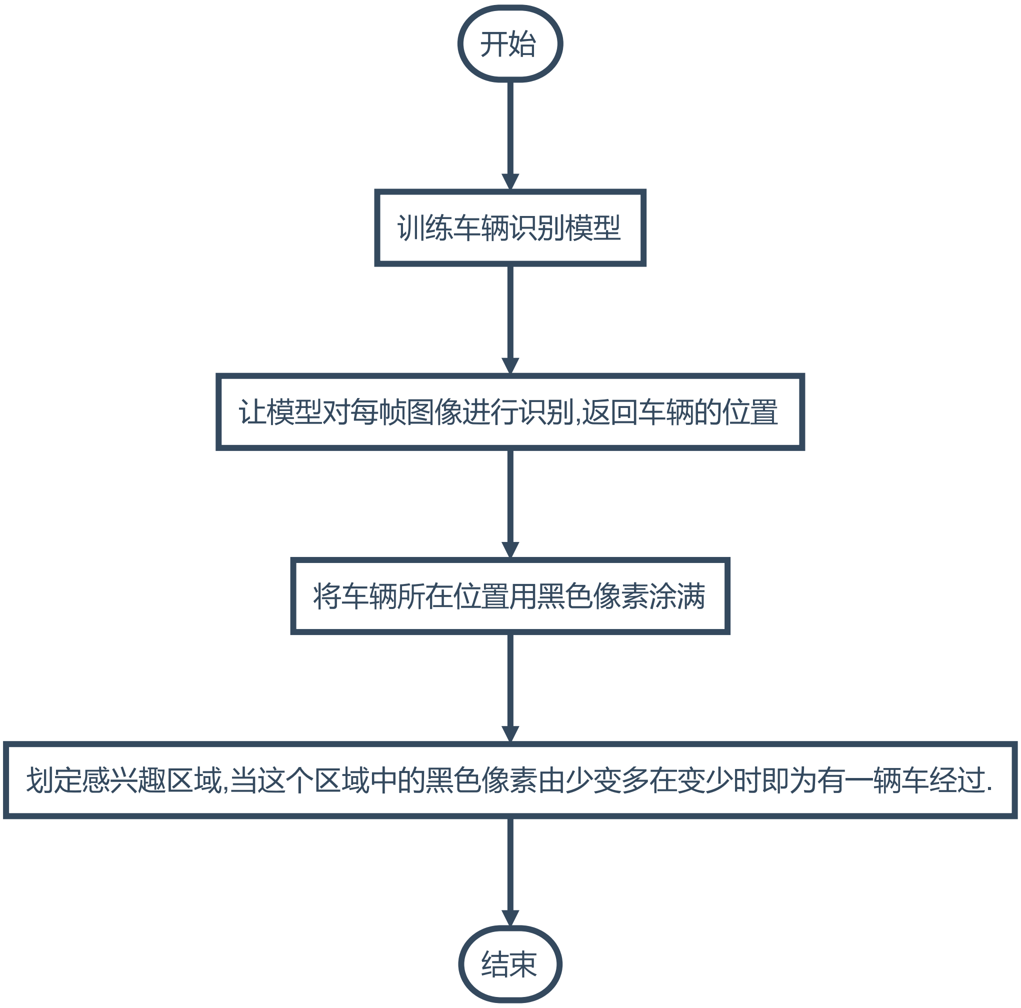
图像处理课程设计

实验四

对一个车流视频进行实时的车辆跟踪和车流统计

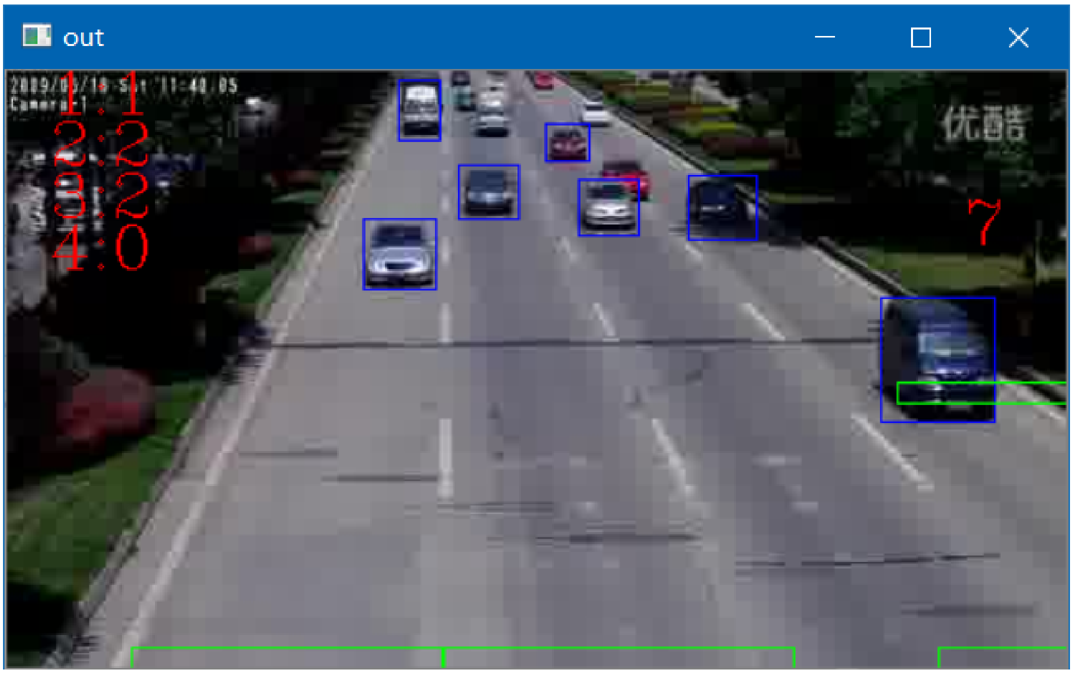
- 解题思路


开始
训练车辆识别模型
让模型对每帧图像进行识别,返回车辆的位置
将车辆所在位置用黑色像素涂满
划定感兴趣区域,当这个区域中的黑色像素由少变多在变少时即为有一辆车经过.
结束

- 代码

```python
# encoding:utf-8

import requests
import cv2
import numpy as np
import copy

def car_detect(img):
    res = requests.post('http://127.0.0.1:24401/',
                        params={'threshold': 0.2},
                        data=img).json()
    return res['results']

def main():
        cv2.namedWindow('out')
        count1 = 0
        count2 = 0
        count3 = 0
        count4 = 0
        flag1  = 0
        flag2  = 0
        flag3  = 0
        flag4  = 0
        for i in range(1,1564,1):
            img = cv2.imread('car_frame/frames_{:0>5d}.jpg'.format(i))
            img_copy = copy.deepcopy(img)
            res = car_detect(cv2.imencode('.jpg',img)[1].tostring())

            for p in res:
                x1 = int(512 * p.get('x1'))
                y1 = int(288 * p.get('y1'))
                x2 = int(512 * p.get('x2'))
                y2 = int(288 * p.get('y2'))

                contours = np.array([[x1,y1],[x2,y1],[x2,y2],[x1,y2]])
                cv2.fillPoly(img,pts=[contours],color=(0,0,0))
                k1 = img[278:288,60:210]
                k2 = img[278:288,211:380]
                k3 = img[278:288,450:512]
                k4 = img[150:160,430:512]

                cv2.rectangle(img_copy,(x1,y1),(x2,y2),(255,0,0),1)
                cv2.rectangle(img_copy,(60,278),(210,288),(0,255,0),1)
                cv2.rectangle(img_copy,(211,278),(380,288),(0,255,0),1)
                cv2.rectangle(img_copy,(450,278),(512,288),(0,255,0),1)
                cv2.rectangle(img_copy,(430,150),(512,160),(0,255,0),1)

                n1 = len(k1[k1==0])
                n2 = len(k2[k2==0])
                n3 = len(k3[k3==0])
                n4 = len(k4[k4==0])

            if(n1>1500):
                flag1=1
            if(n2>1500):
                flag2=1
            if(n3>1000):
                flag3=1
            if(n4>1500):
                flag4=1

            if(n1<800 and flag1==1):
                count1=count1+1
                flag1=0
                cv2.rectangle(img_copy,(60,278),(210,288),(0,0,255),1)
            if(n2<800 and flag2==1):
                count2=count2+1
                flag2=0
                cv2.rectangle(img_copy,(211,278),(380,288),(0,0,255),1)
            if(n3<400 and flag3==1):
                count3=count3+1
                flag3=0
            cv2.rectangle(img_copy,(450,278),(512,288),(0,0,255),1)
        if(n4<800 and flag4==1):
            count4=count4+1
            flag4=0
            cv2.rectangle(img_copy, (430, 150), (512, 160), (0, 0, 255), 1)
        cv2.putText(img_copy,
                    "1:{}".format(count1),
                    (20,20),
                    cv2.FONT_HERSHEY_COMPLEX,
                    1.0,
                    (0, 0, 255))
        cv2.putText(img_copy,
                    "2:{}".format(count2),
                    (20,45),
                    cv2.FONT_HERSHEY_COMPLEX,
                    1.0,
                    (0, 0, 255))
        cv2.putText(img_copy,
                    "3:{}".format(count3),
                    (20,70),
                    cv2.FONT_HERSHEY_COMPLEX,
                    1.0,
                    (0, 0, 255))
        cv2.putText(img_copy,
                    "4:{}".format(count4),
                    (20,95),
                    cv2.FONT_HERSHEY_COMPLEX,
                    1.0,
                    (0, 0, 255))

        cv2.putText(img_copy,
                    "{}".format(len(res)),
                    (462,83),
                    cv2.FONT_HERSHEY_COMPLEX,
                    1.0,
                    (0, 0, 255))
        cv2.imshow('out',img_copy)
        cv2.waitKey(5)
    cv2.waitKey()

main()
```

- 效果


out
2009/05/16 Sat 11:40:05
Camera 1
1:1
2:2
3:2
4:0
优酷
7