1)

a) The following kernel comes from putting a one In the (-1, 2) position to shift the image left one pixel and up 2 pixels

0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

This is separable with the vectors as follows

0	1	0	0	0

1
0
0
0
0

b)

-1	-1	-1
-1	8	-1
-1	-1	-1

This is similar to the Laplacian filter.

c)

i) If we convolve as is then the image will get high numbers and the image becomes really white/bright. We need to normalize the image.

ii) Using the formula for convolution we get n^2 from going over every pixel of the image and 9 to represent multiple each element in the kernel and 8 to represent the number of additions for each pixel of the image.

<u>Multiple</u>: $9n^2$
<u>Add:</u> $8n^2$

iii)

1	2	1

1
2
1

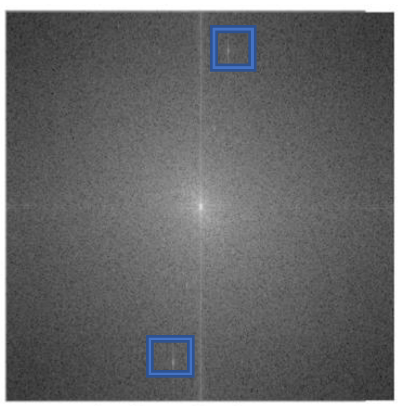
iv) Multiplication: $6n^2$
Add: $4n^2$

v) **B.)** Multiplication: pqn^2
Add: $(pq-1)n^2$

D.) Multiplication: $qn^2 + pn^2$
Add: $(q-1)n^2 + (p-1)n^2$

2)

a)



log Power Spectrum

b) They forgot to do non-max suppression on the image to get rid of the white spots. So what I would do is pick a pixel as corner if cornerness at patch centered on it > cornerness of neighboring pixels. And if cornerness exceeds a threshold, then mark it. Do the same for neighboring pixels of a lower threshold.

c) 1: B
2: D
3: C
4. A

3)

a) $dF/dx = F[x+h, y] – F[x-h, y]$
$= F[x+1, y] – F[x-1, y]$

d^2F/dx^2 = = F[x+2, y] – F[x,y] – F[x, y] – F[x - 2] = F[x-2] + F[x+2] - 2F[x]

1		-2		1

Rest filled with zeros

b) Blur with the gaussian to get rid of high frequencies and reduce noise. We need to do before edge detection in order to reduce the level of noise in the image, which improves the result of the following edge-detection algorithm.

c)

i) $\lambda_{max} >> \lambda_{min}$

ii) The eigenvector are perpendicular to each other. λ_{max} is perpendicular to the edge and λ_{min} is parallel to the edge.

iii) The eigen values are multiplied by α^2 and the eigenvectors do not change.

4)

a)

i) F

ii) T

iii) F

iv) T

b)

i) Brightness is related to the b because adding a constant increase. The Ia represents the contrast because by multiplying the scale the Image we change the contrast between the old image pixel and the new one.

ii) Harris corner detector (both locations and probability of corner detection) is invariant to additive changes in intensity which is changes in overall "Brightness". It is not invariant to scaling of intensity or changes in "Contrast". What was one patch earlier is now many. Because in the moment matrix what u do is $M' = a^2 M$, so which doesn't change the brightness, but rather the contrast. Another way to look at this is that after adding the b the gradient doesn't change versus when you multiple the gradient dues which means this is not contrast invariant.

iii) Scale invariance by using scale identified by corner detector. Intensity normalize the window by subtracting the mean, dividing by the standard deviation in the window

5)

a) We know that

$N_xX + N_yY + N_zZ = d$

Plug our point that we know that's on the plane we get:

$N_x(D_x/D_z)+ N_yY(D_y/D_z)+ N_z = d/Z$
If we take the limit as Z → infinity
We Get $N_x x + N_y y + N_z = 0$
We use the fact that vanishing point of line is DX/dz, Dy/dz and we know vanishing line equation so from that we know that the vanishing point is on the vanishing line of the plane

b)

i) F
ii) T
iii) F
iv) F