**Sobel Edge Detection**

Introduction

<https://medium.com/datadriveninvestor/understanding-edge-detection-sobel-operator-2aada303b900>

Result

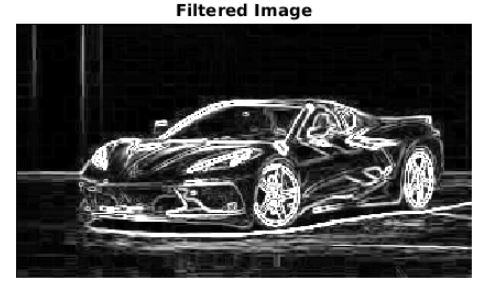
Original image:



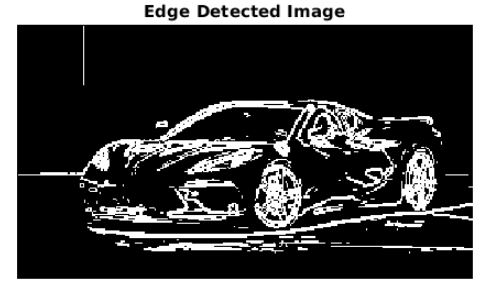
Convert into black and white image:



Filtered image:



Edge detected:



Appendix

% SOBEL OPERATOR EDGE DETECTION PROGRAMMING

% Muhammad Hafizuddin Bin Redzuan

% Afiqa

% Umi

% 23 June 2020

% Read Input Image

original\_image = imread('car.jpg');

figure, imshow(original\_image); title('Original Image');

% Convert into gray image

BW = rgb2gray(original\_image);

figure, imshow(BW); title('Black & White');

% Convert into double before can use filtering

input\_image = double(BW);

% Sobel Operator Mask

Mx = [-1 0 1; -2 0 2; -1 0 1];

My = [-1 -2 -1; 0 0 0; 1 2 1];

% Edge Detection Process

% When i = 1 and j = 1, then filtered\_image pixel

% position will be filtered\_image(2, 2)

% The mask is of 3x3, so we need to traverse

% to filtered\_image(size(input\_image, 1) - 2

% size(input\_image, 2) - 2)

% Thus we are not considering the borders.

for i = 1:size(input\_image, 1) - 2

for j = 1:size(input\_image, 2) - 2

% Gradient approximations

Gx = sum(sum(Mx.\*input\_image(i:i+2, j:j+2)));

Gy = sum(sum(My.\*input\_image(i:i+2, j:j+2)));

% Calculate magnitude of vector

filtered\_image(i+1, j+1) = sqrt(Gx.^2 + Gy.^2);

end

end

% Displaying Filtered Image

filtered\_image = uint8(filtered\_image);

figure, imshow(filtered\_image); title('Filtered Image');

% Define a threshold value

thresholdValue = 100; % varies between [0 255]

output\_image = max(filtered\_image, thresholdValue);

output\_image(output\_image == round(thresholdValue)) = 0;

% Displaying Output Image

output\_image = im2bw(output\_image);

figure, imshow(output\_image); title('Edge Detected Image');