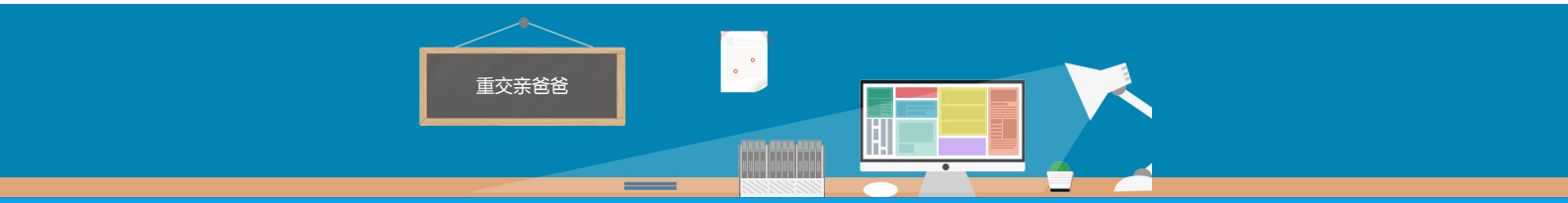


基于opencv3.0下的人脸检测和检测部分的高斯模糊处理

如题

这里将任务分解为三大部分：

1.录播放视频

2.人脸检测

3.部分高斯模糊

其中重点放在人脸检测和部分高斯模糊上

1.录播放视频（以opencv中的VideoCapture类进行实现）

首先罗列下操作环境：win10+vs2013+opencv3.0+单摄像头

opencv中提供了**VideoCapture**和**CvCapture**对视频进行操作

其中官方给出**CvCapture**的API为

C: CvCapture* cvCaptureFromCAM(int **device**)

C: CvCapture* cvCaptureFromFile(const char* **filename**)

Parameters:
- **filename** – name of the opened video file (eg. video.avi) or image sequence (eg. img_%02d.jpg, which will read samples like img_00.jpg, img_01.jpg, img_02.jpg, ...)
- **device** – id of the opened video capturing device (i.e. a camera index). If there is a single camera connected, just pass 0.

实例化CvCapture对象的时候，需要调用cvCaptureFromCAM(int device)进行实例化。

博主的是单摄像头没有外接摄像头，根据API进行填写操作

```
#include<opencv2\highgui\highgui.hpp>
#include<opencv2\imgproc\imgproc.hpp>
#include<opencv2\core\core.hpp>

int main(int argc, const char** argv)
{
    CvCapture* capture;
    capture = cvCaptureFromCAM(0);
    IplImage* frame;
    namedWindow("video", 1);
    while (true)
    {
        frame = cvQueryFrame(capture);
        cvShowImage("video",frame);
        char c = cvWaitKey(0);
        if (c == 27)break;//捕获Esc
    }
    destroyWindow("video");
    cvReleaseCapture(&capture);
    return 0;
}
```

通过简单的进行捕获摄像头的图像，结果如下：

并没有捕获到摄像头的所采集的图像。查阅的相关资料，有个方法将cvCaptureFromCAM()中填入-1。很遗憾是同样的效果。感兴趣的童鞋可以研究下，提出解决方案，共同促进学习。

所以这里采用**VideoCapture**进行采集视频

首先定义一个视频采集器，一个视频写入器；

VideoCapture capture;
VideoWriter writer;

初始化

```
/*****************************************************
函数名称: MyClass
函数功能: 初始化
传入参数:
返 回 值:
建立时间: 2018-05-13
修改时间:
建 立 人:
修 改 人:
其它说明:
*****************************************************/
MyClass::MyClass()
{
    if (!cascade.load(cascadeName))
    {
        cerr << "ERROR: Could not load classifier cascade" << endl;
        exit(1);
    }
    capture = 0;
    string filepath = "test.avi";
    //获得帧的宽高
    int w = static_cast<int>(capture.get(CV_CAP_PROP_FRAME_WIDTH));
    int h = static_cast<int>(capture.get(CV_CAP_PROP_FRAME_HEIGHT));
    Size S(w, h);
    //获得帧率
    double r = capture.get(CV_CAP_PROP_FPS);
    //打开视频文件，准备写入
    writer.open(filepath, -1, r, S, true);
    //打开失败
    if (!capture.isOpened())
    {
        exit(1);
    }
```

```
}
```

开始显示并录制：

```
/*************************************************
函数名称: play
函数功能: 开始播放
传入参数:
返 回 值:
建立时间: 2018-05-13
修改时间:
建 立 人:
修 改 人:
其它说明:
*************************************************/
void MyClass::play(){
    Mat image;
    namedWindow("直播页面按Esc退出", 1);
    while (true)
    {
        capture >> image;
        if (image.empty())break;
        imshow("直播页面按Esc退出", image);
        writer.write(image);
        char c = waitKey(30);
        if (c == 27)break;
    }
    cvDestroyWindow("直播页面按Esc退出");
}
/*************************************************
```

下面是播放录制好的视频，校验原来是否成功录制：

```
/*************************************************
函数名称: replay
函数功能: 播放测试
传入参数:
返 回 值:
建立时间: 2018-05-13
修改时间:
建 立 人:
修 改 人:
其它说明:
*************************************************/
void replay()
{
    char* path = "test.avi";
    IplImage* frame;
    CvCapture* capture=cvCreateFileCapture(path);
    namedWindow("播放界面按Esc退出", 1);
    while (true)
    {
        frame = cvQueryFrame(capture);
        cvShowImage("播放界面按Esc退出", frame);
        char c = waitKey(33);
        if (c == 27)break;
    }

}
```

结果：

播放界面按Esc退出

成功播放了^.^.

2.人脸识别（以opencv中CascadeClassifier类进行实现）

在opencv安装处"../opencv/sources/samples/cpp/facedetect.cpp"，里面有示例代码。（示例代码有图片和视频的识别，但是这种方式识别精准度不是很好。）

其中示例代码的视频处理部分主要的思想就是将视频中的每一帧提出来，然后根据CascadeClassifier类和haarcascade_frontalface_alt.xml所生成的对象，将图片进行类比区分。

ps：haarcascade_frontalface_alt.xml存储在"../opencv/sources/data/haarcascades"中。

将"../opencv/sources/data"包拷到项目中去，开始人脸识别。

```
/*************************************************
函数名称: detectAndDraw
函数功能: 在人脸上绘制圆圈
传入参数:
返 回 值:
建立时间: 2018-05-13
修改时间:
建 立 人:
修 改 人:
其它说明:
*************************************************/
void MyClass::detectAndDraw(Mat& img,char* title)
{
    Mat dog;
    dog = imread("F:\\Pictures\\gdog.png",1);
    Mat tempdog;
    double alpha = 1; double beta = 1 - alpha;

    double scale = 1;
    vector<Rect> faces;
    Mat gray,smallImg(cvRound(img.rows / scale), cvRound(img.cols / scale), CV_8UC1);;
    cvtColor(img, gray, COLOR_BGR2GRAY);
    resize(gray, smallImg, smallImg.size(), 0, 0, INTER_LINEAR);
    equalizeHist(smallImg, smallImg);
    cascade.detectMultiScale( smallImg, faces,1.1,2,0|CASCADE_SCALE_IMAGE,Size(30,30));//匹配人脸
    for (vector<Rect>::const_iterator r = faces.begin(); r != faces.end(); r++){
        Scalar color = colors[6];//红色
        int radius;
        Point center;
        double aspect_ratio = (double)r->width / r->height;
        if (0.75 < aspect_ratio && aspect_ratio < 1.3)
        {
            center.x = cvRound((r->x + r->width*0.5)*scale);
            center.y = cvRound((r->x + r->height*0.5)*scale);
            radius = cvRound((r->width + r->height)*0.25*scale);
            circle(img, center, radius, color, 3, 8, 0);
        }
        else
        {
            rectangle(img, CvPoint(cvRound(r->x*scale), cvRound(r->y*scale)),
                CvPoint(cvRound((r->x + r->width - 1)*scale), cvRound((r->y + r->height - 1)*scale)),
                color, 3, 8, 0);

        }
    }
    writer.write(img);
    cv::imshow(title,img);
}
```

然后在采集的方法（void play()）中调用这个方法即可实现人脸识别。这里不展示结果。

3.部分高斯模糊

部分高斯模糊的关键是找到ROI（感兴趣区域）

其中在人脸识别的时候已经找出ROI了，所以只要对找到ROI进行高斯模糊就可以了

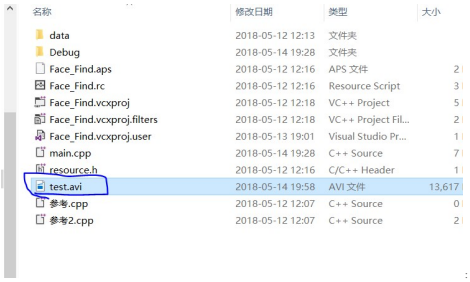
```
/*************************************************
函数名称: detectAndDraw
函数功能: 绘制圆圈和高斯模糊
传入参数:
返 回 值:
建立时间: 2018-05-13
修改时间:
建 立 人:
修 改 人:
其它说明:
*************************************************/
void MyClass::detectAndDraw(Mat& img,char* title)
{
```

```cpp
    Mat dog;
    dog = imread("F:\\Pictures\\gdog.png",1);
    Mat tempdog;
    double alpha = 1; double beta = 1 - alpha;

    double scale = 1;
    vector<Rect> faces;
    Mat gray,smallImg(cvRound(img.rows / scale), cvRound(img.cols / scale), CV_8UC1);;
    cvtColor(img, gray, COLOR_BGR2GRAY);
    resize(gray, smallImg, smallImg.size(), 0, 0, INTER_LINEAR);
    equalizeHist(smallImg, smallImg);
    cascade.detectMultiScale( smallImg, faces,1.1,2,0|CASCADE_SCALE_IMAGE,Size(30,30));//匹配人脸
    for (vector<Rect>::const_iterator r = faces.begin(); r != faces.end(); r++){
        Scalar color = colors[6];//红色
        int radius;
        Point center;
        double aspect_ratio = (double)r->width / r->height;
        if (0.75 < aspect_ratio && aspect_ratio < 1.3)
        {
            center.x = cvRound((r->x + r->width*0.5)*scale);
            center.y = cvRound((r->x + r->height*0.5)*scale);
            radius = cvRound((r->width + r->height)*0.25*scale);
            circle(img, center, radius, color, 3, 8, 0);
            Mat *imageROI=&img(Rect(r->x, r->y, 2 * radius, 2 * radius));//感兴趣区域
            Mat temp = img(Rect(r->x, r->y, 2 * radius, 2 * radius));
            cv::GaussianBlur(temp, temp, Size(21, 21), 3, 3);
            cv::GaussianBlur(temp, temp, Size(21, 21), 3, 3);
        }
        else
        {
            rectangle(img, CvPoint(cvRound(r->x*scale), cvRound(r->y*scale)),
                CvPoint(cvRound((r->x + r->width - 1)*scale), cvRound((r->y + r->height - 1)*scale)),
                color, 3, 8, 0);

        }
    }
    writer.write(img);
    cv::imshow(title,img);
}
```

结果：



名称	修改日期	类型	大小
data	2018-05-12 12:13	文件夹	
Debug	2018-05-14 19:28	文件夹	
Face_Find.aps	2018-05-12 12:16	APS 文件	2
Face_Find.rc	2018-05-12 12:16	Resource Script	3
Face_Find.vcxproj	2018-05-12 12:18	VC++ Project	5
Face_Find.vcxproj.filters	2018-05-12 12:18	VC++ Project Fil...	2
Face_Find.vcxproj.user	2018-05-13 19:01	Visual Studio Pr...	1
main.cpp	2018-05-14 19:28	C++ Source	7
resource.h	2018-05-12 12:16	C/C++ Header	1
test.avi	2018-05-14 19:58	AVI 文件	13,617
参考.cpp	2018-05-12 12:07	C++ Source	0
参考2.cpp	2018-05-12 12:07	C++ Source	2

成功生成人脸识别视频文件。

如需要源码请转移至码云：https://gitee.com/cjqbaba/MediaTest/tree/Face_Find进行源码克隆下载

如有问题请留言评论。转载请注明出处，谢谢。

标签： C++， opencv3.0， 人脸识别， 高斯模糊， ROI

好文要顶 关注我 收藏该文

重交亲爸爸
关注 - 0
粉丝 - 6
+加关注

0 推荐 0 反对

« 上一篇： 基于opencv下对视频的灰度变换，高斯滤波，canny边缘检测处理，同窗体显示并保存

» 下一篇： 基于opencv3.0下的运动车辆检测

posted @ 2018-05-14 20:05 重交亲爸爸 阅读(1692) 评论(1) 编辑 收藏 举报

公告

昵称： 重交亲爸爸
园龄： 4年
粉丝： 6
关注： 0
+加关注

我的标签

二维码识别(1)

更多

随笔档案

2019年3月(1)

2018年5月(6)

2018年3月(2)

阅读排行榜

1. 基于opencv3.0和zbar下条形码和二维码的识别与解码(19991)

2. 基于opencv3.0下的运动车辆检测(8511)

3. 在win10下给vs2013配置opencv3.0(6169)

4. 使用opencv调用24*24点阵字库和8*16ASCII字库在图片显示文字数字(2317)

5. 基于opencv下对视频的灰度变换，高斯滤波，canny边缘检测处理，同窗体显示并保存(2078)

评论排行榜

1. 基于opencv3.0下的运动车辆检测(5)

2. 算法训练 2的次幂表示(蓝桥杯C++写法)(4)

3. 基于opencv3.0和zbar下条形码和二维码的识别与解码(2)

4. 基于opencv3.0下的人脸检测和检测部分的高斯模糊处理(1)

推荐排行榜

1. 基于opencv3.0和zbar下条形码和二维码的识别与解码(5)

2. 基于opencv3.0下的运动车辆检测(3)

3. 算法训练 2的次幂表示(蓝桥杯C++写法)(1)

最新评论

1. Re:算法训练 2的次幂表示(蓝桥杯C++写法)

if(i-1==1){
    show(2);
这个不是很理解

--邢兵

2. Re:算法训练 2的次幂表示(蓝桥杯C++写法)

大佬牛批

--邢兵

3. Re:基于opencv3.0下的运动车辆检测

良心文章，感谢，学到了。

--penglaixian

4. Re:基于opencv3.0和zbar下条形码和二维码的识别与解码

请教下博主， angle的值应该是在负90到0之间，所以检测条形码这段判断代码有什么用处？新手求教//为了防止找错,要检查这个矩形的偏斜角度不能超标 //如果超标,那就是没找到 if (minRect...

--码农翻身做主人

5. Re:基于opencv3.0下的人脸识别和识别部分的高斯模糊处理

这篇也是，人脸检测被当成人脸识别了

--ChrisZZ