


```
24 SingleFace Recognition(Mat input_224, SingleFace &singleface);//输入一个224*224的图片，查找他属于哪个人
25 vector<SingleFace> FaceArray;//众多的SingleFace
26
27 //仅仅用于人脸检测
28 void drawFaceImage(Mat input);//多线程有用
29 private:
30     caffe::MemoryDataLayer<float> *memory_layer;//进行数据输入的层
31     caffe::Net<float>* net;//整个layer和权重
32     bool FaceDetect(Mat input, Rect &roi);//人脸检测的接口
33     mutex thread_mutex;//线程锁
34 };
```

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

YotoFace.cpp :

```
1 #include <YotoFace.h>
2 #include <facedetect-dll.h>
3
4 namespace caffe
5 {
6     extern INSTANTIATE_CLASS(InputLayer);
7     extern INSTANTIATE_CLASS(InnerProductLayer);
8     extern INSTANTIATE_CLASS(DropoutLayer);
9     extern INSTANTIATE_CLASS(ConvolutionLayer);
10    REGISTER_LAYER_CLASS(Convolution);
11    extern INSTANTIATE_CLASS(ReLULayer);
12    REGISTER_LAYER_CLASS(ReLU);
13    extern INSTANTIATE_CLASS(PoolingLayer);
14    REGISTER_LAYER_CLASS(Pooling);
15    extern INSTANTIATE_CLASS(LRNLayer);
16    REGISTER_LAYER_CLASS(LRN);
17    extern INSTANTIATE_CLASS(SoftmaxLayer);
18    REGISTER_LAYER_CLASS(Softmax);
19    extern INSTANTIATE_CLASS(MemoryDataLayer);
20 }
21
22 //构造
23 YotoFace::YotoFace()
24 {
25     net = new caffe::Net<float>("vgg_extract_feature_memorydata.prototxt", caffe::TEST);
26     net->CopyTrainedLayersFrom("VGG_FACE.caffemodel");
27     memory_layer = (caffe::MemoryDataLayer<float> *)net->layers()[0].get();
28 }
29
30 //提取特征
31 vector<float> YotoFace::ExtractFeature(Mat img_224) //ensure input 224*224!!!
32 {
33     std::vector<Mat> test{ img_224 };
34     std::vector<int> testLabel{ 0 };
35     memory_layer->AddMatVector(test, testLabel);// memory_layer and net , must be define be a global variable.
```

内容举报

返回顶部



内容举报



2

```
36 vector<caffe::Blob<float>*> input_vec;
37 net->Forward(input_vec);
38 auto fc7 = net->blob_by_name("fc7");//提取fc7层！4096维特征
39 float* begin = fc7->mutable_cpu_data();
40 vector<float> feature{ begin, begin + fc7->channels() };
41 //cout << fc7->channels();

42 return move(feature);
43 }
44
45 //私有的人脸检测函数
46 bool YotoFace::FaceDetect(Mat input, Rect &roi)
47 {
48     thread_mutex.lock();
49     {
50         Mat gray;
51         cvtColor(input, gray, CV_BGR2GRAY);
52         int * pResults = NULL;
53         pResults = facedetect_multiview_reinforce((unsigned char*)gray.ptr(0), gray.cols, gray.rows, gray.step,
54             1.2f, 5, 24);
55         //dlib.
56         //
57         int p_num = (pResults ? *pResults : 0);
58         if (p_num == 0)
59         {
60             thread_mutex.unlock();
61             return false;
62         }
63
64         short * p = ((short*)(pResults + 1));
65         Point left(p[0], p[1]);
66         Point right(p[0] + p[2], p[1] + p[3]);
67         roi = Rect(left, right);
68         thread_mutex.unlock();
69         return true;
70     }
71 }
72
73 //单张图片生成SingleFace
74 bool YotoFace::Generate(Mat input, SingleFace &singleface)
75 {
76     Rect roi;
77     if (FaceDetect(input, roi))
78     {
79         Mat img_224 = input(roi);
80         resize(img_224, img_224, Size(224, 224));
81         auto feature_ = ExtractFeature(img_224);
82
83         if (feature_.empty())
84             return false;
85         else
86         {
87             singleface.sourceImage = input;
```



返回顶部



内容举报



返回顶部





```
88     singleface.position = roi;
89     singleface.feature = feature_;
90     singleface.Roi_224 = img_224;
91     return true;
92 }
93 }
94 else
95 {
96     return false;
97 }
98 }
99
100 //有标签的数据
101 bool YotoFace::Generate(Mat input, SingleFace &singleface,string label_)
102 {
103     Rect roi;
104     if (label_.empty())
105         return false;
106     if (FaceDetect(input, roi))
107     {
108         Mat img_224 = input(roi);
109         resize(img_224, img_224, Size(224, 224));
110         auto feature_ = ExtractFeature(img_224);
111
112         if (feature_.empty())
113             return false;
114         else
115         {
116             singleface.sourceImage = input;
117             singleface.position = roi;
118             singleface.feature = feature_;
119             singleface.Roi_224 = img_224;
120             singleface.label = label_;
121             return true;
122         }
123     }
124
125     else
126     {
127         return false;
128     }
129
130 void YotoFace::drawFaceImage(Mat input)
131 {
132     Rect rec;
133     if (FaceDetect(input, rec))
134     {
135         //有人脸
136         Mat draw = input;
137         rectangle(draw, rec, Scalar(0, 0, 255), 2);
138     }
139 }
140 }
```





这里解释一下YotoFace::FaceDetect这个函数需要加锁的原因。在具体落实到Qt上的时候，我们应该是希望有一个窗口是始终在检测人脸的，这就需要死循环，若不能跳出则会影响其他程序的执行。所以要采用多线程。而如果我们用的是libfacedetection，在两条线程同时调用函数时将会出错，所以要保证同时段只能调用一次。

thread_mutex.lock()表示锁上当前的线程，当别的线程碰到它时，会处于挂起状态，等待唤醒。

thread_mutex.unlock()进行解锁。

问题来了。人脸也检测了，特征也提取了，如何计算向量距离？

LikeValue的实现是？

ComputeDistance.cpp:

```
1  #include <YotoFace.h>
2
3  inline double LikeValue(float *v1, float *v2, int channels)
4  {
5      //计算内积：
6      double mult = 0;
7      double v1_2 = 0;
8      double v2_2 = 0;
9      for (int i = 0; i < channels; i++)
10     {
11         mult += v1[i] * v2[i];
12         v1_2 += pow(v1[i], 2);
13         v2_2 += pow(v2[i], 2);
14     }
15
16     return mult / (sqrt(v1_2)*sqrt(v2_2));
17 }
18
19
20 SingleFace YotoFace::Recognition(Mat input_, SingleFace &singleface)
21 {
22     //解析：
23     if (Generate(input_, singleface))
24     {
25         float *single_feature = &singleface.feature[0];
26         int single_channel = singleface.feature.size();
27
28         int size_ = FaceArray.size();//有多少个人脸需要对比的
29         vector<double> like_array;
```



2



```
29     vector<double> like_array;
30     for (int i = 0; i < size_; i++)
31     {
32         float *faces_feature = &FaceArray[i].feature[0];
33         like_array.push_back(LikeValue(single_feature, faces_feature, single_channel));
34     }
35
36     vector<double>::iterator biggest = std::max_element(std::begin(like_array), std::end(like_array));
37     int max_ = distance(std::begin(like_array), biggest);
38
39     return FaceArray[max_];
40
41 }
42
43 else
44 {
45     return singleface;
46 }
47
48 }
```

转成数组运算，速度更快。

实际上在应用时，在未按注册按钮时，A窗口是通过第2个线程调用的drawFaceImage()：



按下确认按钮的一瞬间，主线程也会调用drawFaceImage()中的FaceDetect()，由于加了锁，所以很安全啦。



⚠

内容举报

⬆

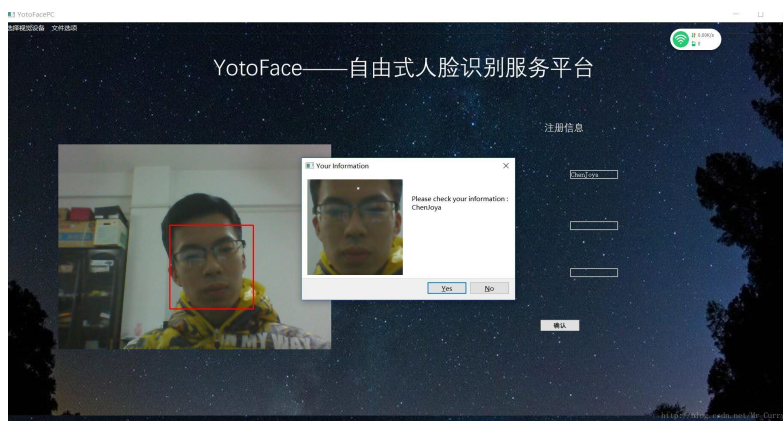
返回顶部

⚠

内容举报

⬆

返回顶部



识别的时候，调用的是Recognition()函数，也有锁保护。



结语

无GPU环境，速度杠杠的。

版权声明：本文为博主原创文章，未经博主允许不得转载。

本文已收录于以下专栏：

深度学习的具体程序应用——Caffe带给我们的可能 (<http://blog.csdn.net/column/details/13863.html>)



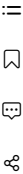
发表你的评论



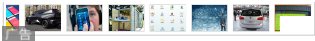
内容举报

返回顶部





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



wp434448785 (wp434448785) 2017-11-09 11:14 10楼
(wp434448785) 好，楼主能不能分享一下源码，非常感谢，邮箱：434448785@qq.com
回复

mikael_hu (/mikael_hu) 2017-07-21 16:57 9楼
(/mikael_hu) 这篇文章帮助很大，楼主能否分享一下源代码，多谢！我的邮箱：hnc0823@qq.com
回复

sstriving (/sstriving) 2017-04-27 21:14 8楼
(/sstriving) 为啥用你的程序会出现这样的错误
/home/ryan/FaceDetect/judgeface.cpp:4: error: undefined reference to `caffe::Net<float>::Net(std::string const&, c
affe::Phase, int, std::vector<std::string, std::allocator<std::string> > const*)'
/home/ryan/FaceDetect/judgeface.cpp:15: error: undefined reference to `caffe::Blob<float>::mutable_cpu_data
/home/ryan/build-FaceDetect-Desktop-Debug/judgeface.o:1: In function `std::string* google::MakeCheckOpString
<int, int>(int const&, int const&, char const*)':
弄了一天了 也没弄好 求解答 谢谢
回复

查看 13 条热评

相关文章推荐

Qt+Caffe+OpenCV——【一个基于VGG网络的人脸识别考勤系统】（二）创建一个登录界面和...

前言本次重点阐述在http://blog.csdn.net/mr_curry/article/details/53859654《Qt+Caffe+OpenCV——【一个基于VGG网络的人脸识别考勤系统...

Mr_Curry (http://blog.csdn.net/Mr_Curry) 2017年01月06日 20:45 3314

深度学习与人脸识别系列（5）__利用训练好的vgg模型进行人脸识别(利用摄像头) (http://blog....

利用训练好的vgg模型进行人脸识别(利用摄像头)

zchang81 (http://blog.csdn.net/zchang81) 2017年05月05日 12:56 1011



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_pyfqHmknjnvPjc0lZ0qnfK9uYzP1f4PjDs0Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1YsPANBPYNbnhPhnWK9uWTY0AwY5HDdnHc3rj0LPW60lgF_5y9YI0IQzq-uZR8mLPbUB48ugfEIAqspynETZ-YpAq8nWqdlAdxTvqdThP-5yF_UvTkn0KzujYk0AFV5H00TZcq0KdpyfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfq1civrHnz0AqLUWYs0ZK45HcsP6KWThnqn164P1b)

Qt+Caffe+OpenCV——【一个基于VGG网络的人脸识别考勤系统】(http://blog.csdn.net/Mr_...

前言将PC端的一个基于深度学习的考勤系统的初版在这里发一发，会不断的更新。这一次的代码比以往的有一个【基于深度





内容举报



返回顶部





学习的人脸识别系统】要更加简洁完善，并且外带了GUI，可以移植到USB摄像头，正在开发连接...

 Mr_Curry (http://blog.csdn.net/Mr_Curry) 2016年12月19日 21:41  3268



Ubuntu Qt下pro配置caffe和opencv出现的问题及解决方法 (<http://blog.csdn.net/wangs1167...>)

Ubuntu Qt下pro配置caffe和opencv出现的问题及解决方法

 wangs11678 (<http://blog.csdn.net/wangs11678>) 2017年11月15日 10:37  41

Qt在Linux环境下如何进行动态链接(i.e. Caffe+Qt) (http://blog.csdn.net/hyichao_csdn/articl...)

利用Qt进行c++的GUI开发，在我看来是所有GUI方案中，最有吸引力的一个。一方面，Qt的API封装的特别好，很接近其他“先进”的UI框架，就算是进行大规模的程序开发，Qt也能够胜任。 另一方...

 hyichao_csdn (http://blog.csdn.net/hyichao_csdn) 2015年11月12日 16:40  2602

AI 工程师职业指南





我们请来商汤、杜邦、声智、希为、58同城、爱因互动、中科视拓、鲁朗软件等公司 AI 技术一线的专家，请他们从实践的角度来解析 AI 领域各技术岗位的合格工程师都是怎样炼成的。


(http://www.baidu.com/cb.php?c=lgF_pyfqhHmknjzrj00IZ0qnfK9ujYzP1f4Pjnd0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1Y1mvNBrij03uWlWnANBnH0s0AwY5HDdnHc3rj0LPW60lgF_5y9YIZ0IQzqMpgwB1rA7Wuj0YmhP9PARvujmYmH0vm1qdlAdxTVqdThP-5HDknWF9mhkEusKzujYk0AFV5H00TZcq0KdpyfqhHRLPjnvnfKEpyfqhHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK...)

CAFFE-QT编译调试 (<http://blog.csdn.net/u012968002/article/details/51462545>)

作者：肖良 链接：<http://www.zhihu.com/question/27987666/answer/80728225> 来源：知乎 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注...



 u012968002 (<http://blog.csdn.net/u012968002>) 2016年05月20日 16:15  1835

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](#)



基于opencv和QT的人脸（人眼）检测程序 (<http://blog.csdn.net/zyx1990412/article/details/...>)

本篇博客是在作者的上一篇博客《基于QT和opencv的摄像头（本地图片）读取并输出程序》的基础上进行开发的，利用opencv自带的分类器进行人脸或者人眼的识别。 在上一篇博客里，我们...

 zyx1990412 (<http://blog.csdn.net/zyx1990412>) 2016年04月21日 10:49  2415

Qt5.7+Opencv2.4.9人脸识别(一)项目介绍 (<http://blog.csdn.net/qg78442761/article/details/7...>)

Qt5.7+Opencv2.4.9人脸识别(一)项目介绍

 qg78442761 (<http://blog.csdn.net/qg78442761>) 2017年05月04日 09:58  7456



uhF-



内容举报



返回顶部



2



QT5+OpenCV人脸识别 (http://blog.csdn.net/CSDN_SXL/article/details/50601168)

本文介绍使用OpenCV扩展包里的人脸识别

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

2017年04月01日 14:19

9098

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

ZIP

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

2003年04月30日 00:00

676KB

下载

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

ZIP

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

2003年04月30日 00:00

676KB

下载

基于VGG-Face的人脸识别测试 (http://blog.csdn.net/u013078356/article/details/60955197)

VGG Face Descriptor 是牛津大学VGG小组的工作，现在已经开源训练好的网络结构和模型参数，本文将基于此模型在caffe上使用自己的人脸数据微调，并进行特征提取与精确度验证。数据传...

u013078356 (http://blog.csdn.net/u013078356)

2017年03月09日 10:02

2037

深度学习与人脸识别系列（6）__利用训练好的vgg模型进行人脸识别(利用摄像头) (http://blog....

作者： wjmishuai 出处： http://blog.csdn.net/wjmishuai/article/details/50854178 声明: 版权所有, 转载请注明出处 一: 人脸识别系统...

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

2016年03月11日 10:50

8694

深度学习与人脸识别系列（3）__基于VGGNet的人脸识别系统 (http://blog.csdn.net/wjmishu...

作者： wjmishuai 出处： 声明： 版权所有，转载请联系作者并注明出处 1. 引言 本文中介绍的人脸识别系统是基于这两篇论文： 《Very deep convolutional netwo...

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

2016年03月11日 10:48

20284

Deep Learning学习 - VGG-Face网络人脸识别 (http://blog.csdn.net/jinglingli_SJTU/article/...

记录一下使用VGGNet进行人脸识别的实验过程。数据集：训练集 9W+张人脸图片，包含10000 ID。 1. 数据集准备 将数据集图片分为训练集以及测试集两个部分，并生成标签文件，记录在.txt...

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

2016年12月02日 09:40

3037

深度学习与人脸识别系列（5）__vgg人脸识别模型测试 (http://blog.csdn.net/wjmishuai/artic...

作者： wjmishuai 出处： http://blog.csdn.net/wjmishuai/article/details/50854168 声明: 版权所有, 转载请注明出处 代码下载地址:...

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

2016年03月11日 10:50

6968

👍

2

☰

🔖

💬

🔗

内容举报

🔼

返回顶部

⚠️

内容举报

🔼

返回顶部

http://blog.csdn.net/Mr_Curry/article/details/53859654

11/12



深度学习与人脸识别系列（4）__vgg人脸识别模型测试 (http://blog.csdn.net/zchang81/articl...

vgg人脸识别模型测试

 zchang81 (http://blog.csdn.net/zchang81) 2017年05月05日 12:55 586

QT5+OpenCV人脸、人眼检测 (http://blog.csdn.net/CSDN_SXL/article/details/50600152)

一：在QLabel上显示Mat图像 void showMat(Mat mat) { Mat rgb; QImage img; if(mat.channels() == 3) ...

 CSDN_SXL (http://blog.csdn.net/CSDN_SXL) 2016年01月28日 13:48 8090

 ZIP

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](#)



 内容举报

 返回顶部