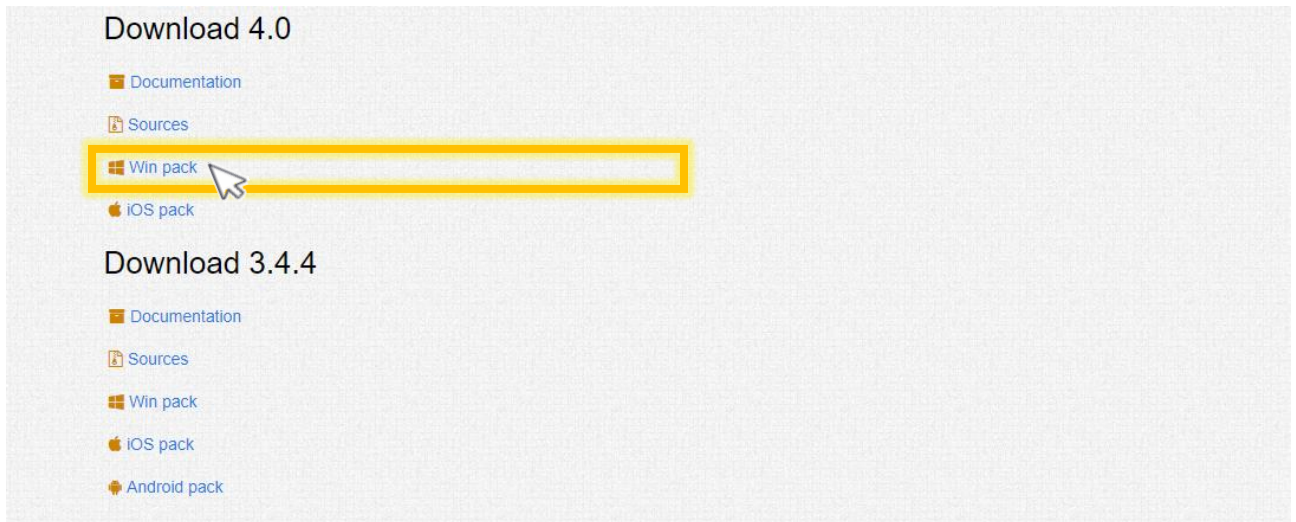


OPENCV 4.0 LIBRARY INSTALLATION FOR WINDOWS WITH VISUAL STUDIO

1. Go to <https://www.opencv.org/opencv-4-0-0.html> and download the **Win pack** version of opencv



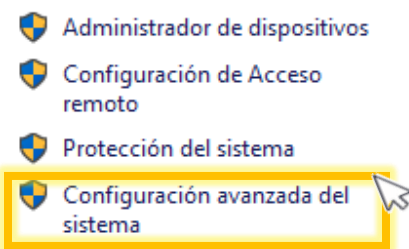
Note: Documentation can be found here <https://docs.opencv.org/4.0.0/>

2. Opendv library installation

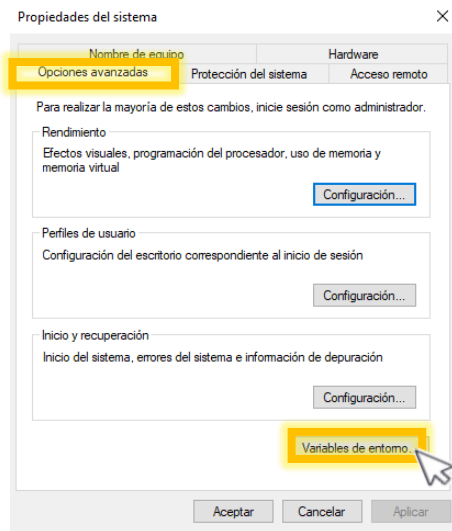
- 2.1. Create the path **C:\opencv\opencv4**
- 2.2. Double click to the **opencv-4.0.0-vc14_vc15.exe** file
- 2.3 Install on the previous path

3. Configure opencv library

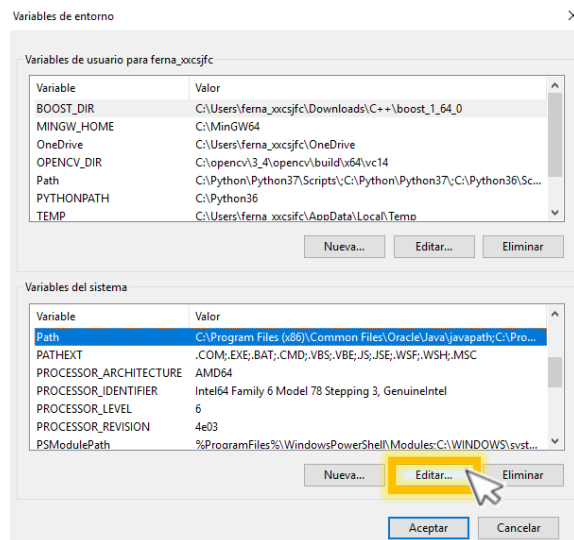
- 3.1. Open windows file explorer
- 3.2. Right click on **This PC \ Properties**
- 3.3. Click on **Advanced System Setting**



3.4. Click on **Advanced Option** \ **Environment Variables**

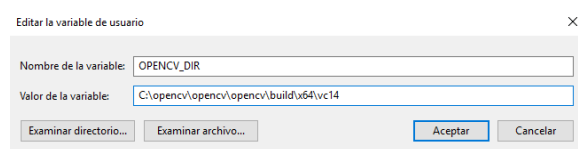


3.5. Go to **System Variables** tab, search for **Path** variable and click on the **Edit** button



3.6. Click on **New** button and set to **C:\opencv\opencv4\opencv\build\x64\vc14\bin**

3.7. On **User Variables** click on **New** button and set as the next Figure (**C:\opencv\opencv4\opencv\build\x64\vc14** and **OPENCV_DIR**)



Note: **vc14** is for **Visual Studio 2015**, for **Visual Studio 2017** use **vc15**

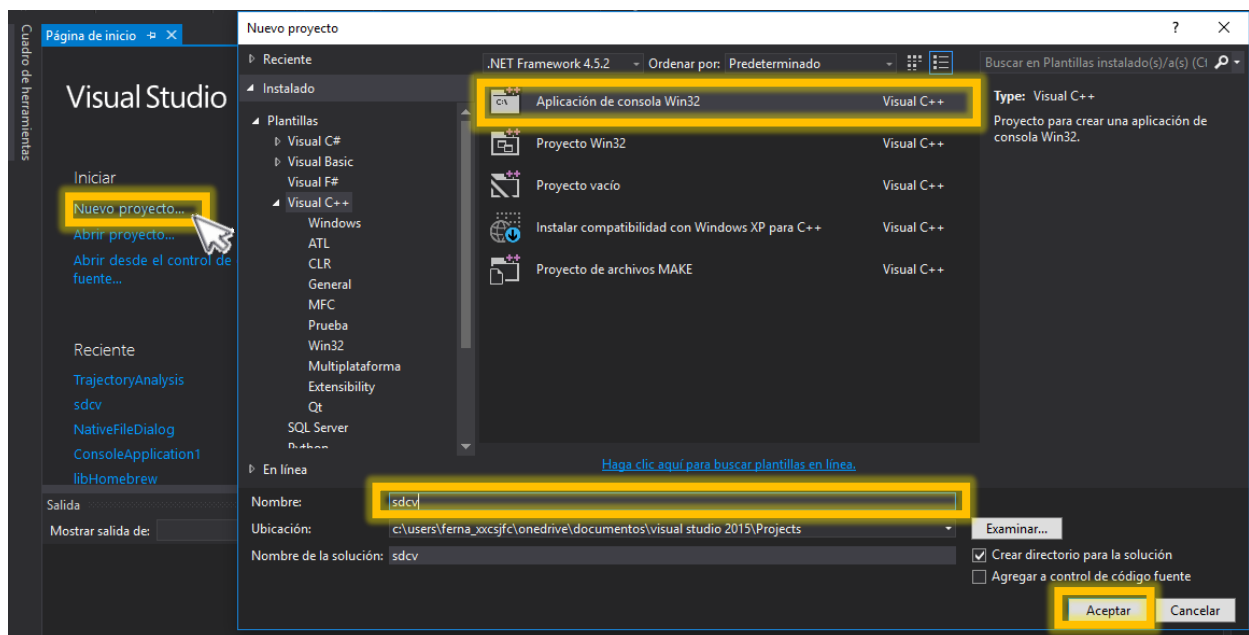
CONFIGURE SDCV PROJECT ON WINDOWS AND VISUAL STUDIO

1. Open [Visual Studio](#) 2015 or 2017

2. Create a new project

2.1. Click on [New Project](#)

2.2. Configure as the next figure



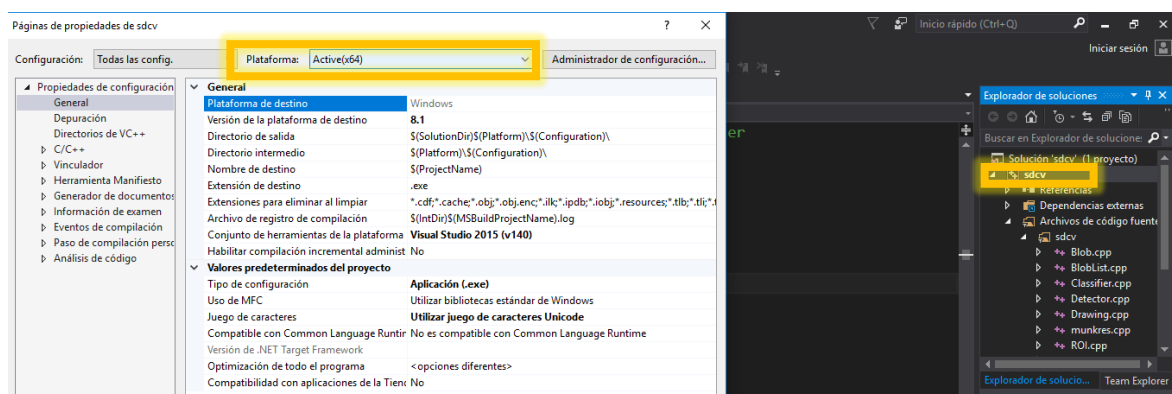
2.3. Click on next until the project is created

3. Configure the project

3.1. Copy the content of the [main.cpp](#) and paste on the [main.cpp](#) of the new project

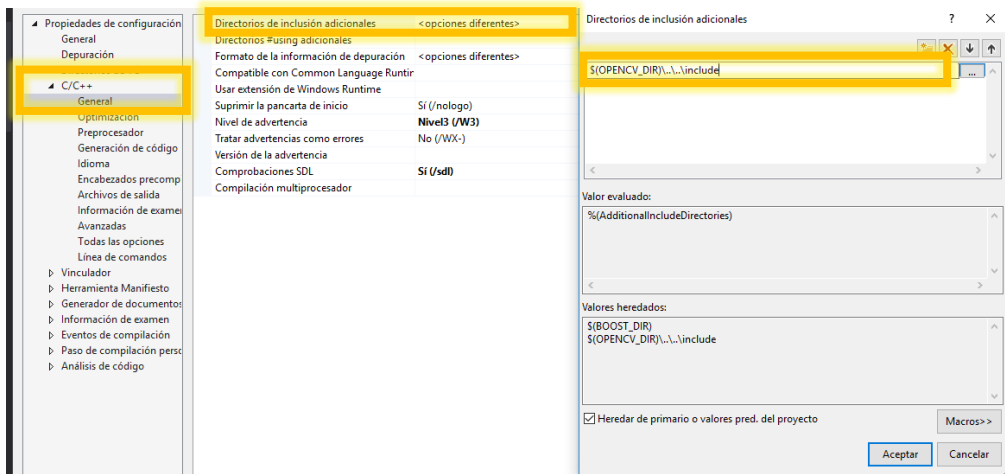
3.2. Click on the [sdcv](#) project and then press [Alt+Enter](#)

3.3. On [Platform](#), change to [Active\(x64\)](#)



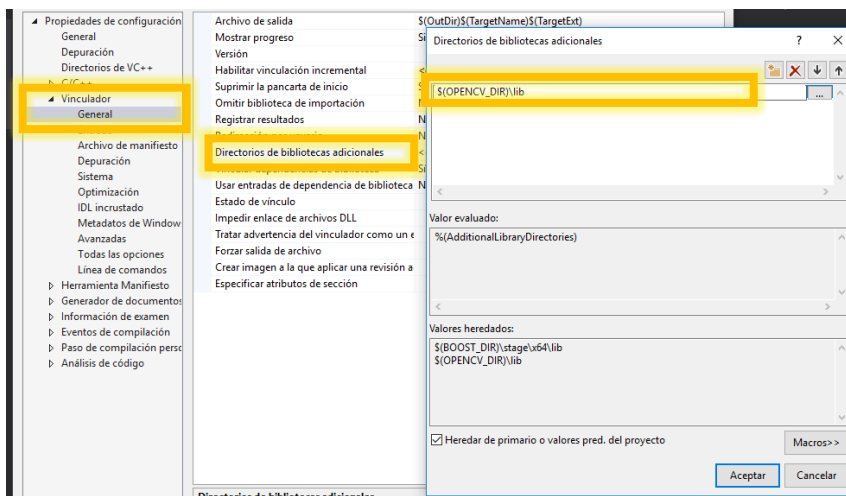
3.4. Expand the **C/C++** tab, and select **General**

3.5. Click on **Additional Include Directories**, expand and click on **Edit**, set as the Figure
(`$(OPENCV_DIR)\..\..\include`)



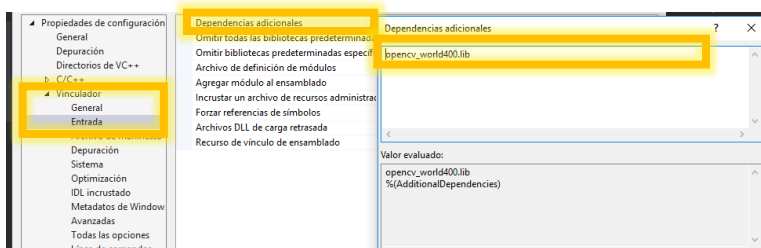
3.6. Expand the **Linker** tab, and select **General**

3.7 Click on **Additional Library Directories**, expand and click on **Edit**, set as the Figure
(`$(OPENCV_DIR)\lib`)



3.8. Expand the **Linker** tab, and select **Input**

3.9. Click on **Additional Dependencies**, expand and click on **Edit**, set as the Figure
(`opencv_world400.lib`)



HOW TO RUN THE SDCV SOFTWARE

1. Before build the solution set the **video name** as well as the number of **lane division lines**

```
cv::String videoname = "V01-2016-07-10"; int NumDivLines = 4;  
//cv::String videoname = "V18-2016-10-05-1340";  
//cv::String videoname = "V06-2016-06-15-1838"; int NumDivLines = 4;  
//cv::String videoname = "V07-2016-07-15-1840";  
//cv::String videoname = "video720";  
//cv::String videoname = "20161005_134318";  
//cv::String videoname = "V19-2016-10-05-1343-420x240-25F-19";  
//cv::String videoname = "V07-2016-07-15-1840";  
//cv::String videoname = "V18-2016-10-05-1340-420x240-25F-18";
```

In this example the video name is **V01-2016-07-10**, and the number of lane division lines is **4**

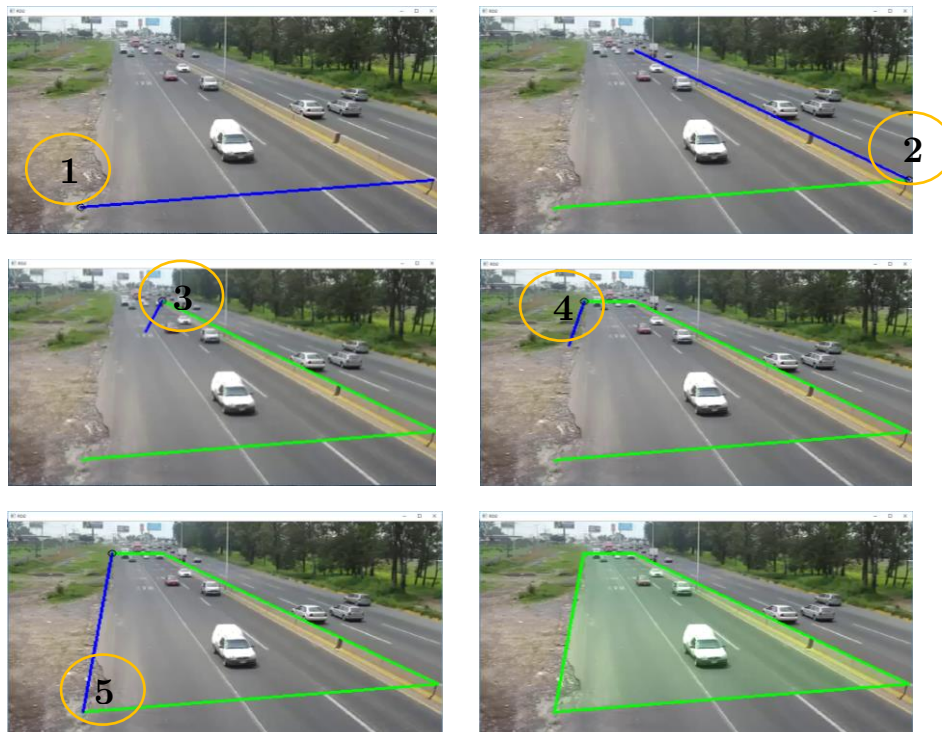
1.1. All videos must be in the folder **sdcv\sdcv** where the **main.cpp** can be found

1.2. All videos are **mp4** files

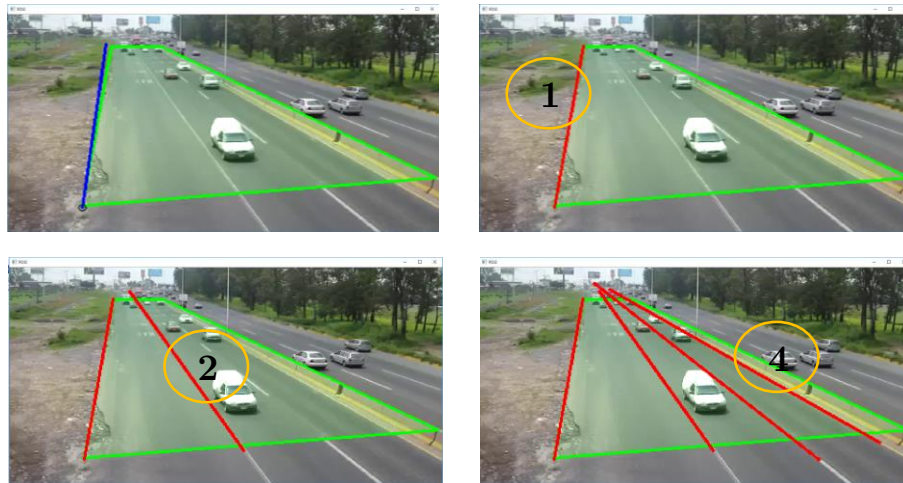
2. When a new video is loaded, you need to define several parameters of the **Region Of Interest (ROI)**:

2.1. The ROI: In the same order as can be shown in the figures

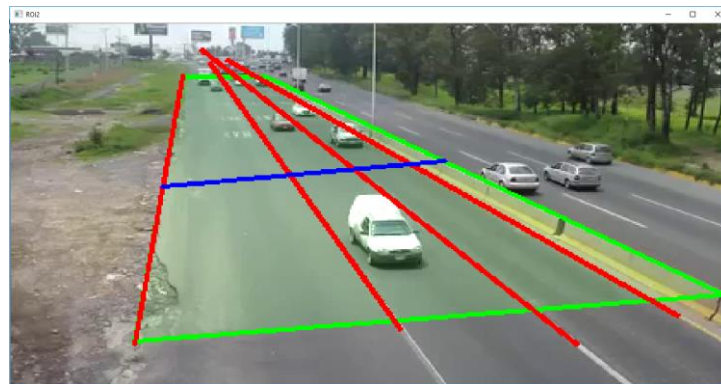
2.1.1. The first point is **left bottom**, next right bottom, and son on



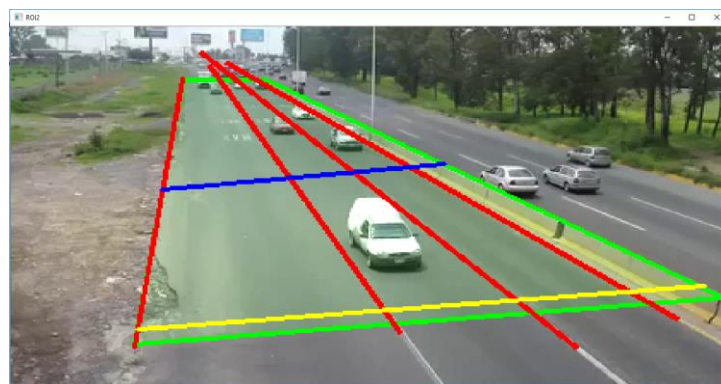
2.2. Lane division lines: Left to right



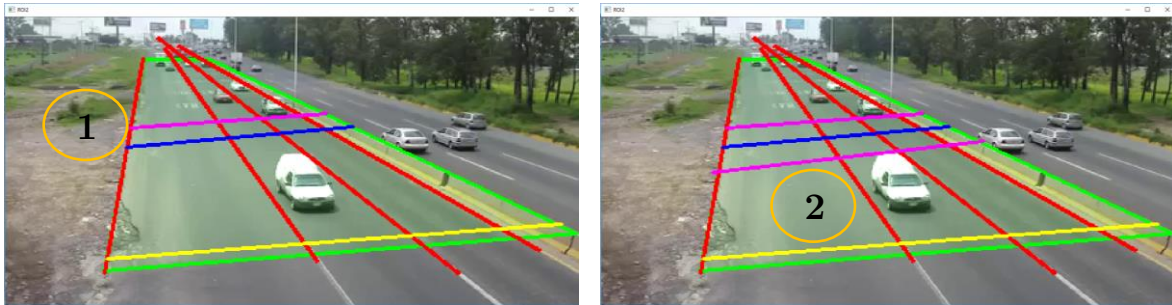
2.3. Detection Line (Blue Line)



2.4. End Line (Yellow)



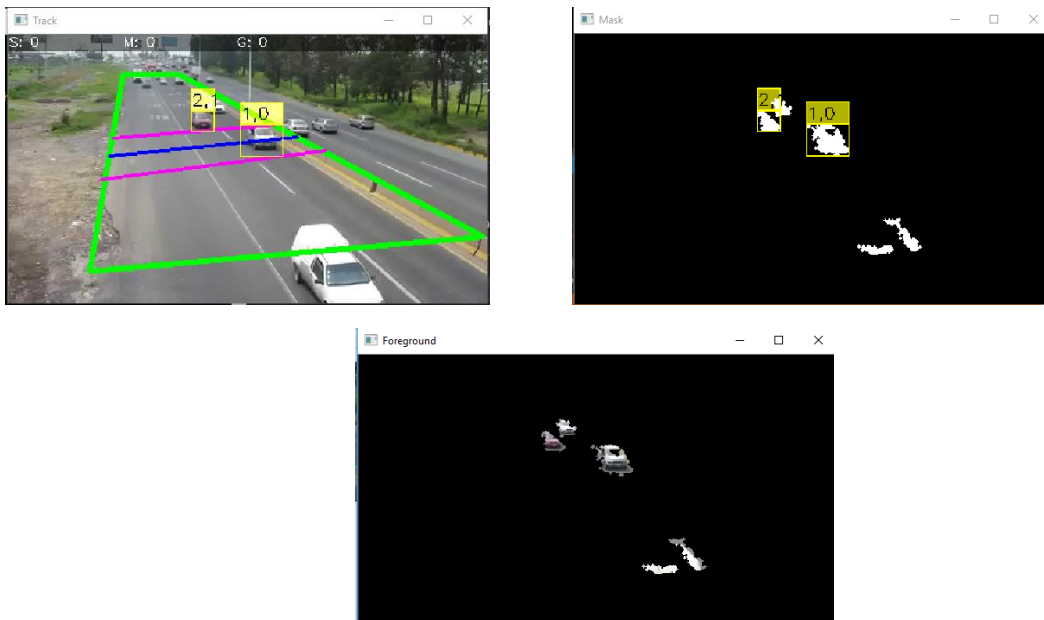
2.5. Classification Region Lines (Magenta Lines): For a vehicle flow top to bottom or **front view**, the first line to be defined is the nearest to the (0,0) pixel.



2.6. To continue press enter

3. After ROI parameters definition, a folder will be created in `sdcv\sdcv\DATA\{videoname}` which contains a configuration file named as `{videoname}.yaml`

4. If the configuration file exist, step 2 will be skipped, and the software will start



4.1. Track window shows the rgb frame where the tracking and counting will be displayed

4.2. Mask window shows the mask of the detections as well as the tracking results

4.3. Foreground window shows the detected vehicles in RGB

4.4. Keyboard Controls

4.4.1. The video can be paused if you press **SPACE** key, and resume pressing again **SPACE**

4.4.2. Also the video can be stopped with the **ESC** key

4.4.3 The video will be restarted (just the video, the count and the background model cannot be restarted) with the **1** key.

4.5 When the software finishes, several files will be created at **sdcv\sdcv\DATA\{videoname}**

4.5.1 **background.bmp**, the background image at a frame N (where the software was finished)

4.5.2 **class.csv**: the classification results in matrix columns **[ID,OCC,CLASS]**, ID is the vehicle ID, OCC indicates wheatear the vehicle was detected as a result of the occlusion handling algorithm of not, CLASS is the class ID, 0,1,...,N, if a -1 is shown, the vehicle has not been classified yet.

4.5.3 **time.csv**: The timing results, displayed in matrix columns **[Total, Detection, Tracking, Drawing]** total is the elapsed time per frame, Detection shows the detection step results, as well as for Tracking and Drawing. The first value is the **Frames Per Second** (FPS) of the video.

4.5.4 **voi.csv**

4.5.5 **statistics.csv**: Traffic statistics

4.5.6 **video_out.mp4**: Processed video

4.6. The SDCV software cannot automatically detect the Traffic Flow Direction (TFD) due to the Kalman filter dynamics matrix, so to configure the TFD in the Tracker.cpp file a variable denoted as **trafficOrientation** is used for. To setup the TFD for a front view:

```
trafficOrientation = sdcv::eOrientationLine::TOP_DOWN;
```

Otherwise

```
trafficOrientation = sdcv::eOrientationLine::BOTTOM_UP;
```

