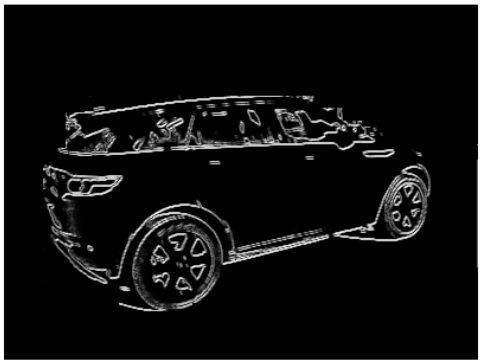
Exercise 1

```
f = imread('P1030975.jpg');
f = imresize(f, [300 400]);

f(:,:,2) = 0;
f = mat2gray(rgb2gray(f));
f = 1./(1+(0.289./f).^30);

wSv = [-1 -2 -1; 0 0 0; 1 2 1];
wS45 = [-2 -1 0; -1 0 1; 0 1 2];
wL = [0 -1 0; -1 4 -1; 0 -1 0];

f = abs(conv_2d(f, wL)) + abs(conv_2d(f, wSv)) +
abs(conv_2d(f, wS45));


% add squares to hide unwanted data
f(:, 1:33) = 0;
f(76:121, 32:52) = 0;
f(253:260, 160:188) = 0;
f(249:254, 168:187) = 0;
f(246:249, 178:190) = 0;
f(1:77, :, :) = 0;
f(260:end,:, :) = 0;
f(223:end, 187:end, :) = 0;
f(1:103, 303:end) = 0;
f(73:98, 287:303) = 0;
f(75:87, 268:289) = 0;
f(102:118, 325:400) = 0;
f(120:131, 368:400) = 0;
f(73:91, 51:68) = 0;
f(235:256, 32:75) = 0;
f(238:255, 68:97) = 0;
f(215:228, 277:317) = 0;

figure, imshow(f, [])
```



Exercise 2

(1) Distance(M,N) is used to create an MxN matrix with each element corresponds to the distance to the center of the matrix.

It is observed that the distance to the center of the matrix is equivalent for a position with that at the position of its complex conjugate. When creating a filter, simply choosing a distance value as threshold, the filter will set those with distance less than threshold to 0, while others remain 1. This can create a low pass filter at ease as frequencies within certain radius will all be removed.

(2) **Code:**

```
function [ g ] = lpfilter_2( f, r )

 [M, N] = size( f ) ;

F = fft2(double( f )) ;
Ds = distance(M, N) ;
Hlp = (Ds<=r) ;

Lp = 1 - Hlp;
Lp(1,1) = 0;
G = F.*Lp;
g = ifft2(G);
```


```
figure, imshow(g,[ ])

end
```

Observation:

Parameter 'r' is for choosing radius: the frequencies with r radius to the center shall be removed in the low pass filter.