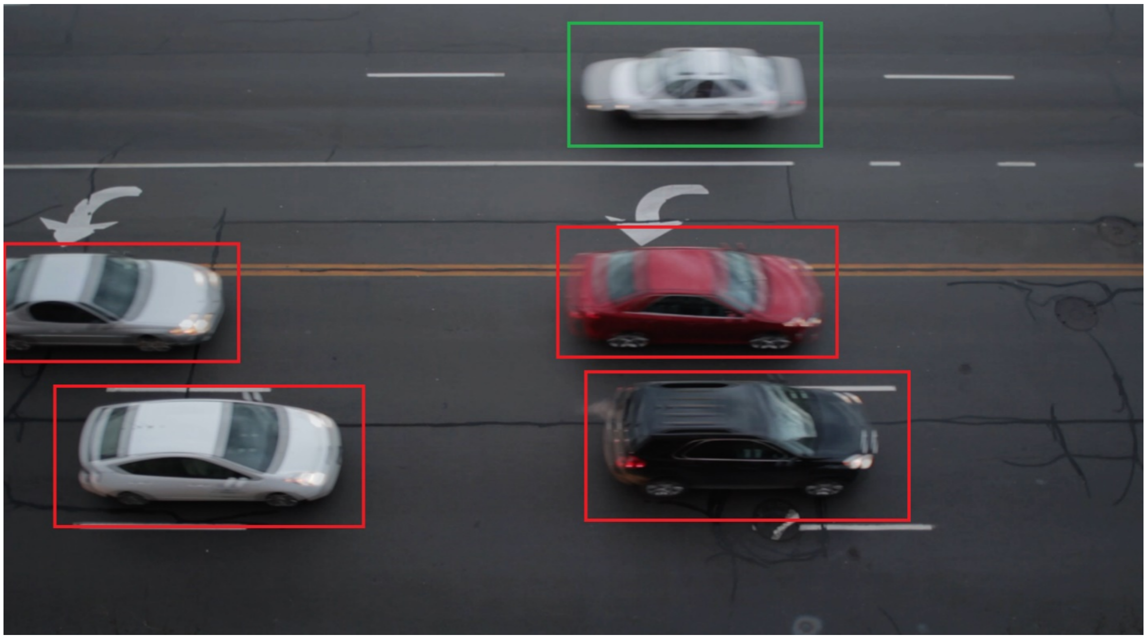
CSE 4310: Introduction to Computer Vision
Summer 2021
Program #3: Counting Traffic

In this assignment, you will implement a traffic counting program using OpenCV. The goal is to design a basic video processing pipeline that will count the number of cars that have traveled through a certain stretch of road in either direction.

The program should take a single command line argument, which will contain the file path for the input video which is provided as part of the assignment specification. The same video file we be used for grading, so you do not need to worry about "tuning" your pipeline for multiple video files.

In the input video, a horizontal stretch of road containing 6 lanes of traffic (3 lanes in either direction) is shown. We will assume that right-to-left traffic is Westbound, and left-to-right traffic is Eastbound. Your program will process each video frame and maintain a count of cars that have passed in either direction (two counts total). Additionally, you will need to draw a rectangular bounding box around each vehicle as it passes through the scene (for the duration of the pass). Westbound traffic should be marked with a green rectangle, while Eastbound traffic should be marked red. An example of the rectangle output is shown below:



When counting traffic, the program console should print updated totals each time a vehicle is detected. The precise frame in which a vehicle is counted is up to you, as long as the count is updated while the target vehicle is still at least partially visible in the frame. The count should be displayed as follows:

```
WESTBOUND COUNT: 5
EASTBOUND COUNT: 3
```

Your program should run on the class development environment using OpenCV. You will need to download the updated VirtualBox image or recompile OpenCV with the missing video codecs (see update notes on Github or the end of 7/12 class video for more info) in order for the video to open correctly. If you implement your program using C++ (highly recommended), you must include a working CMakeLists.txt file with your source code. The program should be able to be compiled and executed by running the following set of commands in the program directory if C++ is used:

```
cmake .
make
./program3 <PATH TO FILE>
```

For example…

```
cmake .
Make
./program3 road_traffic.mp4
```

You may choose to use python for the assignment, though this is not recommended. Your program should then be able to run in the class development environment with the following set of commands in the program directory:

```
python program3 <PATH TO FILE>
```

For example…

```
python program3 road_traffic.mp4
```

You may develop your application on your own system / OS, but it must run properly on the class development environment at the time it is evaluated. Submit your source code, test images, and all other necessary files to blackboard in a single zip file by the deadline. Late submissions will incur a penalty of 10 points per day after the deadline.

Points will be assigned as follows:

1. Program counts Eastbound traffic successfully – 20 points
2. Program counts Westbound traffic successfully – 20 points
3. Program draws rectangles on Eastbound traffic successfully – 30 points
4. Program draws rectangles on Eastbound traffic successfully – 30 points

Partial credit may be given, at the discretion of the grader, for items which are not fully functional or contain bugs. You may be asked to demo your program to the grader if errors occur during the initial run. Write your code as cleanly as possible with proper formatting and comments in order to bolster your case for partial credit on non-functional features. It is highly recommended to follow a standard coding style, such as ANSI C++, Google C++, etc. in your program (HINT: The Code::Blocks IDE provided with the development environment has a built in formatter under the plugins menu).