特别说明

基于隐马尔可夫模型的人脸识别 C/C++源代码

中国视觉网网友　徐慧 提供

将生物特征识别应用于人脸，实际上是包含两个方面：第一，从图像或视频帧中检测人脸，即所谓的“人脸检测”（face detection）；第二，对检测到的人脸进行识别，判断这张脸是谁，即“人脸识别”（face recognition）。就实际应用而言，采用人脸做生物特征识别，其识别率、可靠性都无法与指纹、虹膜识别相提并论，但失为模式识别中的一个典型应用，至少可以起到抛砖引玉的作用。

下面的源代码采用隐马尔可夫模型（HMM）做人脸识别，它是 OPENCV 3.1 版本的一个应用示例程序，不再包含在 4.0 版本中。因此如果想编译源代码，则需要安装 OPENCV 3.1 版本。该版本可以从 SOURCEFORG 上下载。关于程序使用以及算法说明，参考下面的网页（英文）：

http://www.assuredigit.com/incoming/sourcecode/opencv/chinese_docs/appPage/FaceRecognition/FaceRecognition.htm

以及论文：

http://www.assuredigit.com/incoming/sourcecode/opencv/chinese_docs/papers/avbpa99.pdf

下载地址：

http://www.assuredigit.com/program/HMMDemo.rar
(转自阿须数码)

稿件 B: 人脸检测的 C/C++源代码

人脸检测的 C/C++源代码，曾发表于 OPENCV 的 MAILING LIST，主要是对 OPENCV 3.1 版本发布的代码做了一些速度上的优化，并且解决了内存泄漏的问题。这个程序所使用的 Paul Viola 提出（该论文“Rapid Object Detection using a Boosted Cascade of Simple Features”发表在 CVPR'01）的 Ada Boosted Cascade 算法可以说是目前最好最快的目标检测算法。

关于 OPENCV 的介绍，参考：

http://blog.csdn.net/hunnish/archive/2004/09/13/102535.aspx

关于该算法的详细介绍，也可参考：

http://www.merl.com/people/viola/research/publications/CVPR-2001.pdf

以及：

http://www.assuredigit.com/forum/display_topic_threads.asp?ForumID=11&TopicID=325

http://www.assuredigit.com/forum/display_topic_threads.asp?ForumID=11&TopicID=463

运行文件下载：

http://www.assuredigit.com/product_tech/Demo_Download_files/Face.exe

该程序可以对静止图像以及视频序列进行 face tracking。对视频序列，请先插入 USB 接口的摄像头。

====

在 OPENCV 3.1 版本，VC6.0 下编译通过

====

```
===
#ifdef _CH_
#define WIN32
#error "The file needs cvaux, which is not wrapped yet. Sorry"
#endif

#ifndef _EiC
#include "cv.h"
#include "cvaux.h"
#include "highgui.h"

#endif

#ifdef _EiC
#define WIN32
#endif

#define ORIG_WIN_SIZE 24
static CvMemStorage* storage = 0;
static CvHidHaarClassifierCascade* hid_cascade = 0;

#define WINNAME "Result"

void detect_and_draw( IplImage* image, IplImage* TempImage );

int main( int argc, char** argv )
```

```
{
CvCapture* capture = 0;

CvHaarClassifierCascade* cascade =
cvLoadHaarClassifierCascade( "",
cvSize( ORIG_WIN_SIZE, ORIG_WIN_SIZE ));
hid_cascade = cvCreateHidHaarClassifierCascade( cascade, 0, 0, 0, 1 );
cvReleaseHaarClassifierCascade( &cascade );

cvNamedWindow( WINNAME, 1 );
storage = cvCreateMemStorage(0);

if( argc == 1 || (argc == 2 && strlen(argv[1]) == 1 && isdigit(argv[1][0])))
capture = cvCaptureFromCAM( argc == 2 ? argv[1][0] - '0' : 0 );
else if( argc == 2 )
capture = cvCaptureFromAVI( argv[1] );

if( capture )
{
IplImage *frame, *temp;
cvGrabFrame( capture );
frame = cvRetrieveFrame( capture );

temp = cvCreateImage( cvSize(frame->width/2,frame->height/2), 8, 3 );

for(;;)
{
if( !cvGrabFrame( capture ))
break;
frame = cvRetrieveFrame( capture );
if( !frame )
break;

detect_and_draw( frame, temp );

if( cvWaitKey( 10 ) >= 0 )
{
//cvReleaseImage( &frame );
//cvReleaseImage( &temp );
cvReleaseCapture( &capture );
cvDestroyWindow(WINNAME);
return 0;
}
}
```

```
}
else
{
char* filename = argc == 2 ? argv[1] : (char*)"lena.jpg";
IplImage* image = cvLoadImage( filename, 1 );
IplImage* temp = cvCreateImage( cvSize(image->width/2,image->height/2), 8, 3 );

if( image )
{
cvFlip( image, image, 0 );
image->origin = IPL_ORIGIN_BL;
detect_and_draw( image, temp );
cvWaitKey(0);
cvReleaseImage( &image );
cvReleaseImage( &temp );
}
cvDestroyWindow(WINNAME);
return 0;
}
return 0;
}

void detect_and_draw( IplImage* img, IplImage* temp )
{
int scale = 2;
CvPoint pt1, pt2;
int i;

cvPyrDown( img, temp, CV_GAUSSIAN_5x5 );
#ifdef WIN32
cvFlip( temp, temp, 0 );
#endif
cvClearMemStorage( storage );

if( hid_cascade )
{
CvSeq* faces = cvHaarDetectObjects( temp, hid_cascade, storage,
1.2, 2, CV_HAAR_DO_CANNY_PRUNING );
for( i = 0; i < (faces ? faces->total : 0); i++ )
{
CvRect* r = (CvRect*)cvGetSeqElem( faces, i, 0 );
pt1.x = r->x*scale;
pt2.x = (r->x+r->width)*scale;
#ifdef WIN32
```

```
pt1.y = img->height - r->y*scale;
pt2.y = img->height - (r->y+r->height)*scale;
#else
pt1.y = r->y*scale;
pt2.y = (r->y+r->height)*scale;
#endif
cvRectangle( img, pt1, pt2, CV_RGB(255,255,0), 3 );
}
}

cvShowImage(WINNAME, img );
//cvReleaseImage( &temp );
(转自阿须数码)
}

#ifdef _EiC
main(1,"facedetect.c");
#endif
```