报价单

 [内容举报](#)

 [返回顶部](#)





0



```

        else if (j<b&&i >= b&&i<im.rows + b)
            pData1[j] = im.ptr<float>(i - b)[0];
        //右方
        else if (j >= im.cols + b&&i >= b&&i<im.rows + b)
            pData1[j] = im.ptr<float>(i - b)[im.cols - 1];/**/
    }
}

else
{
    if (sigma0>0)
    {
        imT1 = gaussianfilter(imtemp, sigma0, sigma0);
        imT2 = gaussianfilter(imtemp, -sigma1, -sigma1);
        imtemp = imT1 - imT2;
        //imtemp=gaussianfilter(imtemp,sigma0,sigma0)-gaussianfilter(imtemp,-sigma1,-sigma1);
    }
    else
        imtemp = imtemp - gaussianfilter(imtemp, -sigma1, -sigma1);
}

if (border)
{
    //再取回中间部分
    for (int i = 0; i<rows; i++)
    {
        pData1 = im.ptr<float>(i);
        for (int j = 0; j<cols; j++)
            pData1[j] = imtemp.ptr<float>(i + b)[j + b];
    }
    // test=im.ptr<float>(19)[19];
}

/*
% Global contrast normalization. Normalizes the spread of output
% values. The mean is near 0 so we don't bother to subtract
% it. We use a trimmed robust scatter measure for resistance to
% outliers such as specularities and image borders that have
% different values from the main image. Usually trim is about
% 10.
*/

if (do_norm)
{
    double a = 0.1;
    double trim = abs(do_norm);

    //im = im./mean(mean(abs(im).^a)^(1/a);
    imtemp2 = abs(im);

    //cvPow(&im,&im,a)//为每个元素求pow
    for (int i = 0; i<rows; i++)
    {
        pData1 = imtemp2.ptr<float>(i); //imtemp2为零矩阵
        for (int j = 0; j<cols; j++)
            pData1[j] = pow(imtemp2.ptr<float>(i)[j], a);
    }

    //求平均值s
    s = 0.0;
    for (int i = 0; i<rows; i++)
    {
        pData1 = imtemp2.ptr<float>(i);

```



0



内容举报



返回顶部





0



```
        for (int j = 0; j<cols; j++)
            s += imtemp2.ptr<float>(i)[j];
    }
    s /= (im.rows*im.cols);
    double temp = pow(s, 1 / a);
    for (int i = 0; i<rows; i++)
    {
        pData1 = im.ptr<float>(i);
        for (int j = 0; j<cols; j++)
            pData1[j] = pData1[j] / temp;//点除
    }

    //im = im./mean(mean(min(trim,abs(im)).^a)^(1/a);
    imtemp2 = abs(im);
    for (int i = 0; i<rows; i++)
    {
        pData1 = imtemp2.ptr<float>(i);
        for (int j = 0; j<cols; j++)
            if (pData1[j]>trim)
                pData1[j] = trim;//min(trim,abs(im))
    }
    //cvPow(&im,&im,a);///为每个元素求pow
    for (int i = 0; i<rows; i++)
    {
        pData1 = imtemp2.ptr<float>(i);
        for (int j = 0; j<cols; j++)
            pData1[j] = pow(pData1[j], a);
    }
    //求平均值
    s = 0.0;
    for (int i = 0; i<rows; i++)
    {
        pData1 = imtemp2.ptr<float>(i);
        for (int j = 0; j<cols; j++)
            s += pData1[j];
    }
    s /= (im.rows*im.cols);
    temp = pow(s, 1 / a);//
    for (int i = 0; i<rows; i++)
    {
        pData1 = im.ptr<float>(i);
        for (int j = 0; j<cols; j++)
            pData1[j] = pData1[j] / temp;//点除
    }
}

if (do_norm>0)
{
    //im = trim*tanh(im/trim);
    for (int i = 0; i<rows; i++)
    {
        pData1 = im.ptr<float>(i);
        for (int j = 0; j<cols; j++)
            pData1[j] = trim*tanh(pData1[j] / trim);
    }
}

}

//归一化处理
double Min;
Min = MatMaxMin(im, "MIN");//找到矩阵的最小值
for (int i = 0; i<rows; i++)
{
    pData1 = im.ptr<float>(i);
    for (int j = 0; j<cols; j++)
        pData1[j] += Min;
}
//im.convertTo(im, CV_32F, 1.0/255.0);
```

内容举报

返回顶部



app开发报价单

```

        normalize(im, im, 0, 255, NORM_MINMAX);
        //normalize(m,im,0,255,NORM_MINMAX);
        /* for(int i=0;i<rows;i++)
        {
            pData1=im.ptr<float>(i);
            for(int j=0;j<cols;j++)
                pData1[j]*=255;
        }*/
        im.convertTo(im, CV_8UC1);
        return im;
    }
}

```

1 FaceDetect.cpp :

2

3

```

4 #include <FaceDetect.h>
5 #include <FaceRotate.h>
6 #include <FaceProcessing.h>
7 void Dlib_Prodefine()
8 {
9     deserialize("shape_predictor_68_face_landmarks.dat") >> sp; //读入标记点文件
10 }
11 Mat Facedetect(Mat frame)//脸是否存在
12 {
13     Mat gray,error;
14     cvtColor(frame, gray, CV_BGR2GRAY);
15     int * pResults = NULL;
16     pResults = facedetect_frontal_tmp((unsigned char*)(gray.ptr(0)), gray.cols, gray.rows, gray.step, 1.2f, 5, 24);
17     int peopleNUM = (pResults ? *pResults : 0);
18     for (int i = 0; i < peopleNUM; i++)//代表有几张人脸(pResults ? *pResults : 0)
19     {
20         short * p = ((short*)(pResults + 1)) + 6 * i;
21         Rect opencvRect(p[0], p[1], p[2], p[3]);
22         //gray=gray(opencvRect);
23         dlib::rectangle dlibRect((long)opencvRect.tl().x, (long)opencvRect.tl().y, (long)opencvRect.br().x - 1, (long)opencvRect.br().y - 1);
24         dlib::full_object_detection shape = sp(dlib::cv_image<uchar>(gray), dlibRect); //标记点
25         std::vector<dlib::full_object_detection> shapes;
26         shapes.push_back(shape); //把点保存在了shape中
27         dlib::array<array2d<rgb_pixel>> face_chips;
28         extract_image_chips(dlib::cv_image<uchar>(gray), get_face_chip_details(shapes), face_chips);
29         Mat pic = toMat(face_chips[0]);
30         cvtColor(pic, pic, CV_BGR2GRAY);
31         resize(pic, pic, Size(224, 224));
32         return FaceProcessing(pic);
33     }
34     return error; //如果没有检测出人脸 将返回一个空矩阵
35 }

```

32 1 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：

```

33 2 dlib::array<array2d<rgb_pixel>> face_chips;
34 3     extract_image_chips(dlib::cv_image<uchar>(gray), get_face_chip_details(shapes), face_chips);
35 4     Mat pic = toMat(face_chips[0]);

```

36 5 1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。

37 6 2将Mat类型转换为array2d< rgb\_pixel >则可以用：

```

38 7 3     Mat gray;
39 8 4     dlib::cv_image<uchar>(gray);

```

40 9 1在这个地方，我们特别需要注意，还要转换一次灰度：

```

41 10 2     cvtColor(pic, pic, CV_BGR2GRAY);

```

内容举报

返回顶部



app开发报价单

内容举报

返回顶部

```
42 11         resize(pic, pic, Size(224, 224));
43 12         return FaceProcessing(pic);
44 131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
45 142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。
46 153我们可以看看识别的效果。调用FaceDetect()函数接口：
47 16         Dlib_Prodefine();
48 17         //Caffe_Prodefine();
49 18         Mat lena = imread("lena.jpg");
50 19         imshow("Face Detect", Facedetect(lena));
51 201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
52 212imshow.
53 223检测与处理图片效果显示：
54 234
55 245
56 25
57 26
58 27
59 28
60 29
61 30
62 31
63 32
64 33
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
```



app开发报价单



app开发报价单

内容举报

返回顶部





- 93
- 94
- 95
- 96
- 97
- 98
- 99
- 100
- 101
- 102
- 103
- 104
- 105
- 106
- 107
- 108
- 109
- 110
- 111
- 112
- 113
- 114
- 115
- 116
- 117
- 118
- 119
- 120
- 121
- 122
- 123
- 124
- 125
- 126
- 127
- 128
- 129
- 130
- 131
- 132
- 133
- 134
- 135
- 136
- 137
- 138
- 139
- 140
- 141
- 142
- 143



  
内容举报

  
返回顶部



144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
---



  
内容举报

  
返回顶部



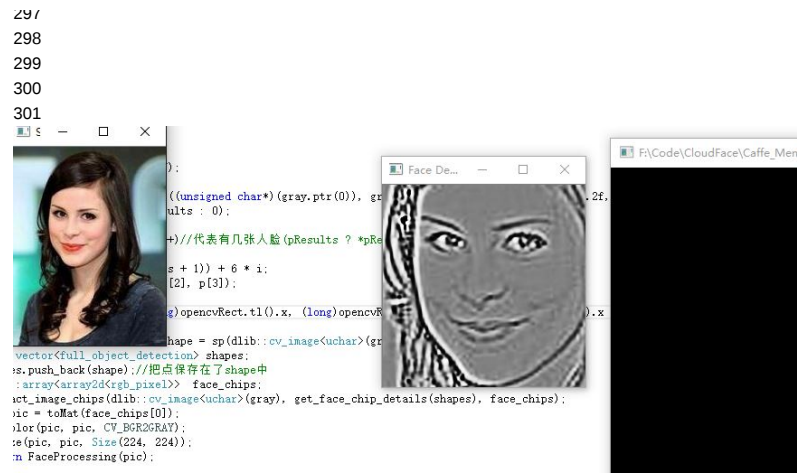
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245





246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297





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。

发表评论

(http://my.csdn.net/weixin\_35068028)

相关文章推荐

**Dlib+OpenCV深度学习人脸识别 (http://blog.csdn.net/jcix0315/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...


jcix0315 (http://blog.csdn.net/jcix0315) 2017年06月19日 01:10 3266

基于深度学习的人脸识别系统系列（Caffe+OpenCV+Dlib）——【一】如何在Visual Studio中...  
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：OpenCV（计算机视觉库）、Caffe（深度学习库）、Dlib（机器学习库）、libfacedetection（人脸检测库）、cudnn（gpu...

内容举报

返回顶部

app开发报价单

 Mr\_Curry ([http://blog.csdn.net/Mr\\_Curry](http://blog.csdn.net/Mr_Curry)) 2016年09月05日 20:35 12886




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？他们是这么说...

([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjnvPjc0lZ0qnfK9ujYzP1f4PjDs0Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1dbuWuhPHDdPhu-PHR4ujlW0AwY5HDdnHc3rj0Lnjn0lgF\\_5y9YlZ0lQzq-uZR8mLPbUB48ugfElAqspynETZ-YpAq8nWqdAdxTvqdThP-5yF\\_UvTkn0KzujYk0AFV5H00TZcqn0KdpyfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWTnqnW0YP1T](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjnvPjc0lZ0qnfK9ujYzP1f4PjDs0Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1dbuWuhPHDdPhu-PHR4ujlW0AwY5HDdnHc3rj0Lnjn0lgF_5y9YlZ0lQzq-uZR8mLPbUB48ugfElAqspynETZ-YpAq8nWqdAdxTvqdThP-5yF_UvTkn0KzujYk0AFV5H00TZcqn0KdpyfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWTnqnW0YP1T))


基于深度学习的人脸识别系统系列（Caffe+OpenCV+Dlib）——【三】使用Caffe的MemoryD...

本篇是该系列的第三篇博客，介绍如何使用VGG网络模型与Caffe的 MemoryData层去提取一个OpenCV矩阵类型Mat的特征。##思路 VGG网络模型是牛津大学视觉几何组提出的一种深度模型，...

 Mr\_Curry ([http://blog.csdn.net/Mr\\_Curry](http://blog.csdn.net/Mr_Curry)) 2016年09月07日 10:19 12458

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...

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：OpenCV（计算机视觉库）、Caffe（深度学习库）、Dlib（机器学习库）、libfacedetection（人脸检测库）、cudnn（gpu...

 Mr\_Curry ([http://blog.csdn.net/Mr\\_Curry](http://blog.csdn.net/Mr_Curry)) 2016年09月06日 20:53 13213



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




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\\_pyfqHmknjzrjD0lZ0qnfK9ujYzP1f4PjnY0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1YsPWnzmyF-PvnYnvw-PjRv0AwY5HDdnHc3rj0Lnjn0lgF\\_5y9YlZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq\\_ljd8Q1R4uWc4uHf3uAckPHRkPWN9PhcsmW9huWqdAdxTvqdThP-5HDknWFBmhcEusKzujYk0AFV5H00TZcqn0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWTnqnPj4PHc](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjzrjD0lZ0qnfK9ujYzP1f4PjnY0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1YsPWnzmyF-PvnYnvw-PjRv0AwY5HDdnHc3rj0Lnjn0lgF_5y9YlZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq_ljd8Q1R4uWc4uHf3uAckPHRkPWN9PhcsmW9huWqdAdxTvqdThP-5HDknWFBmhcEusKzujYk0AFV5H00TZcqn0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWTnqnPj4PHc))

基于深度学习的人脸识别系统系列（Caffe+OpenCV+Dlib）——【四】使用CUBLAS加速计算...

前言 本篇是该系列的第四篇博客，介绍如何使用CUBLAS加速进行两个向量间余弦距离的计算。##思路 我们先来温习一下两个向量之间余弦距离的数学公式，大家自己可以回忆一下：x,y均为同维度的向量...

 Mr\_Curry ([http://blog.csdn.net/Mr\\_Curry](http://blog.csdn.net/Mr_Curry)) 2016年09月09日 21:25 6096

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](http://dlib.net/face_detection_ex.cpp.html) 做了一些修改：`#include #include #include #include u...`



内容举报



返回顶部



内容举报



返回顶部

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

深度学习 VGG 网络 实现 face landmark 与 head pose (http://blog.csdn.net/luojun2007/arti...

深度学习实现 人脸特征点（68点）标注及人脸3D姿态估计

luojun2007 (http://blog.csdn.net/luojun2007) 2016年08月09日 14:26 1713

Haar、pico、npd、dlib等多种人脸检测特征及算法结果比较 (http://blog.csdn.net/wishchin/...

Pico（Pixel Intensity Comparison-based Object detection）发表于2014年，不同于VJ的Haar特征，pico则是提取点对特征，对两个像素点进行对比...

wishchin (http://blog.csdn.net/wishchin) 2017年05月04日 11:44 1456

深度学习与人脸识别系列（2）\_\_基于VGNet的人脸识别系统 (http://blog.csdn.net/gqixf/arti...

作者：wjmishuai 出处：http://blog.csdn.net/wjmishuai/article/details/50854155 1.引言 本文中介绍的人脸识...

gqixf (http://blog.csdn.net/gqixf) 2017年09月19日 15:27 118

基于深度学习的人脸识别系统系列——使用CUBLAS加速计算人脸向量的余弦距离 (http://blog...

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：OpenCV（计算机视觉库）、Caffe（深度学习库）、Dlib（机器学习库）、libfacedetection（人脸检测库）、cudnn（g...

Allyii0022 (http://blog.csdn.net/Allyii0022) 2017年07月04日 16:09 555

基于深度学习的人脸识别系统系列（Caffe+OpenCV+Dlib）——【三】使用Caffe的MemoryD...

原文地址：http://m.blog.csdn.net/article/details?id=52456548 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：OpenCV...

w113691 (http://blog.csdn.net/w113691) 2016年12月17日 10:40 1037

Dlib编译与搭建 (http://blog.csdn.net/u014114636/article/details/56678572)

windows上编译 cmake gui + vs cmake 编译生成之后sln解决方案 VS打开分别在Release和Debug下生成，可以生成相应的lib文件 cmake cmd cd ex...

u014114636 (http://blog.csdn.net/u014114636) 2017年02月23日 19:24 754

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 12059



2017.12.10

内容举报

返回顶部



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

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...

zmdsjtu (http://blog.csdn.net/zmdsjtu) 2016年09月03日 13:49 11544



表情识别的简单实现（实时,C++,SVM+Dlib）(http://blog.csdn.net/zmdsjtu/article/details/5...

SVM+Dlib实现表情识别

zmdsjtu (http://blog.csdn.net/zmdsjtu) 2016年12月15日 13:17 9266

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 16436

基于opencv+Dlib的面部合成（Face Morph）(http://blog.csdn.net/wangxing233/article/de...

零、前言前段时间看到文章【1】和【2】，大概了解了面部合成的基本原理。这两天空下来了，于是参考【3】自己实现了下。虽然【1】和【2】已经讲的很清楚了，但是有一些细节没有提到。所以我在这里记录一下实现的...

wangxing233 (http://blog.csdn.net/wangxing233) 2016年05月31日 22:39 8534

基于opencv和Dlib的人脸交换(face swap) (http://blog.csdn.net/wangxing233/article/details...

基于opencv和Dlib的人脸交换(face swap)主要分为两个步骤：人脸对齐(face alignment)和无缝融合(Seamless Cloning,可能翻译的不准确)。其中人脸对齐又分为人...

wangxing233 (http://blog.csdn.net/wangxing233) 2016年06月27日 21:00 4340

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 17930



内容举报



返回顶部

