Result

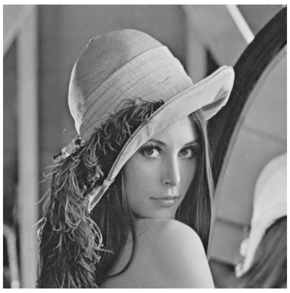

Figure 1: original figure

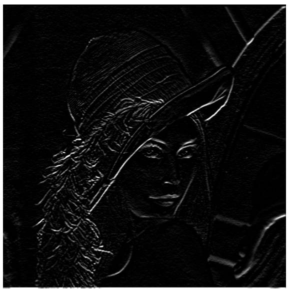

Figure 2: horizon prewitt filter

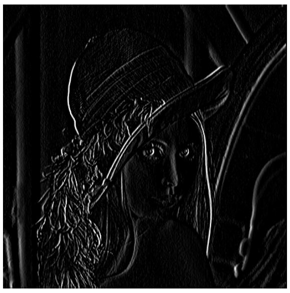

Figure 3: vertical prewitt filter

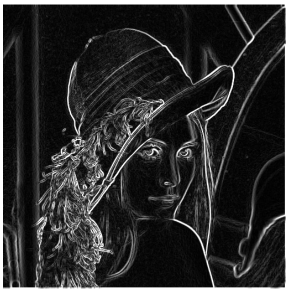

Figure 4: grad norm prewitt filter

Resource Code

```
clc
img = imread('lena.png'); % load original image
img = rgb2gray(img);
figure()
imshow(img);
[hight, width] = size(img);

k_shape = [3, 3]; % set kernel size
padding = (k_shape(1)-1) / 2; % padding size

% generate prewitt kernel
h_kernel = [1,1,1;0,0,0;-1,-1,-1]; % horizon prewitt kernel
v_kernel = [-1,0,1;-1,0,1;-1,0,1]; % vertical prewitt kernel

h_img = zeros(hight + padding*2, width + padding*2);
% generate new image
h_img(1+padding:hight+padding, 1+padding:width+padding) = double(img);
new_h = zeros(hight + padding*2, width + padding*2);
```

```
for i = (1+padding) : (hight+padding)
    for j = (1+padding) : (width+padding)
        % execute horizon filter
        new_h(i, j) = sum(sum(h_img(i-padding:i+padding, j-padding:j+padding).*h_kernel));
    end
end
new_h = uint8(new_h(1+padding:hight+padding, 1+padding:width+padding));
figure();
imshow(new_h);

v_img = zeros(hight + padding*2, width + padding*2);
% generate new image
v_img(1+padding:hight+padding, 1+padding:width+padding) = double(img);
new_v = zeros(hight + padding*2, width + padding*2);

for i = (1+padding) : (hight+padding)
    for j = (1+padding) : (width+padding)
        % execute vertical filter
        new_v(i, j) = sum(sum(v_img(i-padding:i+padding, j-padding:j+padding).*v_kernel));
    end
end
new_v = uint8(new_v(1+padding:hight+padding, 1+padding:width+padding));
figure();
imshow(new_v);

norm_img = zeros(hight + padding*2, width + padding*2);
% generate new image
norm_img(1+padding:hight+padding, 1+padding:width+padding) = double(img);
new_norm = zeros(hight + padding*2, width + padding*2);

for i = (1+padding) : (hight+padding)
    for j = (1+padding) : (width+padding)
        % execute grad_norm filter
        h_norm = sum(sum(norm_img(i-padding:i+padding,j-padding:j+padding).*h_kernel))^2;
        v_norm = sum(sum(norm_img(i-padding:i+padding,j-padding:j+padding).*v_kernel))^2;
        new_norm(i, j) = sqrt(h_norm + v_norm);
    end
end
new_norm = uint8(new_norm(1+padding:hight+padding, 1+padding:width+padding));
figure();
imshow(new_norm);
```