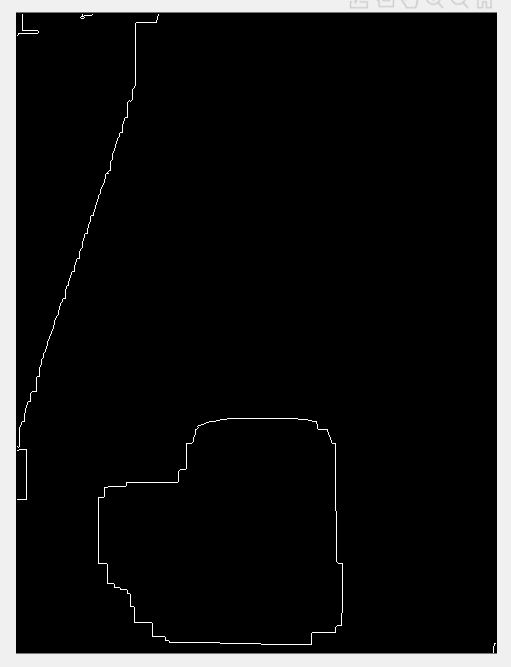
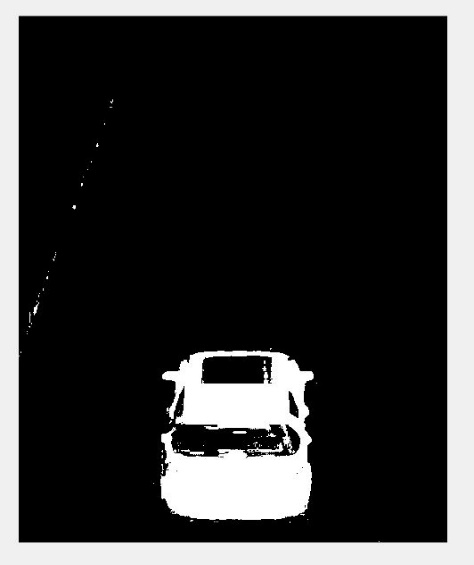
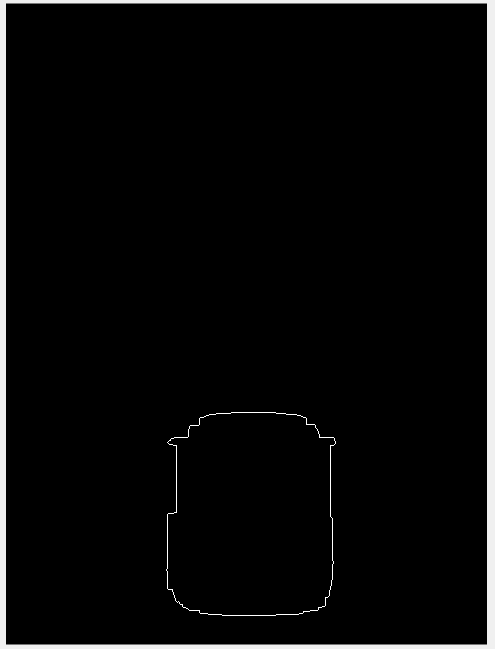
**Part 2 - Report**

At first I just tried to filter out the image by converting the original image into grey but that did not help with the edge detection because, it would read the backgrounds and the shadow of the car. To fix this problem I changed from rgb2gray to rgb2hsv. This was filtered out the blue colour in the image so that it would only detect the edges of the car.

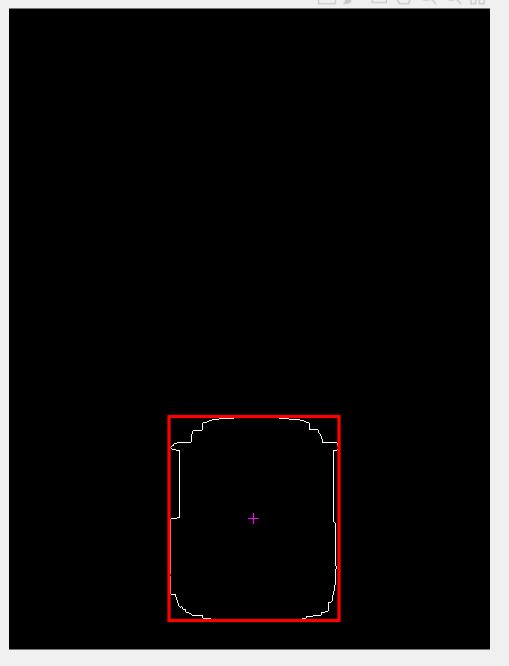
With rgb2gray:   


With  
rgb2hsv:  


I used the technique I learned from tutorial 14 to erode the background noises I had to experiment with the threshold values to get the right image and also used gaussian to filter out the background noises. After that I got the image of the car being filtered out with no background noises and no shadows being captured. I proceeded to detect the edges of the car by using the function edge(image, ‘zerocross’, filter). By doing that I got an image of only the edges of the of the car it was smooth

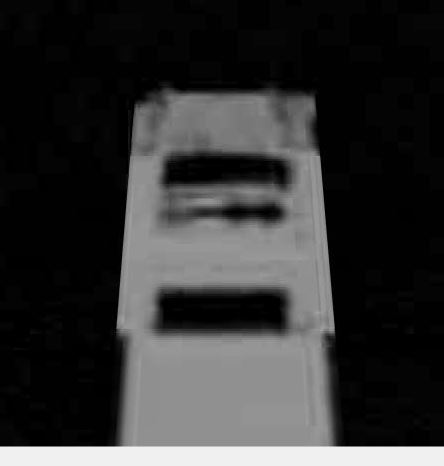
**Here is what it looks like:**

**Finding out the car’s width**  
I used regionprops to plot the centre of the car, draw a bounding box to get the size of the car. Once, I get the width value i set an if function to check if the car’s width exceeds the limit of 2.5m. If it does exceeds the limit it will print out “Oversized” otherwise “Not Oversized”.

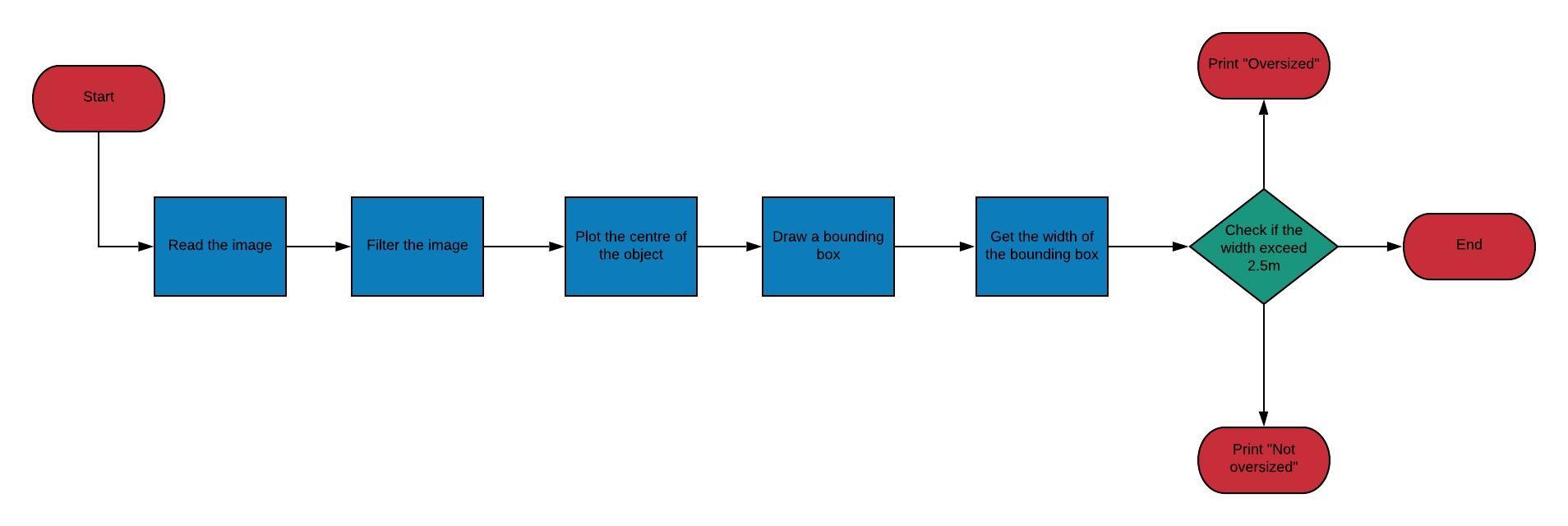
Here is what it looks like after using bounding box:  


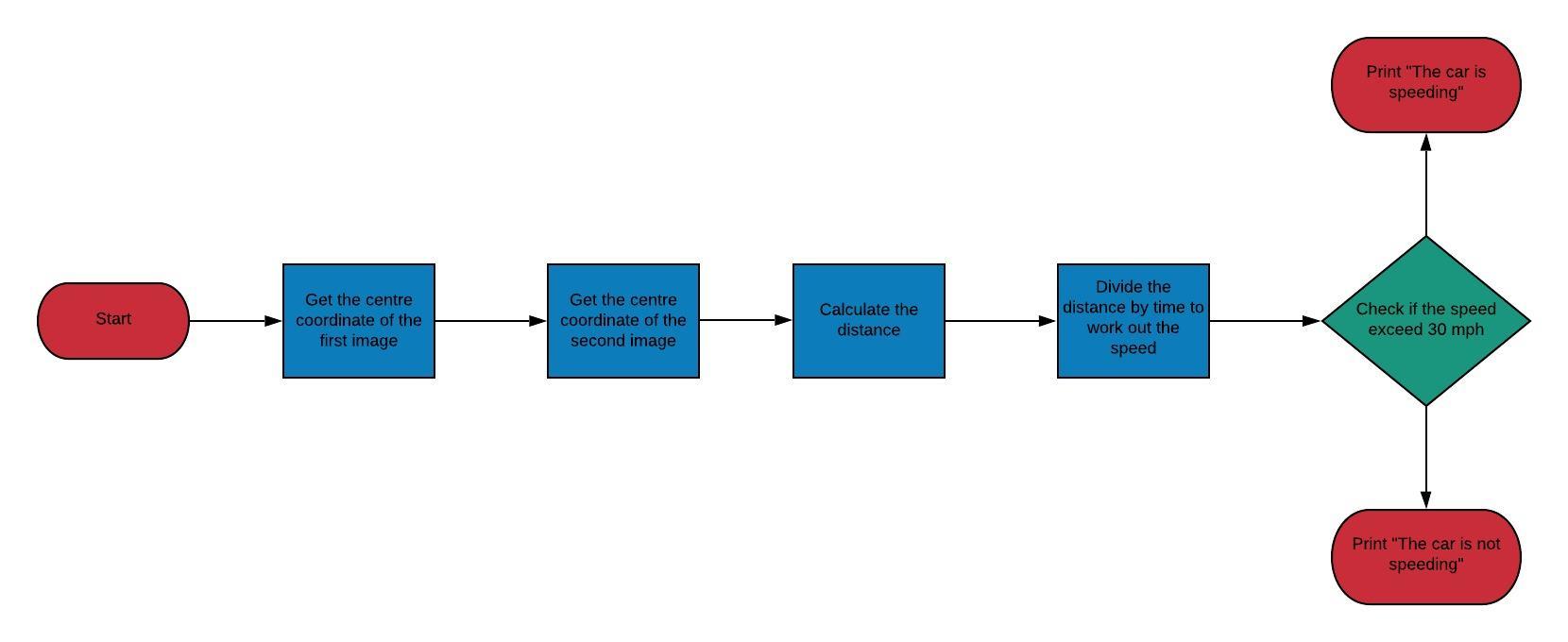
**Finding out the car’s speed**  
In order to find out the car’s speed i used the centre coordinates of the car from two different images. For example, I would find out the centre coordinates of the car from image 001.jpg and 002.jpg divide x and y coordinates by 100 to convert it to m.. Once I get those value i used the formula to find out the length and that will be my distance. I then convert it into miles by dividing the distance by 1609.344. After that i use the Distance/Time(0.000277778) to get the value of speed (mph). I choose to divide the distance by 0.000277778 because 0.1s converted into hours is 0.000277778 h. I also wrote a function which will check if the car is speeding or not. By checking the variable speed’s value. If it is bigger than 30 i will print out “The car is speeding” otherwise it will print out “The car is not speeding”.

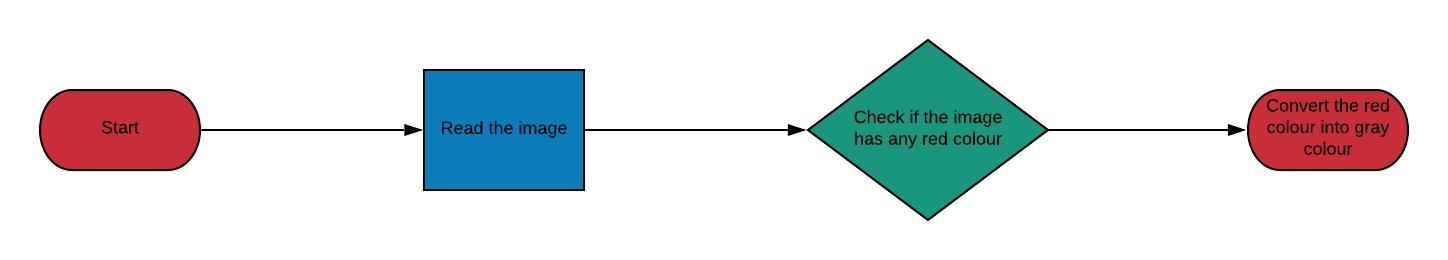
**Finding out the fire engine**  
To find out the car is a fire engine or not i tried to detect the red colours in the image by using the function diff\_im = imsubtract(im(:,:,1) and then convert the red colours in the image into grey colour. If any colour in the image is other than red it will just convert it into black colour. I could not figure out how to create a function that could read an image and print out the colour it sees

Here is what it looks like when it detects red colour:  


**Flow diagram of the entire system**

Check if the car’s width is oversized or not  


Check if the car’s speed is exceeding the speed limit  


Check if the car is a fire engine  


**Test against the application scenarios**

|  |  |  |
| --- | --- | --- |
| Specification | Inputs | Output (Screenshots) |
| Check the width of the car | oversized.jpg |  |
| Check the speed of the car 001.jpg vs 002.jpg | 001.jpg  002.jpg |  |
| Check if the car is a fire engine | fire01.jpg |  |
| Check the speed of the car 001.jpg vs 003.jpg | 001.jpg  003.jpg |  |
| Check the speed of the car 001.jpg vs 004.jpg | 001.jpg  004.jpg |  |