

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



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

- 10
-
-
-

OpenCV实现车牌识别，OCR分割，ANN神经网络

原创 2016年12月08日 22:47:02

标签：opencv (http://so.csdn.net/so/search/s.do?q=opencv&t=blog) /
机器学习 (http://so.csdn.net/so/search/s.do?q=机器学习&t=blog) /
神经网络 (http://so.csdn.net/so/search/s.do?q=神经网络&t=blog) /
车牌识别 (http://so.csdn.net/so/search/s.do?q=车牌识别&t=blog)

6313

主要步骤：
准备车牌单个字符图像作为神经网络分类器的训练数据，越多越好。当然需要对每幅图像提取特征，这里使用的是水平和垂直累计直方图和缩小后的图像信息。
获取车牌图像，这里的车牌图像已经完成抠图，并且是灰度图像。
将车牌图像中每个字符分割成单一图像（OCR类实现）。
提取分割出的字符图像特征信息，并使用分类识别字符（OCR类实现）。
程序运行过程：



原始带有车牌的图片



抠图并输入的车牌图片



二值化并分割成单个字符图片



程序运行结果

代码：

```
[cpp]
1. #ifndef Plate_h
2. #define Plate_h
3.
```

- 10
-



离婚后房屋分割



+ 关注

(http://blog.csdn.net/xukaiwen_2016)
码云

原创 19 粉丝 102 喜欢 0 未开通 (https://github.com/xukaiwen_2016)

他的最新文章
更多文章 (http://blog.csdn.net/xukaiwen_2016)

tensorflow图片预处理，随机亮度，旋转，剪切，翻转。(http://blog.csdn.net/xukaiwen_2016/article/details/77571415)

卷积神经网络CNN原理以及TensorFlow实现 (http://blog.csdn.net/xukaiwen_2016/article/details/70880694)

TensorFlow实现去噪自编码器及使用—Masking Noise Auto Encoder (http://blog.csdn.net/xukaiwen_2016/article/details/70832868)

相关推荐

opencv车牌识别（超详细注释）(http://blog.csdn.net/ding977921830/article/details/43564003)

用opencv3写的超详细注释的车牌检测 (http://blog.csdn.net/Caesar6666/article/details/73043328)



离婚后房屋分割



立即体验

内容举报

返回顶部



10



```
3.
4. #include <string.h>
5. #include <vector>
6.
7. #include <cv.h>
8. #include <highgui.h>
9. #include <cvaux.h>
10.
11. using namespace std;
12. using namespace cv;
13.
14. //车牌类
15. class Plate{
16. public:
17.     Plate();
18.     Plate(Mat img, Rect pos);
19.     string str();
20.     Rect position;//当前车牌在大图的位置，为了把识别出的车牌号显示到原图的车牌位置处
21.     Mat plateImg;//车牌图像，必须是灰度图像
22.     vector<char> chars;
23.     vector<Rect> charsPos;
24. };
25.
26. #endif

[cpp]
1. #include "stdafx.h"
2. #include "Plate.h"
3.
4. Plate::Plate(){
5. }
6.
7. Plate::Plate(Mat img, Rect pos){
8.     plateImg=img;
9.     position=pos;
10. }
11. //将车牌号码按照间隔长短拼接成字符串
12.
13. string Plate::str(){
14.     string result="";
15.     //Order numbers
16.     vector<int> orderIndex;
17.     vector<int> xpositions;
18.     for(int i=0; i< charsPos.size(); i++){
19.         orderIndex.push_back(i);
20.         xpositions.push_back(charsPos[i].x);
21.     }
22.     float min=xpositions[0];
23.     int minIdx=0;
24.     for(int i=0; i< xpositions.size(); i++){
25.         min=xpositions[i];
26.         minIdx=i;
27.         for(int j=1; j<xpositions.size(); j++){
28.             if(xpositions[j]<min){
29.                 min=xpositions[j];
30.                 minIdx=j;
31.             }
32.         }
33.         int aux_i=orderIndex[i];
34.         int aux_min=orderIndex[minIdx];
35.         orderIndex[i]=aux_min;
36.         orderIndex[minIdx]=aux_i;
37.
38.         float aux_xi=xpositions[i];
39.         float aux_xmin=xpositions[minIdx];
40.         xpositions[i]=aux_xmin;
41.         xpositions[minIdx]=aux_xi;
42.     }
43. }
```



在线课程



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnj0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnj0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnj0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnj0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnj0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnj0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnj0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnj0iZ0qnfK9ujYzP1mznWR10Aw-



离婚后房屋分割



/blog.csdn.net/xukaiwen_2016/article/details/52318476)

4825

WINDOWS.H already included.MFC app
s must not #include <windows.h> (http://b
log.csdn.net/xukaiwen_2016/article/detail
s/52088816)

4584

基于OpenCV实现二值图细化，骨骼化并
求出端点和交叉点 (http://blog.csdn.net/xu
kaiwen_2016/article/details/53135866)

4574

```
42.     for(int i=0; i<orderIndex.size(); i++){
43.         result=result+chars[orderIndex[i]];
44.     }
45.     return result;
46. }
```

车牌类代码

```
[cpp]
1.  #ifndef OCR_h
2.  #define OCR_h
3.
4.  #include <string.h>
5.  #include <vector>
6.
7.  #include "Plate.h"
8.
9.  #include <cv.h>
10. #include <highgui.h>
11. #include <cvaux.h>
12. #include <ml.h>
13.
14. using namespace std;
15. using namespace cv;
16.
17.
18. #define HORIZONTAL 1
19. #define VERTICAL 0
20.
21. class CharSegment{
22. public:
23.     CharSegment();
24.     CharSegment(Mat i, Rect p);
25.     Mat img;
26.     Rect pos;
27. };
28.
29. class OCR{
30. public:
31.     bool DEBUG;
32.     bool saveSegments;
33.     string filename;
34.     static const int numCharacters;//字符个数
35.     static const char strCharacters[];//字符数组
36.     OCR(string trainFile);
37.     OCR();
38.     string run(Plate *input);//识别车牌
39.     int charSize;
40.     Mat preprocessChar(Mat in);//将字符图片调整为正方形
41.     int classify(Mat f);//根据特征识别出每个字符图片的字符
42.     void train(Mat trainData, Mat trainClasses, int nlayers);//训练分类器
43.     int classifyKnn(Mat f);//扩展的Knn分类器
44.     void trainKnn(Mat trainSamples, Mat trainClasses, int k);
45.     Mat features(Mat input, int size);//提取每幅字符图片的特征
46.
47. private:
48.     bool trained;
49.     vector<CharSegment> segment(Plate input);//分割车片图片
50.     Mat Preprocess(Mat in, int newSize);//缩放为正方形
51.     Mat getVisualHistogram(Mat *hist, int type);//生成视觉直方图
52.     void drawVisualFeatures(Mat character, Mat hhist, Mat vhist, Mat lowData);//绘制视觉直
    方图
53.     Mat ProjectedHistogram(Mat img, int t);//计算累计直方图
54.     bool verifySizes(Mat r);//判断字符图像大小是否合适
55.     CvANN_MLP ann;//神经网络分类器
56.     CvKNearest knnClassifier;//扩展的k邻域分类器
57.     int K;
```



```

58. };
59.
60. #endif

[cpp]
1. #include "stdafx.h"
2. #include "OCR.h"
3.
4. const char OCR::strCharacters[] = {'0','1','2','3','4','5','6','7','8','9','B', 'C', 'D', 'F',
5. const int OCR::numCharacters=30;
6.
7. CharSegment::CharSegment(){}
8. CharSegment::CharSegment(Mat i, Rect p){
9.     img=i;
10.     pos=p;
11. }
12.
13. OCR::OCR(){
14.     DEBUG=false;
15.     trained=false;
16.     saveSegments=false;
17.     charSize=20;
18. }
19. OCR::OCR(string trainFile){
20.     DEBUG=false;
21.     trained=false;
22.     saveSegments=false;
23.     charSize=20;
24.
25.     //Read file storage.
26.     FileStorage fs;
27.     fs.open("OCR.xml", FileStorage::READ);
28.     Mat TrainingData;
29.     Mat Classes;
30.     fs["TrainingDataF15"] >> TrainingData;
31.     fs["classes"] >> Classes;
32.
33.     train(TrainingData, Classes, 10);
34.
35. }
36. //将单个字符图像变成正方形
37. Mat OCR::preprocessChar(Mat in){
38.     int h=in.rows;
39.     int w=in.cols;
40.     Mat transformMat=Mat::eye(2,3,CV_32F); //缩放矩阵
41.     int m=max(w,h);
42.     transformMat.at<float>(0,2)=m/2 - w/2;
43.     transformMat.at<float>(1,2)=m/2 - h/2;
44.
45.     Mat warpImage(m,m, in.type());
46.     warpAffine(in, warpImage, transformMat, warpImage.size(), INTER_LINEAR, BORDER_CONSTANT, Sc
47.
48.     Mat out;
49.     resize(warpImage, out, Size(charSize, charSize) );
50.
51.     return out;
52. }
53.
54. //判断字符图像长宽是否符合要求
55. bool OCR::verifySizes(Mat r){
56.     //Char sizes 45x77
57.     float aspect=45.0f/77.0f;
58.     float charAspect= (float)r.cols/(float)r.rows;
59.     float error=0.35;
60.     float minHeight=15;
61.     float maxHeight=28;
62.     //We have a different aspect ratio for number 1, and it can be ~0.2
63.     //

```

返回顶部



内容举报

返回顶部





10



10



```

63.         float minAspect=0.2;
64.         float maxAspect=aspect+aspect*error;
65.         //area of pixels
66.         float area=countNonZero(r);
67.         //bb area
68.         float bbArea=r.cols*r.rows;
69.         //%% of pixel in area
70.         float percPixels=area/bbArea;
71.
72.         if(DEBUG)
73.             cout << "Aspect: "<< aspect << " ["
74.             << minAspect << ", " << maxAspect << "]" << "Area "
75.             << percPixels << " Char aspect " << charAspect << " Height char "<< r.rows << "\n";
76.         if(percPixels < 0.8 && charAspect > minAspect && charAspect < maxAspect && r.rows >= minHei
77.             return true;
78.         else
79.             return false;
80.     }
81.
82. //将车牌图像进一步分割成单个字符图片
83. vector<CharSegment> OCR::segment(Plate plate){
84.     Mat input=plate.plateImg;
85.     vector<CharSegment> output;
86.     //将输入图像二值化
87.     Mat img_threshold;
88.     threshold(input, img_threshold, 60, 255, CV_THRESH_BINARY_INV);
89.     if(DEBUG)
90.         imshow("Threshold plate", img_threshold);
91.     Mat img_contours;
92.     img_threshold.copyTo(img_contours);
93.     //查找字符轮廓
94.     vector< vector< Point> > contours;
95.     findContours(img_contours,
96.                 contours, // 轮廓
97.                 CV_RETR_EXTERNAL, //去除内环
98.                 CV_CHAIN_APPROX_NONE); // 轮廓所有像素
99.
100. //将轮廓绘制到车牌图
101. cv::Mat result;
102. img_threshold.copyTo(result);
103. cvtColor(result, result, CV_GRAY2RGB);
104. cv::drawContours(result, contours, -1, cv::Scalar(255,0,0),1);
105.
106. vector<vector<Point> >::iterator itc= contours.begin();
107.
108. //筛选符合条件的闭环
109. while (itc!=contours.end()) {
110.     //创建一个包围矩形
111.     Rect mr= boundingRect(Mat(*itc));
112.     rectangle(result, mr, Scalar(0,255,0));
113.
114.     Mat auxRoi(img_threshold, mr);
115.     if(verifySizes(auxRoi)){//判断长宽是否满足
116.         auxRoi=preprocessChar(auxRoi);//缩放成正方形
117.         output.push_back(CharSegment(auxRoi, mr)); //保存字符图像及位置
118.         rectangle(result, mr, Scalar(0,125,255));
119.     }
120.     ++itc;
121. }
122. if(DEBUG)
123.     cout << "Num chars: " << output.size() << "\n";
124. if(DEBUG)
125.     imshow("SEgmented Chars", result);
126. return output;
127. }
128. //计算累计直方图，统计每列或行的非0像素个数

```



内容举报

返回顶部



```

129. Mat OCR::ProjectedHistogram(Mat img, int t)
130. {
131.     int sz=(t)?img.rows:img.cols;
132.     Mat mhist=Mat::zeros(1,sz,CV_32F);
133.
134.     for(int j=0; j<sz; j++){
135.         Mat data=(t)?img.row(j):img.col(j);
136.         mhist.at<float>(j)=countNonZero(data);//
137.     }
138.
139.     //直方图归1化
140.     double min, max;
141.     minMaxLoc(mhist, &min, &max);
142.
143.     if(max>0)
144.         mhist.convertTo(mhist, -1, 1.0f/max, 0);
145.
146.     return mhist;
147. }
148.
149. //得到直方图图像
150. Mat OCR::getVisualHistogram(Mat *hist, int type)
151. {
152.
153.     int size=100;
154.     Mat imHist;
155.
156.
157.     if(type==HORIZONTAL){
158.         imHist.create(Size(size,hist->cols), CV_8UC3);
159.     }else{
160.         imHist.create(Size(hist->cols, size), CV_8UC3);
161.     }
162.
163.     imHist=Scalar(55,55,55);
164.
165.     for(int i=0;i<hist->cols;i++){
166.         float value=hist->at<float>(i);
167.         int maxval=(int)(value*size);
168.
169.         Point pt1;
170.         Point pt2, pt3, pt4;
171.
172.         if(type==HORIZONTAL){
173.             pt1.x=pt3.x=0;
174.             pt2.x=pt4.x=maxval;
175.             pt1.y=pt2.y=i;
176.             pt3.y=pt4.y=i+1;
177.
178.             line(imHist, pt1, pt2, CV_RGB(220,220,220),1,8,0);
179.             line(imHist, pt3, pt4, CV_RGB(34,34,34),1,8,0);
180.
181.             pt3.y=pt4.y=i+2;
182.             line(imHist, pt3, pt4, CV_RGB(44,44,44),1,8,0);
183.             pt3.y=pt4.y=i+3;
184.             line(imHist, pt3, pt4, CV_RGB(50,50,50),1,8,0);
185.         }else{
186.
187.             pt1.x=pt2.x=i;
188.             pt3.x=pt4.x=i+1;
189.             pt1.y=pt3.y=100;
190.             pt2.y=pt4.y=100-maxval;
191.
192.
193.             line(imHist, pt1, pt2, CV_RGB(220,220,220),1,8,0);
194.             line(imHist, pt3, pt4, CV_RGB(34,34,34),1,8,0);
195.
196.             pt3.x=pt4.x=i+2;
197.             line(imHist, pt3, pt4, CV_RGB(44,44,44),1,8,0);

```

内容举报

返回顶部



```
197.         line(img, pt3, pt4, CV_RGB(255,0,0),1,8,0);
198.         pt3.x=pt4.x+1;
199.         line(imHist, pt3, pt4, CV_RGB(50,50,50),1,8,0);
200.     }
201. }
202. return imHist ;
203. }
204.
205. void OCR::drawVisualFeatures(Mat character, Mat hhist, Mat vhist, Mat lowData){
206.     Mat img(121, 121, CV_8UC3, Scalar(0,0,0));
207.     Mat ch;
208.     Mat ld;
209.
210.     cvtColor(character, ch, CV_GRAY2RGB);
211.
212.     resize(lowData, ld, Size(100, 100), 0, 0, INTER_NEAREST );
213.     cvtColor(ld,ld,CV_GRAY2RGB);
214.
215.     Mat hh=getVisualHistogram(&hhist, HORIZONTAL);
216.     Mat hv=getVisualHistogram(&vhist, VERTICAL);
217.
218.     Mat subImg=img(Rect(0,101,20,20));
219.     ch.copyTo(subImg);
220.
221.     subImg=img(Rect(21,101,100,20));
222.     hh.copyTo(subImg);
223.
224.     subImg=img(Rect(0,0,20,100));
225.     hv.copyTo(subImg);
226.
227.     subImg=img(Rect(21,0,100,100));
228.     ld.copyTo(subImg);
229.
230.     line(img, Point(0,100), Point(121,100), Scalar(0,0,255));
231.     line(img, Point(20,0), Point(20,121), Scalar(0,0,255));
232.
233.     imshow("Visual Features", img);
234.
235.     cvWaitKey(0);
236. }
237.
238. Mat OCR::features(Mat in, int sizeData){
239.     //分别获取垂直和水平直方图信息
240.     Mat vhist=ProjectedHistogram(in,VERTICAL);
241.     Mat hhist=ProjectedHistogram(in,HORIZONTAL);
242.
243.     //低分辨率图像
244.     Mat lowData;
245.     resize(in, lowData, Size(sizeData, sizeData) );//15x15
246.
247.     if(DEBUG)
248.         drawVisualFeatures(in, hhist, vhist, lowData);
249.
250.     //整合低分辨率图像信息和直方图统计信息，
251.     int numCols=vhist.cols+hhist.cols+lowData.cols*lowData.cols;
252.
253.     Mat out=Mat::zeros(1,numCols,CV_32F);
254.     //保存特征信息
255.     int j=0;
256.     for(int i=0; i<vhist.cols; i++)
257.     {
258.         out.at<float>(j)=vhist.at<float>(i);
259.         j++;
260.     }
261.     for(int i=0; i<hhist.cols; i++)
262.     {
263.         out.at<float>(j)=hhist.at<float>(i);
264.         j++;
265.     }
266. }
```



内容举报

返回顶部

内容举报

返回顶部

```

266.     for(int x=0; x<lowData.cols; x++)
267.     {
268.         for(int y=0; y<lowData.rows; y++){
269.             out.at<float>(j)=(float)lowData.at<unsigned char>(x,y);
270.             j++;
271.         }
272.     }
273.     if(DEBUG)
274.         cout << out << "\n=====\\n";
275.     return out;
276. }
277.
278. //训练          //训练样本数据//每条数据对应的字母下标//深度
279. void OCR::train(Mat TrainData, Mat classes, int nlayers){
280.     Mat layers(1,3,CV_32SC1);
281.     layers.at<int>(0)= TrainData.cols;//每个样本宽度
282.     layers.at<int>(1)= nlayers;//深度
283.     layers.at<int>(2)= numCharacters;//结果个数
284.     ann.create(layers, CvANN_MLP::SIGMOID_SYM, 1, 1);
285.
286.     Mat trainClasses;
287.     trainClasses.create( TrainData.rows, numCharacters, CV_32FC1 );//每一条样本都对对应着
numCharacters个可能结果，但是只有一个结果是正确的，
288.     for( int i = 0; i < trainClasses.rows; i++ )
289.     {
290.         for( int k = 0; k < trainClasses.cols; k++ )
291.         {
292.             //将该条训练数据对应的字符下标位置赋值为1，其他赋值为0
293.             if( k == classes.at<int>(i) )
294.                 trainClasses.at<float>(i,k) = 1;
295.             else
296.                 trainClasses.at<float>(i,k) = 0;
297.         }
298.     }
299.     Mat weights( 1, TrainData.rows, CV_32FC1, Scalar::all(1) );
300.     //开始训练学习
301.     ann.train( TrainData, trainClasses, weights );
302.     trained=true;
303. }
304. //识别字符
305. int OCR::classify(Mat f){
306.     int result=-1;
307.     Mat output(1, numCharacters, CV_32FC1);
308.     ann.predict(f, output);
309.     Point maxLoc;
310.     double maxVal;
311.     minMaxLoc(output, 0, &maxVal, 0, &maxLoc);//求最大值以及下标位置，这里没有打印出来最大值
312.     return maxLoc.x;
313. }
314.
315. int OCR::classifyKnn(Mat f){
316.     int response = (int)knnClassifier.find_nearest( f, K );
317.
318.     return response;
319. }
320. void OCR::trainKnn(Mat trainSamples, Mat trainClasses, int k){
321.     K=k;
322.     // learn classifier
323.     knnClassifier.train( trainSamples, trainClasses, Mat(), false, K );
324. }
325. string OCR::run(Plate *input){
326.
327.     //分割车牌中每个字符
328.     vector<CharSegment> segments=segment(*input);
329.
330.     for(int i=0; i<segments.size(); i++){
331.         //分割每个字符

```



内容举报

返回顶部





10



```
331.         //统一所有子图像大小
332.         Mat ch=preprocessChar(segments[i].img);
333.         if(saveSegments){
334.             stringstream ss(stringstream::in | stringstream::out);
335.             ss << "tmpChars/" << filename << "_" << i << ".jpg";
336.             imwrite(ss.str(),ch);
337.         }
338.         //提取每个字符图像特征
339.         Mat f=features(ch,15);
340.         //For each segment feature Classify
341.         int character=classify(f);
342.         input->chars.push_back(strCharacters[character]);
343.         input->charsPos.push_back(segments[i].pos);
344.     }
345.     return "-";//input->str();
346. }
```

OCR类代码

```
[cpp]
1. #include "stdafx.h"
2. #include <cv.h>
3. #include <highgui.h>
4. #include <cvaux.h>
5. #include <ml.h>
6.
7. #include <iostream>
8. #include <vector>
9.
10. #include "DetectRegions.h"
11. #include "OCR.h"
12.
13. using namespace std;
14. using namespace cv;
15.
16. string getFilename(string s) {
17.
18.     char sep = '/';
19.     char sepExt = '.';
20.
21. #ifdef _WIN32
22.     sep = '\\';
23. #endif
24.
25.     size_t i = s.rfind(sep, s.length());
26.     if (i != string::npos) {
27.         string fn = (s.substr(i + 1, s.length() - i));
28.         size_t j = fn.rfind(sepExt, fn.length());
29.         if (i != string::npos) {
30.             return fn.substr(0, j);
31.         }
32.         else{
33.             return fn;
34.         }
35.     }
36.     else{
37.         return "";
38.     }
39. }
40.
41. int main(int argc, char** argv)
42. {
43.     char* filename;
44.     Mat input_image;//必须为灰度图像
45.
46.     //有输入图片才继续
47.     if (argc > 1) {
```



内容举报

返回顶部




```
47.     if (argc >= 2)
48.     {
49.         filename = argv[1];
50.         input_image = imread(filename, 1);
51.     }
52.     else{
53.         printf("Use:\n\t%s image\n", argv[0]);
54.         return 0;
55.     }
56.
57.     string filename_withoutExt = getFilename(filename);//得到去除后缀部分
58.     OCR ocr("OCR.xml");//参数为保存了自己训练数据的xml文件
59.     ocr.saveSegments = true;
60.     ocr.DEBUG = true;
61.     ocr.filename = filename_withoutExt;
62.     Plate plate;
63.     plate.plateImg = input_image;
64.     plate.position = Rect(50, 100, input_image.cols, input_image.rows);//车牌是从大图中抠图出来的，这里说明车牌的位置和大小
65.     imwrite("plateImg.jpg", plate.plateImg);
66.     string plateNumber = ocr.run(&plate);
67.     string licensePlate = plate.str();
68.     cout << "=====\n";
69.
70.     cout << "License plate number: " << licensePlate << "\n";
71.     cout << "=====\n";
72.     rectangle(input_image, plate.position, Scalar(0, 0, 200));
73.     putText(input_image, licensePlate, Point(plate.position.x, plate.position.y), CV_FONT_HERSHEY_SIMPLEX, 1, Scalar(0, 0, 255));
74.     if (false){
75.         imshow("Plate Detected seg", plate.plateImg);
76.         cvWaitKey(0);
77.     }
78.     imshow("Plate Detected", input_image);
79.     for (;;)
80.     {
81.         int c;
82.         c = cvWaitKey(10);
83.         if ((char)c == 27)
84.             break;
85.     }
86.     return 0;
87. }
```

main函数代码

有关ANN神经网络分类器的原理及训练请参考我的另一篇文章：http://blog.csdn.net/xukaiwen_2016/article/details/53293465
最后：例子中没有实现对中文的识别，其实原理都是一样的，大家可以自己寻找中文车牌的图片进行分类器训练即可，代码几乎不用修改。
需要代码以及分类器训练数据xml文件的话，评论留下邮箱。

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)



 内容举报

 返回顶部

 内容举报

- 

qq_35986996 (/qq_35986996) 昨天 22:34
(/qq_35986996)@qq.com 谢谢楼主

145楼
- 回复
- 

moliang136 (/moliang136) 前天 17:12
(/moliang136)@qq.com 谢谢楼主

144楼
- 回复
- 

wrs000001 (/wrs000001) 5天前 11:22
(/wrs000001)sheng@163.com 谢谢楼主。

143楼
- 回复

查看 163 条热评

相关文章推荐

opencv车牌识别（超详细注释）(http://blog.csdn.net/ding977921830/article/details/43564...

/*****\ ...

ding977921830 (http://blog.csdn.net/ding977921830) 2015年02月06日 14:20 12592

用opencv3写的超详细注释的车牌检测 (http://blog.csdn.net/Caesar6666/article/details/7304...

#include<iostream> #include<opencv2/opencv.hpp> #include<opencv2/highgui/highgui.hpp> #include<openc...

Caesar6666 (http://blog.csdn.net/Caesar6666) 2017年06月11日 16:52 915



一个普通程序员的内心独白....躺枪！躺枪！

我，一个普通通程序员，没有过人的天赋，没有超乎寻常的好运，该如何逆袭走上人生巅峰？

(http://www.baidu.com/cb.php?c=lgF_pyfqhHmknjDLnjT0lZ0qnfK9ujYzP1nsrjD10Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1d-uAubPAPhPW9buHnzmWcY0AwY5HDdnHc3nWm3rHR0lgF_5y9YIZ0IQzq-uZR8mLPbUB48ugfEXyN9T-KzUvdEIA-EUBqbugw9pysEn1qdlAdxTvqdThP-5yF_UvTkn0KzujYk0AFV5H00TZcq0NKdpyfqhHRLPjnvnfKEpyfqhHc4rj6kP0KWpyfqP1cwrHnz0AqLUWys0ZK45HcsP6KWThnqn1TdPW0)

基于opencv的车牌识别解析与代码 (http://blog.csdn.net/Linoi/article/details/17761655)

车牌识别太出名了，我也就花几天来了解下这个系统，并结合opencv进行实现。下面是一些介绍：车辆牌照识别（License Plate Recognition，LPR）技术作为交通管理自动化的重要手段...

Linoi (http://blog.csdn.net/Linoi) 2014年01月02日 17:38 23143

《Mastering Opencv ...读书笔记系列》车牌识别（I）(http://blog.csdn.net/jinshengtao/arti...



离婚后房屋分割



返回顶部



内容举报



返回顶部



一、ANPR简介： Automatic Number Plate Recognition (ANPR), 是一种使用Optical Character Recognition (OCR)和其他分...
jinshengtao (<http://blog.csdn.net/jinshengtao>) 2014年01月05日 15:28 42152



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](#)



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_pyfqhHmknjfvP1m0lZ0qnfk9uj`YzP1f4Pjnz0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1dhPjPBuym3PvndPHRzP10v0AwY5HDdnHc3nWm3rHm0lgF_5y9YlZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq_lyd8Q1N9nHmvnj7hnHPWnjFhPAD1Pyn4uW99ujqdlAdxTvqdThP-5HDknWw9mhhkEusKzuYk0AFV5H00TZcqN0KdpyfqhHRLPjnvnfKEpyfqhHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWTThnqPHRYPHD)

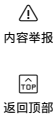
使用opencv的SVM和神经网络实现车牌识别 (<http://blog.csdn.net/AP1005834/article/details...>)
一、前言 本文参考自《深入理解Opencv 实用计算机视觉项目解析》中的自动车牌识别项目，并对其中的方法理解后，再进行实践。深刻认识到实际上要完成车牌区域准确定位、车牌区域中字符的准确分割，字符准确...
AP1005834 (<http://blog.csdn.net/AP1005834>) 2016年05月08日 00:08 8397

opencv实现车牌识别之字符分割 (<http://blog.csdn.net/u011630458/article/details/43733057>)
简介 在前一篇中，我们已经定位出来了在图片中车牌号的位置，并且将车牌号图片复制成了新图片，并显示出来，本章在这些被截取出来的图片上继续处理。 截取出来的新图片如下： ...
u011630458 (<http://blog.csdn.net/u011630458>) 2015年02月11日 13:34 16242

基于OpenCV的车牌识别系统之二 —— 字符分割与识别 (<http://blog.csdn.net/chengchaopeng...>)
基于OpenCV的车牌识别系统之二——字符分割与识别 车牌定位完之后就是车牌的字符识别，字符识别又分为字符分割和字符识别。字符分割的步骤采用OpenCV中寻找外轮廓函数，并根据字符轮廓在车牌图像上...
chengchaopeng520 (<http://blog.csdn.net/chengchaopeng520>) 2017年03月24日 16:23 1181

opencv实现车牌识别之车牌号定位_1 (<http://blog.csdn.net/u011630458/article/details/4360...>)
简介 按照在哪里跌倒就在哪里爬起来的精神，本章继续做车牌号的检测识别。所有步骤分为3步完成：车牌号定位，车牌号字符分割、字符识别。 本章为第一部分：车牌号定位。 效果演示 正式开始讲解之...
u011630458 (<http://blog.csdn.net/u011630458>) 2015年02月07日 17:06 6145

基于Opencv的汽车车牌识别 (<http://blog.csdn.net/u012739230/article/details/44340253>)
第一次写博客，希望大家多多指教。
u012739230 (<http://blog.csdn.net/u012739230>) 2015年03月17日 13:04 7917





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

不错的计算机视觉博客：http://blog.csdn.net/lee_cv/article/details/9180719 篇文章献给所有第一次听说车牌识别ANPR但需要短时间实现的苦...

u012556077 (http://blog.csdn.net/u012556077)

2015年09月17日 10:14

1703



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



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

《Mastering Opencv ...读书笔记系列》车牌识别（II） (http://blog.csdn.net/jinshengtao/art...

继上一篇文章后，现在要做的就是从车牌图像上使用optical character recognition算法将字符提取出来。对于每一块被检测的车牌，使用带监督的神经网络机器学习算法来识别字符。 本文内...

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

2014年01月07日 12:08

17904

从零使用OpenCV快速实现简单车牌识别系统 (http://blog.csdn.net/android_asp/article/detai...

这篇文章献给所有第一次听说车牌识别ANPR但需要短时间实现的苦逼同学们。 最近的小学期实训做的是个车牌识别系统，说实话真不知道学校怎么想的，虽然说图像处理也算的上是数字媒体很重要的一块分支了...

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

2013年06月26日 17:01

7155

Opencv中ANN神经网络使用示例 (http://blog.csdn.net/NNNNNNNNNNNNNY/article/details/5...

前一段儿想用opencv做一下数字识别，用神经网络做识别。 在网上搜了一下关于opencv中ANN的使用方法，@小魏的修行路的【模式识别】OpenCV中使用神经网络 CvANN_MLP 这篇文章写...

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

2016年03月24日 17:42

7545

【模式识别】OpenCV中使用神经网络 CvANN_MLP (http://blog.csdn.net/xiaowei_cqu/artic...

OpenCV的ml模块实现了人工神经网络（Artificial Neural Networks，ANN）最典型的多层感知器（multi-layer perceptrons，MLP）模型。由于ml模...

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

2013年06月05日 09:43

50600

opencv 学习之 SVM / ANN 图片分类（OPENCV3.2） (http://blog.csdn.net/u010477528/arti...



- 10
-
-
-

内容举报
返回顶部