# BBM418 - COMPUTER VISION LABORATORY ASSIGNMENT 1

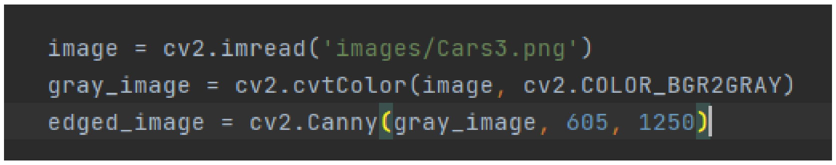
Sadık Can ACAR
21626843

March 27, 2021

## 1 Edge Detection

In this assignment, I used Canny Edge Detection. The reason I used Canny Edge Detection is that I can select thresholds, which is the feature the library offers me. Correctly adjusted thresholds in photos can even give me edge images with only plate lines. I found it by trying the thresholds in the cv2.Canny (image, threshold1, threshold2) function in each photo in order to reduce the pollution in the images where the lines are detected and to locate the plate more easily.

### 1.1 Code Snippet



```
image = cv2.imread('images/Cars3.png')
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
edged_image = cv2.Canny(gray_image, 605, 1250)
```

Figure 1: Code Snippet

### 1.2 Edge Detected Photos

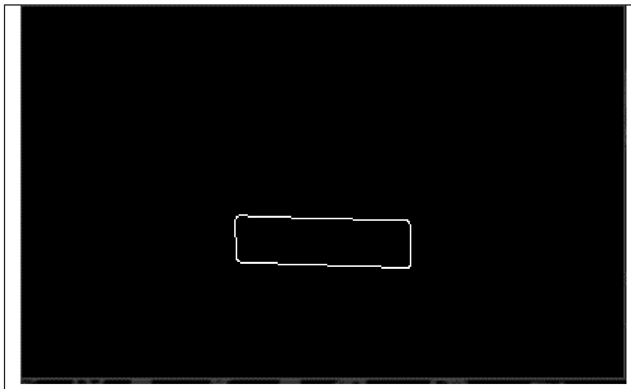
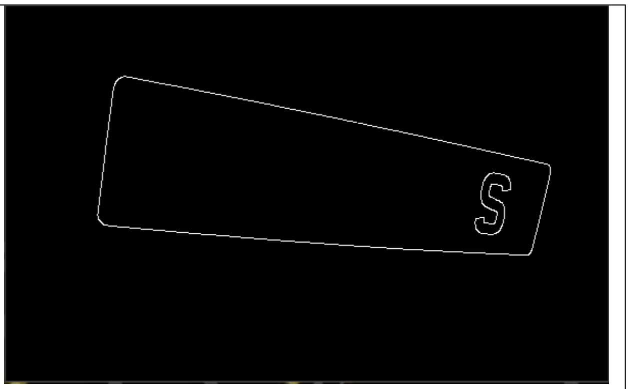
| | |
|---|---|
|  |  |

Table 1: Cars3 and Cars4 Edges

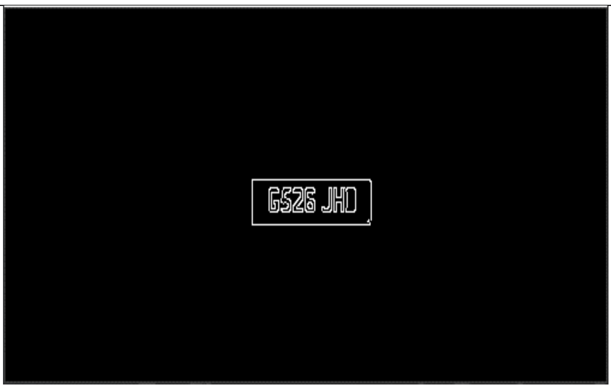
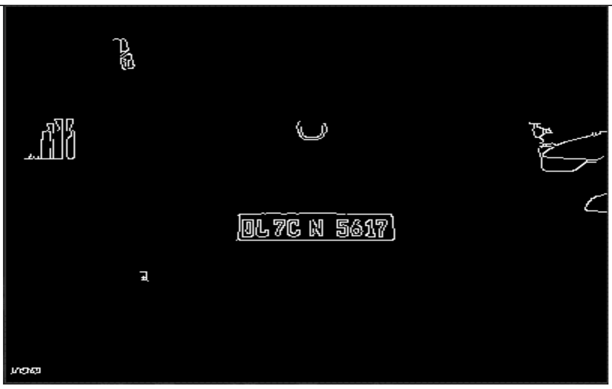
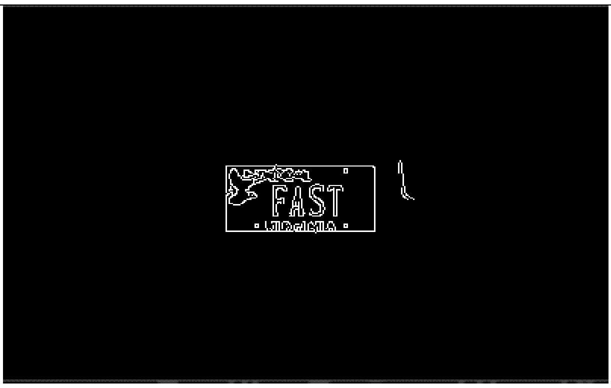
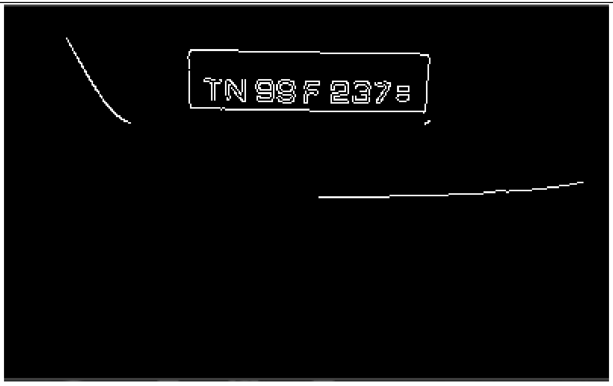
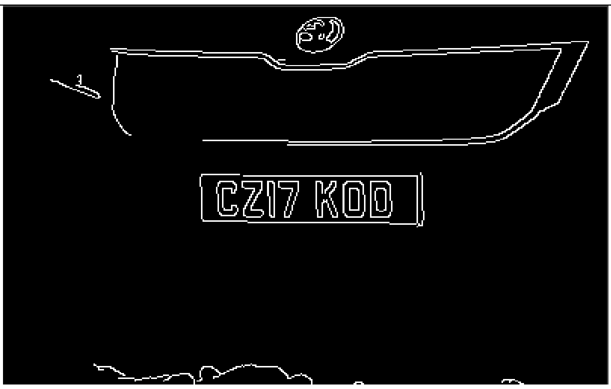
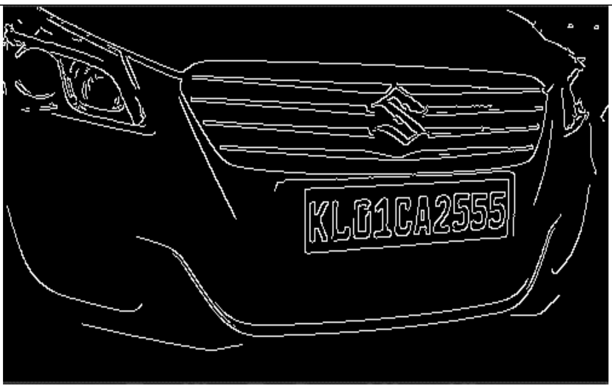
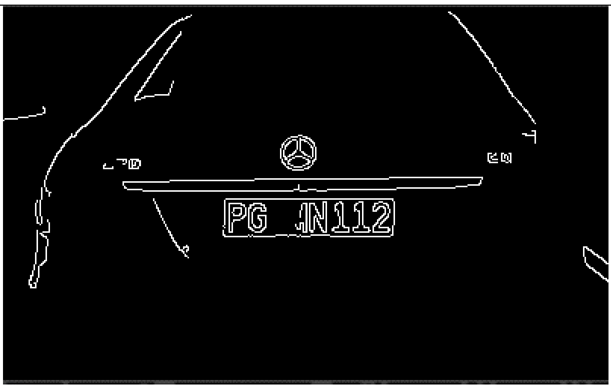
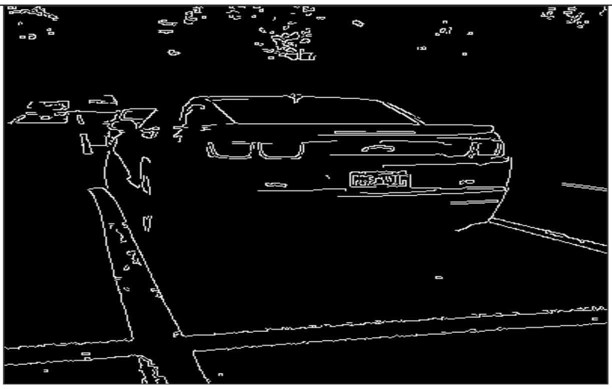
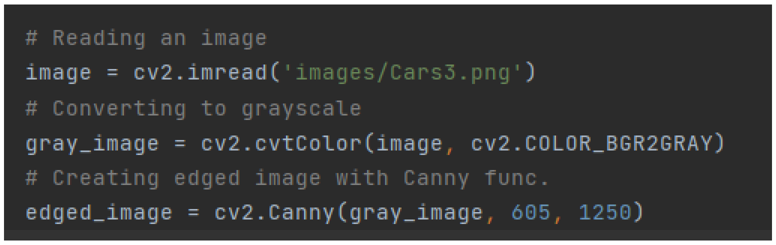
| | |
|---|---|
|  |  |
|  |  |
|  |  |
|  |  |

Table 2: Cars8, Cars34, Cars37, Cars73, Cars74, Cars0, Cars1, Cars2 Edges

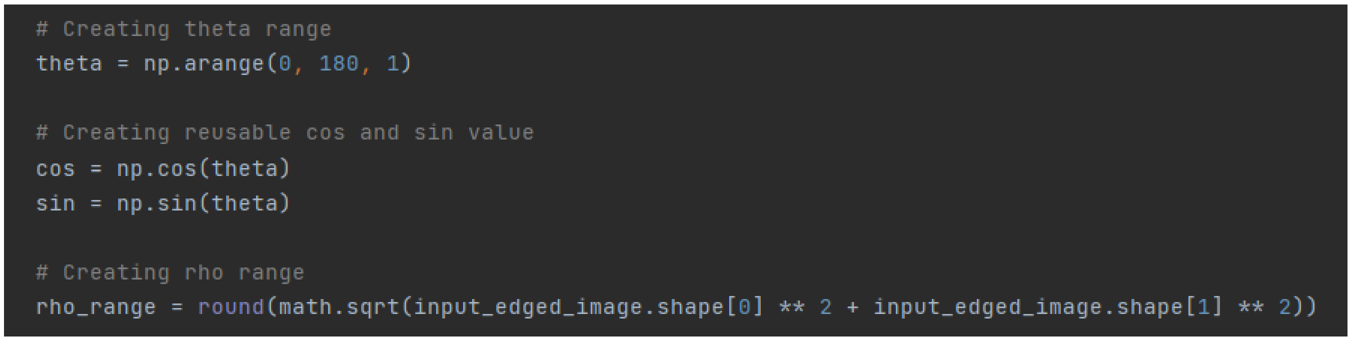
## 2 Hough Transform Implementation

Steps
1. Read image and convert image to grayscale using cv2.cvtColor().
2. Get edge image by using cv2.Canny().
3. Create range for theta and rho.
4. Generate accumulator matrix by using theta and rho.
5. Get edges pixel location (x,y).
6. Convert (x,y) coordinate system to (theta, rho) coordinate system.
7. Threshold the some high values.
8. Finally, use theta, rho value and draw the lines.



```
# Reading an image
image = cv2.imread('images/Cars3.png')
# Converting to grayscale
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# Creating edged image with Canny func.
edged_image = cv2.Canny(gray_image, 605, 1250)
```

Figure 2: Step 1-2

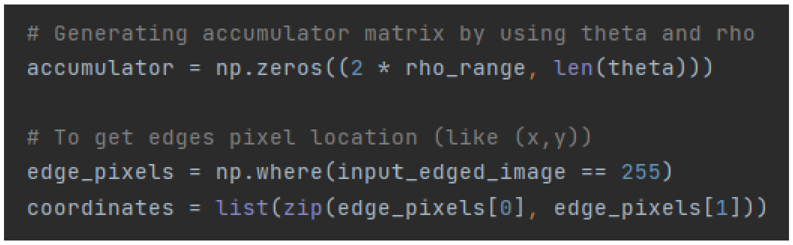


```
# Creating theta range
theta = np.arange(0, 180, 1)

# Creating reusable cos and sin value
cos = np.cos(theta)
sin = np.sin(theta)

# Creating rho range
rho_range = round(math.sqrt(input_edged_image.shape[0] ** 2 + input_edged_image.shape[1] ** 2))
```

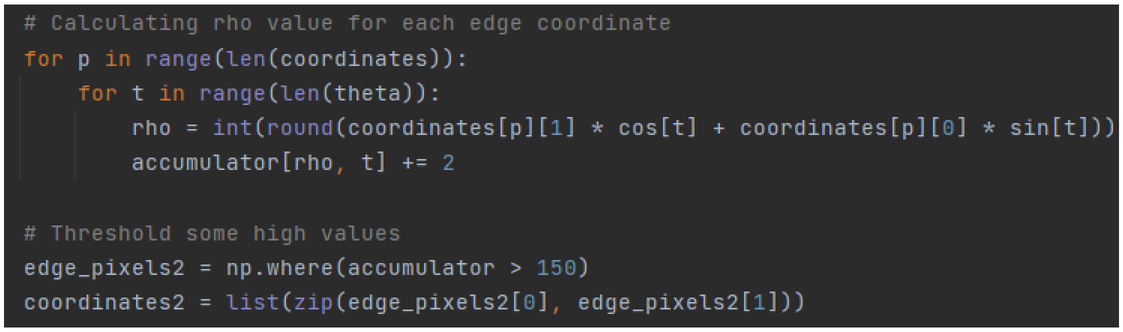
Figure 3: Step 3



```
# Generating accumulator matrix by using theta and rho
accumulator = np.zeros((2 * rho_range, len(theta)))

# To get edges pixel location (like (x,y))
edge_pixels = np.where(input_edged_image == 255)
coordinates = list(zip(edge_pixels[0], edge_pixels[1]))
```

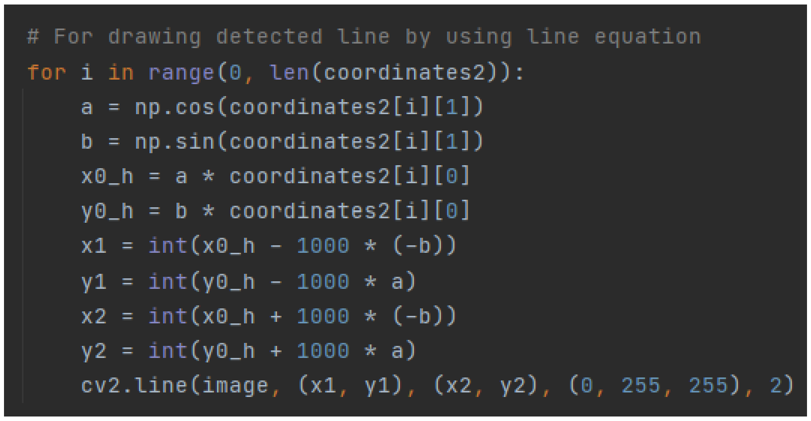
Figure 4: Step 4-5



```
# Calculating rho value for each edge coordinate
for p in range(len(coordinates)):
    for t in range(len(theta)):
        rho = int(round(coordinates[p][1] * cos[t] + coordinates[p][0] * sin[t]))
        accumulator[rho, t] += 2

# Threshold some high values
edge_pixels2 = np.where(accumulator > 150)
coordinates2 = list(zip(edge_pixels2[0], edge_pixels2[1]))
```

Figure 5: Step 6-7



```
# For drawing detected line by using line equation
for i in range(0, len(coordinates2)):
    a = np.cos(coordinates2[i][1])
    b = np.sin(coordinates2[i][1])
    x0_h = a * coordinates2[i][0]
    y0_h = b * coordinates2[i][0]
    x1 = int(x0_h - 1000 * (-b))
    y1 = int(y0_h - 1000 * a)
    x2 = int(x0_h + 1000 * (-b))
    y2 = int(y0_h + 1000 * a)
    cv2.line(image, (x1, y1), (x2, y2), (0, 255, 255), 2)
```

Figure 6: Step 8

## 3 Plate Detected Photos

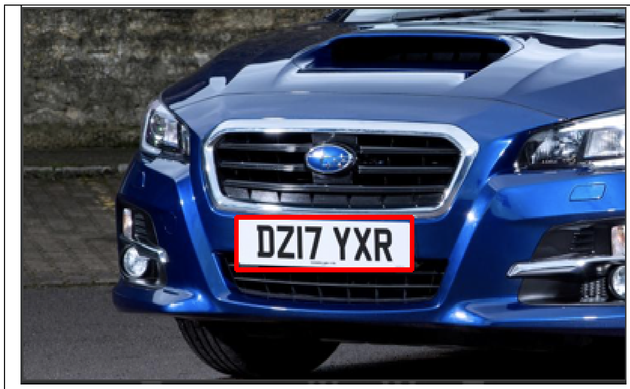
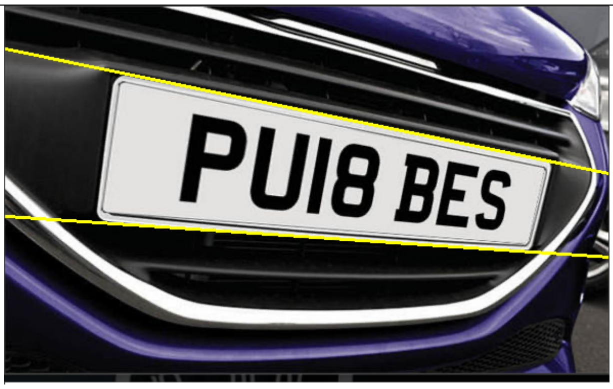
| | |
|---|---|
|  |  |

Table 3: Plate Detected Cars: Cars3, Cars4

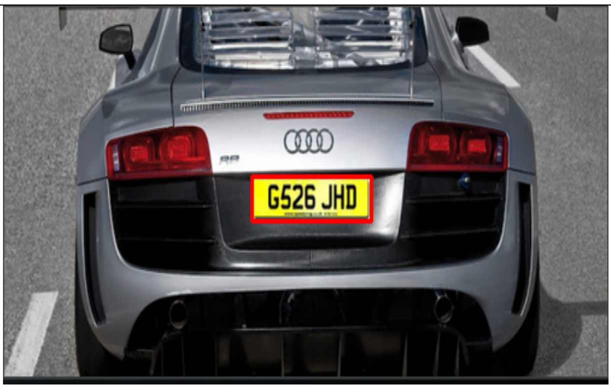
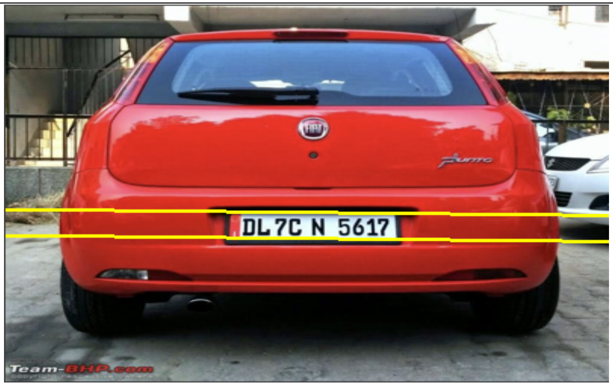
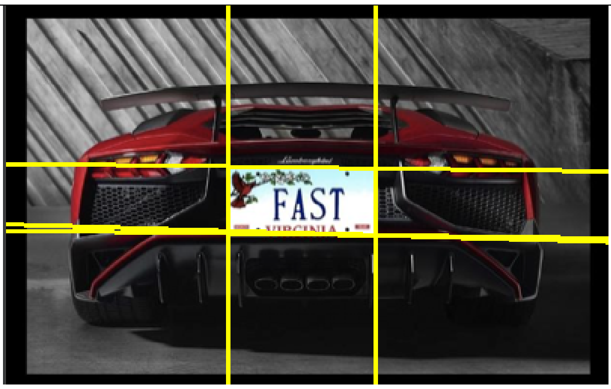
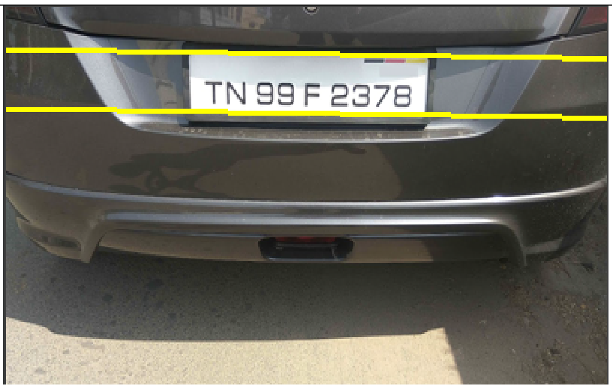
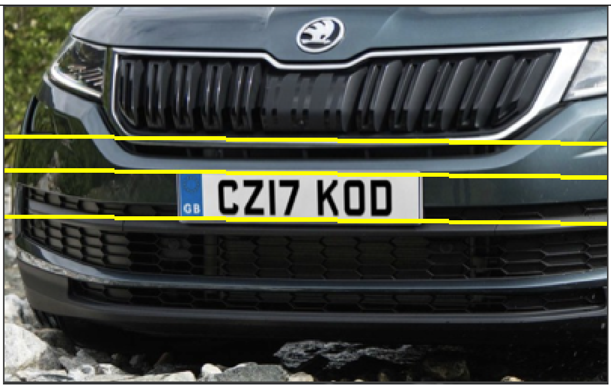
| | |
|---|---|
|  |  |
|  |  |
|  | |

Table 4: Plate Detected Cars: Cars8, Cars34, Cars37, Cars73, Cars74

## 4 Plate Undetectable Photos

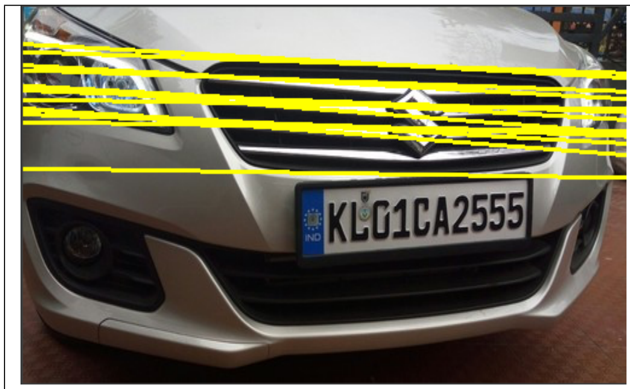
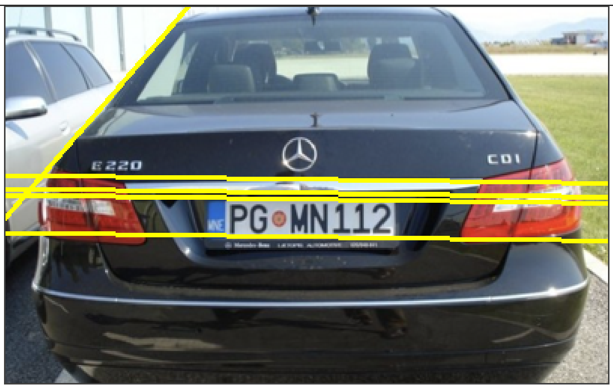
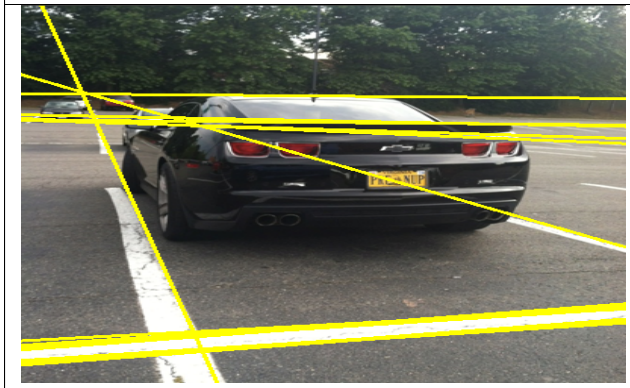
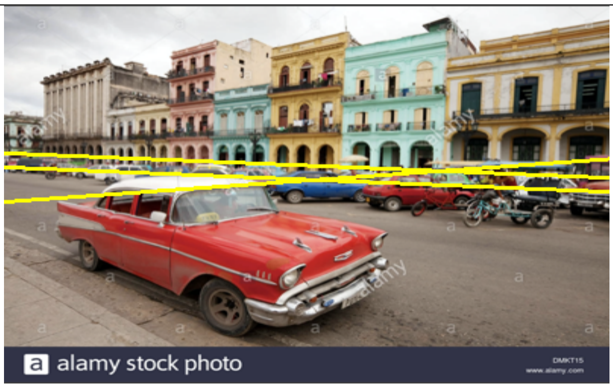
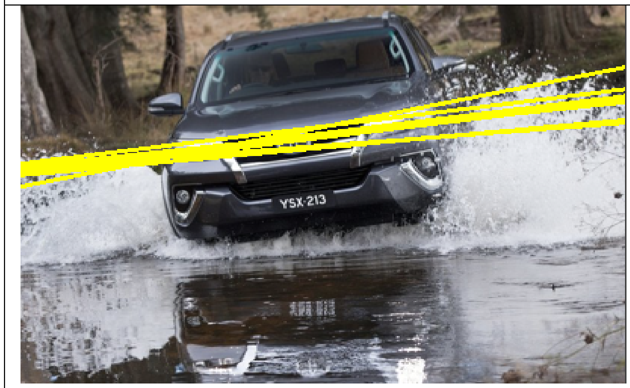
| | |
|---|---|
|  |  |
|  |  |
|  | |

Table 5: Plate Undetectable Cars: Cars0, Cars1, Cars2, Cars5, Cars7

The main reason why the location of the plate is not detected in these photographs is that there is a lot of pollution in the edge images of the photographs. In addition, there are no threshold ranges where plate lines are visible and where there is little pollution. When the gaps are changed, either the plate lines disappear or the pollution becomes too high. Cars7 has a special case. Plate lines cannot be detected with edge detection.