Assignment 4: Data-driven engineering from fleets

Tianshuo Xiao

tianshuo@chalmers.se

MPMOB

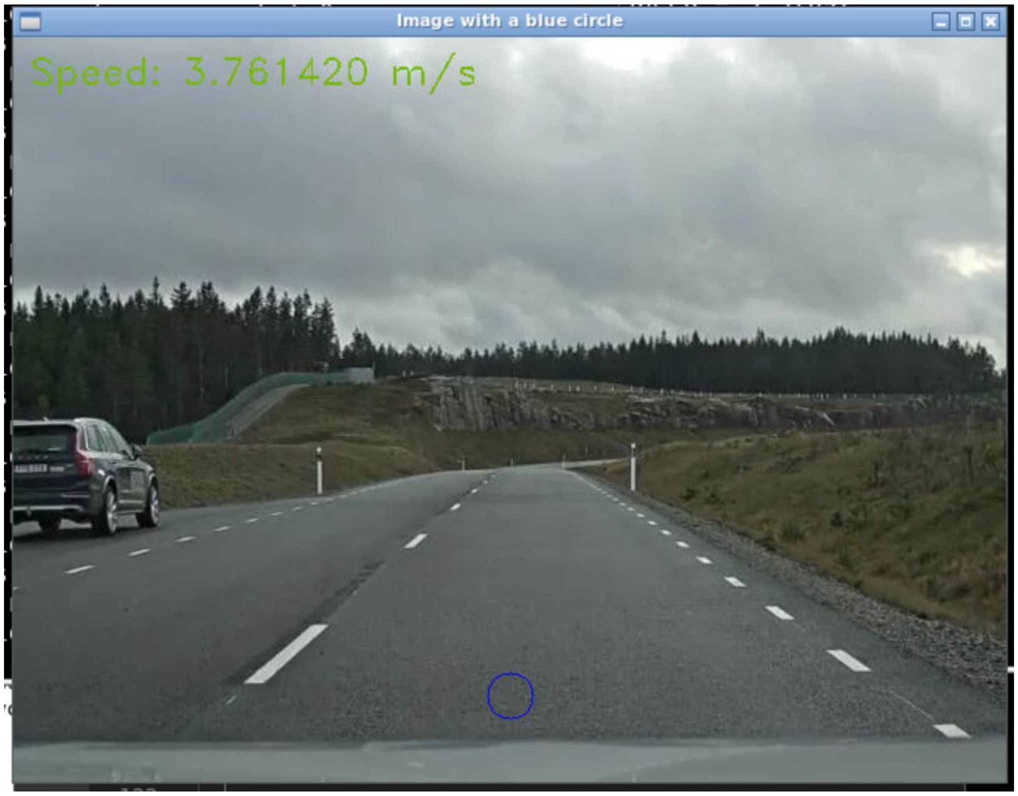
First, I drew a circle with a radius of 15 in the middle of the video. The image width is 640 and height is 480, so I choose the midpoint of the horizontal coordinate (320). We want to make it possible to move the circle up and down depending on the speed when the vehicle accelerates, so we calculate the y coordinate by ***480-int(15*speed)***.

```
// Draw a blue circle
cv::circle(img, cv::Point(320, 480-int(15*speed)), 15, cv::Scalar(255, 0, 0)); // circle
```

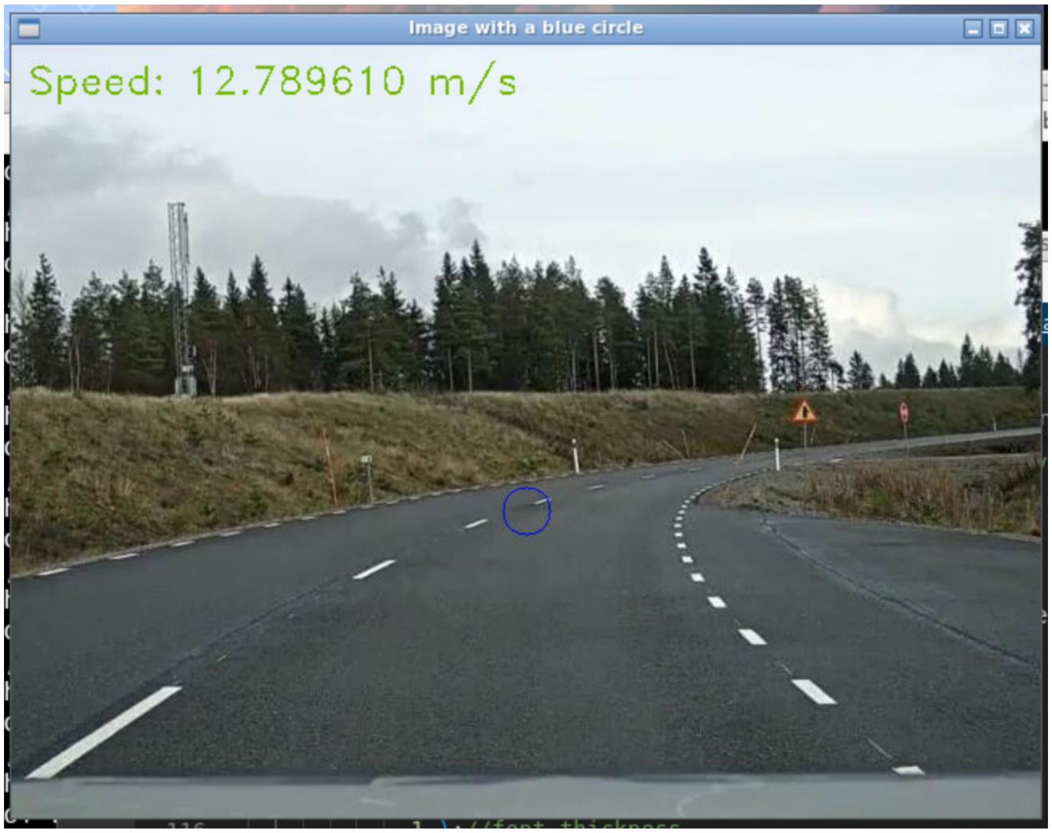
In the top left corner of the image, print the speed of the vehicle.

```
std::string speedText = "Speed: " + std::to_string(speed) + " m/s"; // speed text
cv::putText(img, //target image
        speedText, //text
        cv::Point(10, 30), //top-left position
        cv::FONT_HERSHEY_DUPLEX, //typeface
        0.8,//type size
        CV_RGB(118, 185, 0), //font color
        1 //font thickness);
```

The results are displayed as follows:



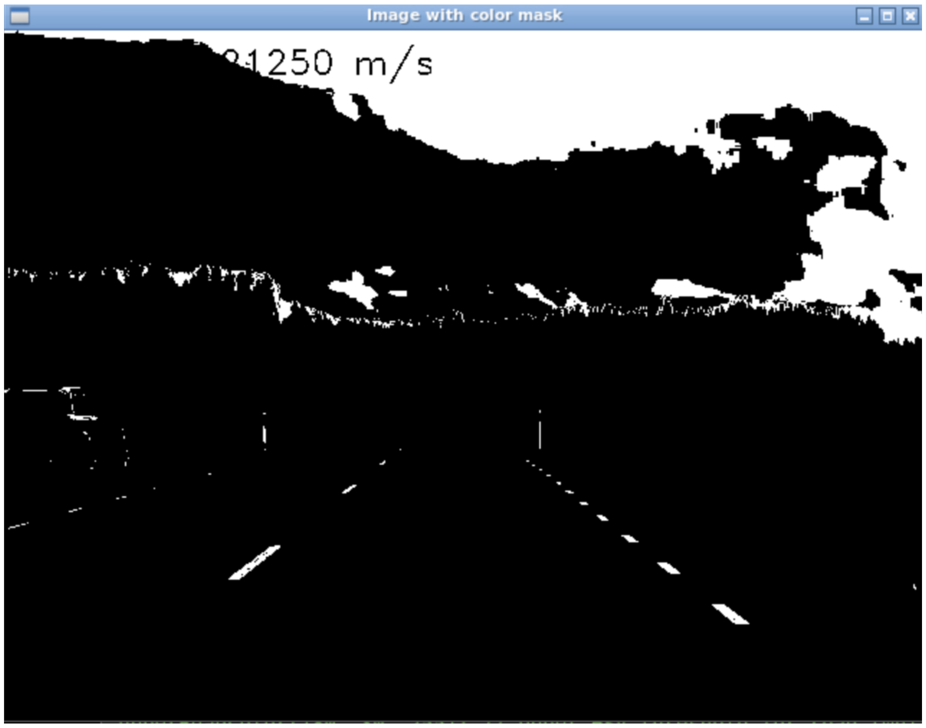
When speed up, the circle's position will move ahead on the road:



Then, I create a color filter to filter out the white road markings. When I try to detect objects by adjusting thresholds through a range of pixel values in the HSV color space, the filtering is not very effective as the white of the sky and the white of the road signs are particularly similar.

```
cv::Mat mask;
cv::inRange(hsvImg, cv::Scalar(0, 0, 180), cv::Scalar(180, 35, 255), mask);
```

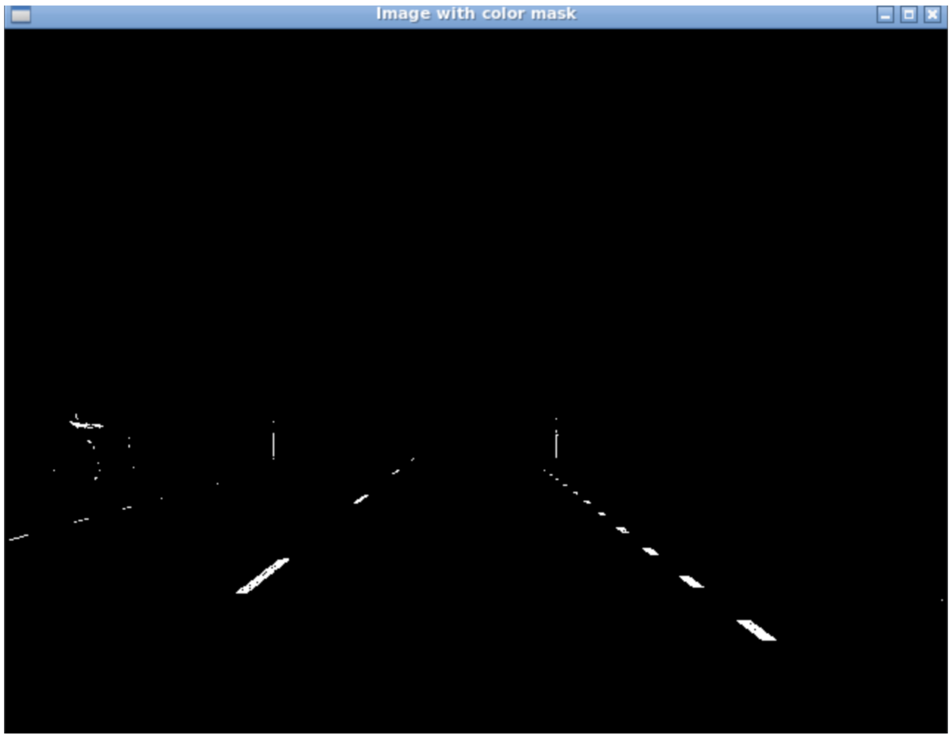
The filtering result shows as follows.



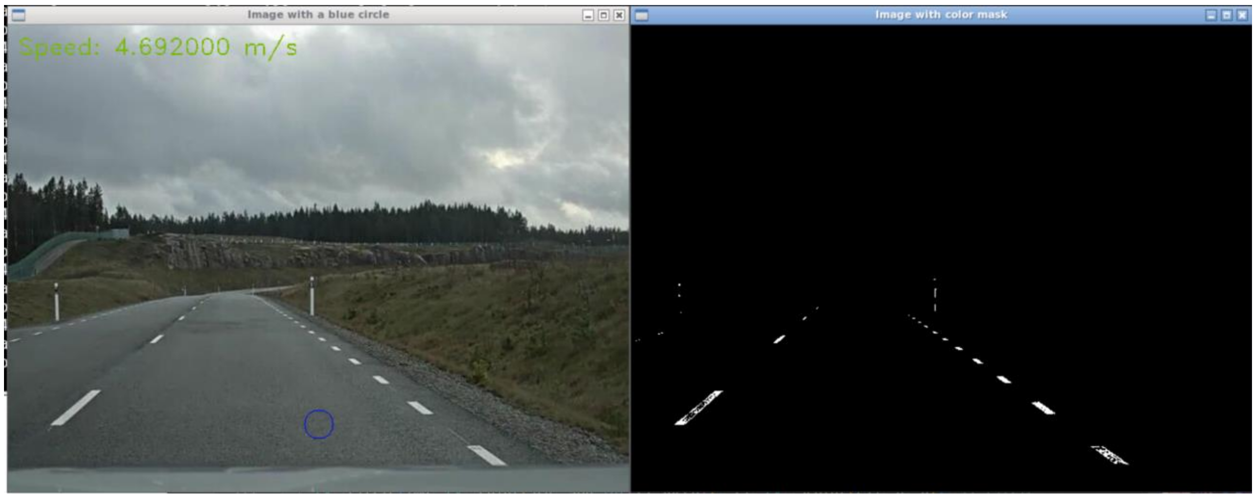
So, I added a masking layer to cover the top half of the image to improve the filtering effect.

```
cv::rectangle(img,cv::Point(0,0),cv::Point(640,260),cv::Scalar(0,0,255),-1); //new layer
```

The final result as following:



Inclusion, my final result is shown below:



Commands:

You need to run ***xhost +*** first

First login the docker:

docker login registry.git.chalmers.se

Then, download the reply video

docker run -ti --rm --init --net=host --ipc=host -v /tmp:/tmp -e DISPLAY=$DISPLAY registry.git.chalmers.se/ola.benderius/mms210-assignment-datareplay:rest_handler

Open another terminal, go into folder as this README.md file, run

docker build -t myapp .

Then run the image

docker run --rm -ti --init --net=host --ipc=host -v /tmp:/tmp -e DISPLAY=$DISPLAY myapp

Finally, you can get the results.