Computer Vision 1
Assignment 2

Nichita Diaconu (11737980), Radu Sibechi(11808527)

1

2 Neighborhood Processing

Question 1

1.They are both linear operators, with between a filter and a signal. The difference is that the convolution rotates the image 180 degrees. Also, correlation is used as a measure of similarity between two signals. Moreover, the convolution is associative:

$$(F(X) \otimes G(X)) \otimes H(X) = F(x) \otimes (G(X) \otimes H(X))$$

2. Given that the difference is that the convolution rotates the image 180 degrees. As a result the operations are identical when the filter is symmetric.

3 Low-level filters

Question 2

$$Gaussian1D\_Xaxis \otimes (Gaussian1D\_Yaxis \otimes Image) = (Gaussian1D\_Xaxis \otimes Gaussian1D\_Yaxis) \otimes Image$$
$$= Gaussian2D \otimes Image$$

Given that an image is size MxN and a filter kxk. There is no difference in terms of how the output looks when we convolve a 1D Gaussian filter on the x axis and then on the y axis or if we convolve a 2D Gaussian filter. In terms of complexity, computing a 2D convolution takes $O(MNkk$ and applying the two 1D convolutions takes $O(2MNk)$

Question 3

In order to design a more meaningful filter. One that could tell us what is the rate of change of the change in the values of the pixels. As a result the Laplace filter detects edges in an image, regardless of orientation.

Question 4

λ represents the wavelength of the sinusoidal factor.

θ represents the orientation of the normal to the parallel stripes of a Gabor function.

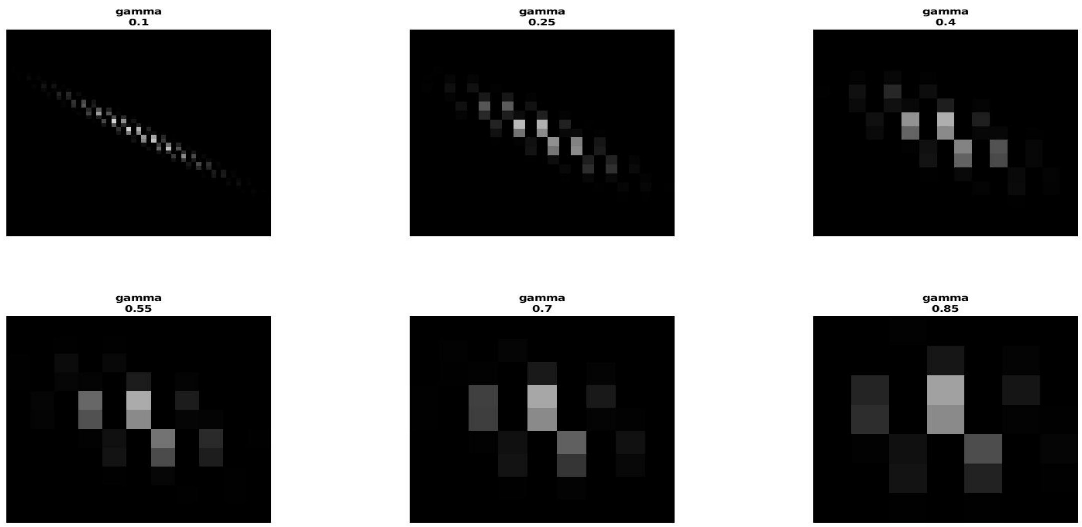
ψ is the phase offset.

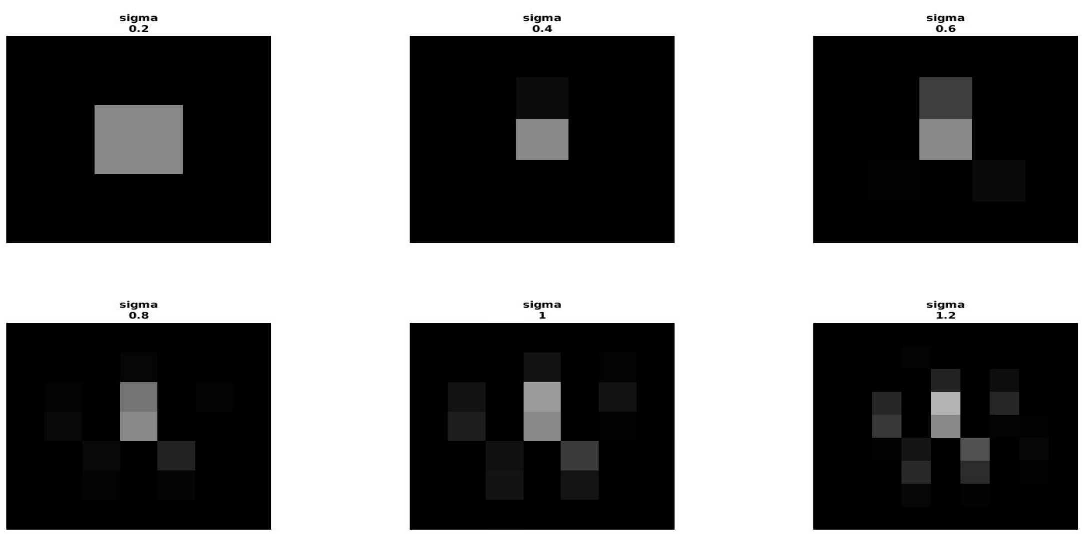
σ is the sigma/standard deviation of the Gaussian envelope.

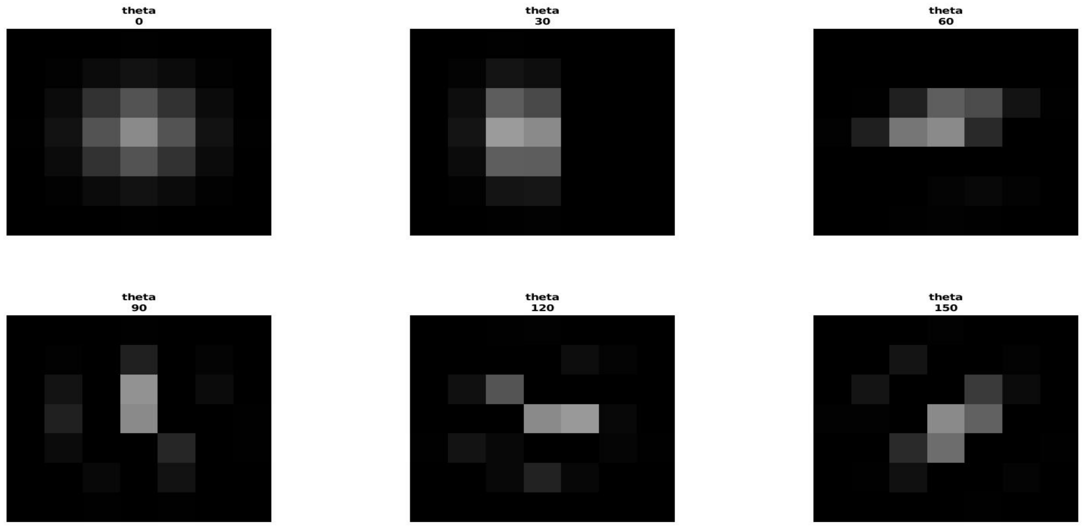
γ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Question 5


gamma
0.1
gamma
0.25
gamma
0.4
gamma
0.55
gamma
0.7
gamma
0.85


sigma
0.2
sigma
0.4
sigma
0.6
sigma
0.8
sigma
1
sigma
1.2



4 Applications in image processing

Question 6

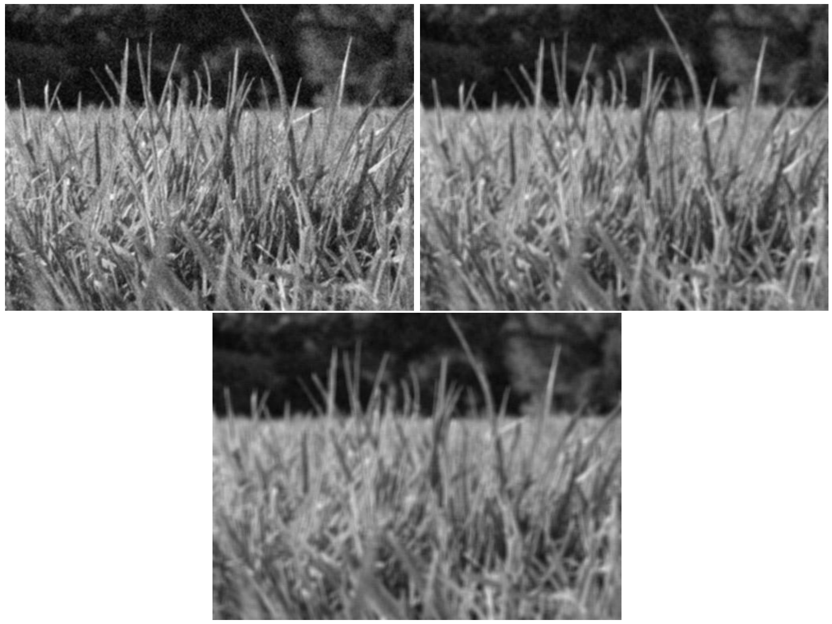
1. PSNR(saltpepepr) = 16.1079
2. PSNR(gaussian) = 20.5835

4.1 Question 7

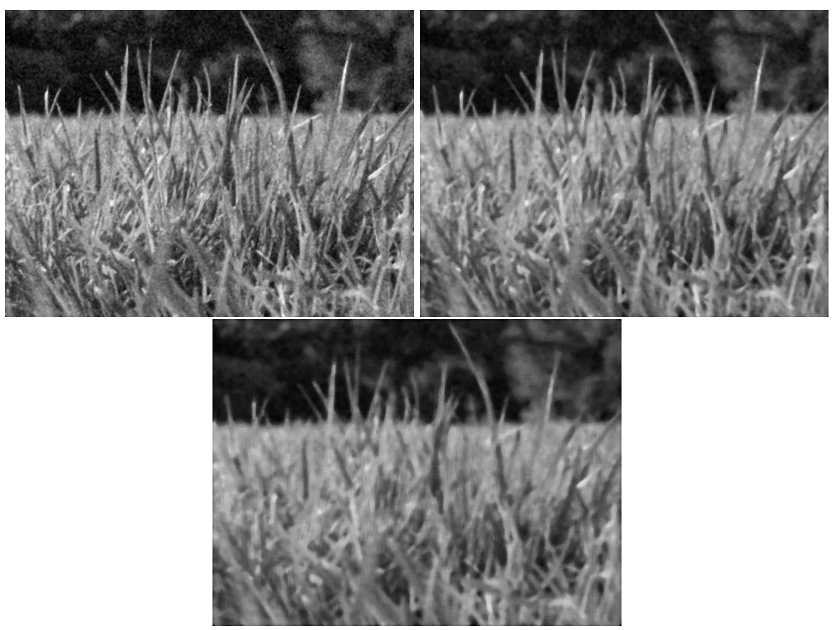
1.

- Gaussian noise

  – Box filter (size 3, 5, 7 from left to right)

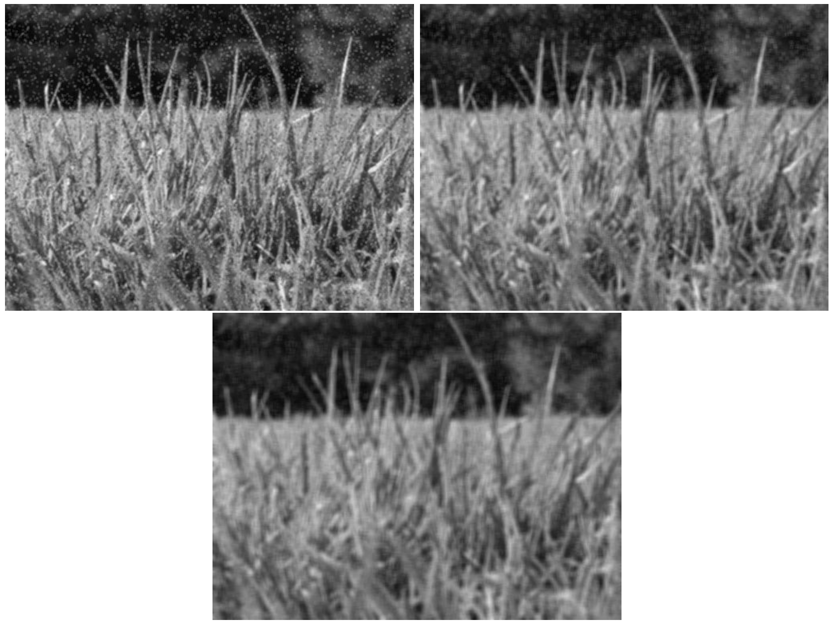


  - Median filter (size 3, 5, 7 from left to right)

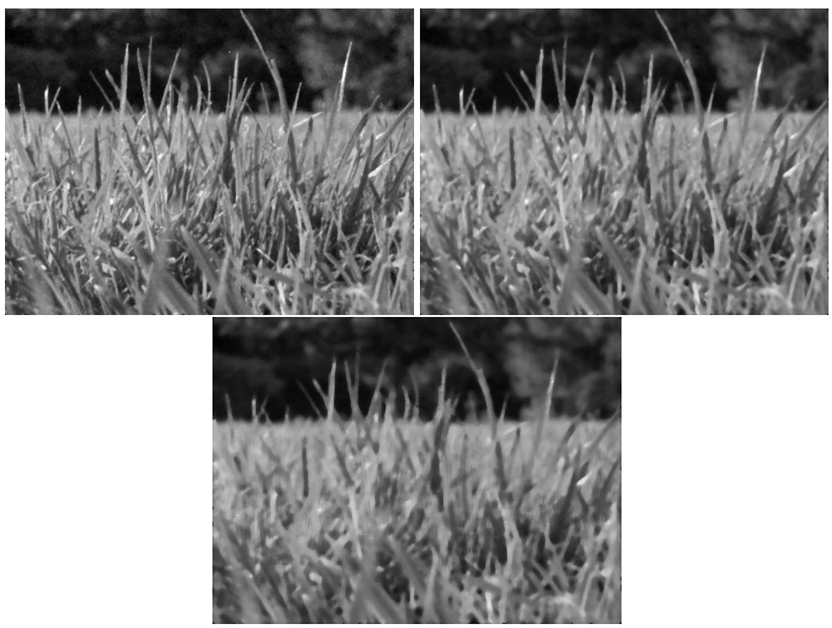


- Salt & Pepper noise

  - Box filter (size 3, 5, 7 from left to right)



- Median filter (size 3, 5, 7 from left to right)



2.

PSNR score	image noise and filter
26.2422	image1_gaussian_noise_box-filter_size=3
23.0343	image1_gaussian_noise_box-filter_size=5
20.6206	image1_gaussian_noise_box-filter_size=7
25.6087	image1_gaussian_noise_median-filter_size=3
23.5194	image1_gaussian_noise_median-filter_size=5
21.1126	image1_gaussian_noise_median-filter_size=7
23.4107	image1_saltpepper_noise_box-filter_size=3
21.9048	image1_saltpepper_noise_box-filter_size=5
19.9013	image1_saltpepper_noise_box-filter_size=7
27.8201	image1_saltpepper_noise_median-filter_size=3
24.5290	image1_saltpepper_noise_median-filter_size=5
21.6990	image1_saltpepper_noise_median-filter_size=7

3. The box filter is actually a mean filter over a box with a specific size. A median filter takes the median in a box with a specific size. In my opinion, in both cases, the median filter of window size 3 is the best filter for denoising (just by human eye, later I explain gaussian is better and even later we show PSNR scores). Even though both filters smooth the image. In the case of salt and pepper noise, the median filter of size 3 smooths the added noise, with minimal effect on the other pixels, due to the window size. In the case of the Gaussian noise, even though the NOISE is removed more by the Gaussian filter, the image is further smoothed a lot and becomes more unclear than the image with median filter, but this preserves the Gaussian noise a little more. In both cases I find the median filter of size 3 to make the image clearer.
4.

- Gaussian noise
  - Window size 3 (sigma 0.5, 1, 2 from left to right)



  - Window size 5 (sigma 0.5, 1, 2 from left to right)



  - Window size 7 (sigma 0.5, 1, 2 from left to right)



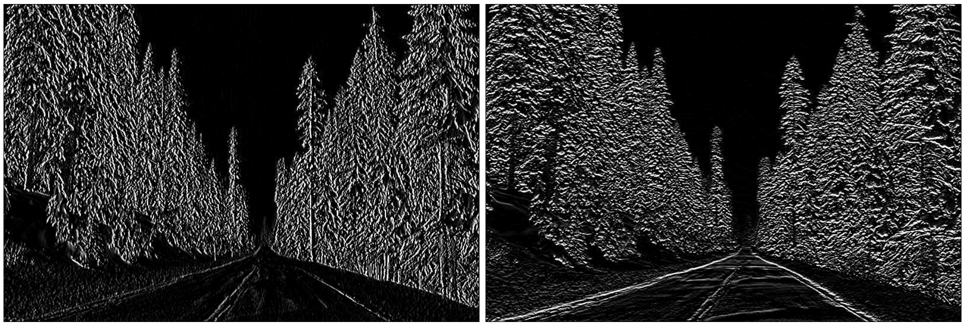
I would say that the best filter is the Gaussian filter with window size 5 (3 can be fine, while 7 blurs image too much), and sigma 1 (0.5 keeps too much noise, while 2 keeps too little image edges). But I don't trust my intuition, so I test it with PSNR:
5.

PSNR score	image noise and filter
22.0874	image1_gaussian_noise_gaussian-filter_sigma=0.5_kernel_size=3
21.2550	image1_gaussian_noise_gaussian-filter_sigma=0.5_kernel_size=5
20.5439	image1_gaussian_noise_gaussian-filter_sigma=0.5_kernel_size=7
24.4246	image1_gaussian_noise_gaussian-filter_sigma=1_kernel_size=3
22.5187	image1_gaussian_noise_gaussian-filter_sigma=1_kernel_size=5
21.4796	image1_gaussian_noise_gaussian-filter_sigma=1_kernel_size=7
24.3116	image1_gaussian_noise_gaussian-filter_sigma=2_kernel_size=3
21.4371	image1_gaussian_noise_gaussian-filter_sigma=2_kernel_size=5
19.6662	image1_gaussian_noise_gaussian-filter_sigma=2_kernel_size=7

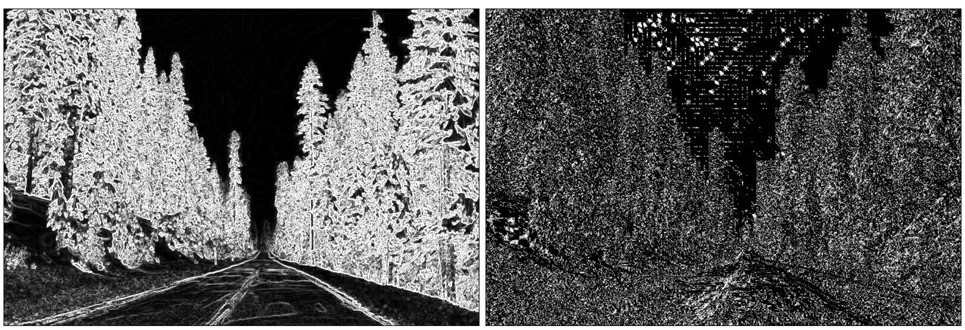
For each kernel size, the PSNR values increase when sigma increases from 0.5 to 1 and then decreases when we increase sigma to 2. That is why we picked sigma 1 as the best sigma.
6. Two filtering methods give around the same PSNR when they have removed the respective noise in each image to the same degree. We could say that median filter with size 3 removes salt & pepper noise almost as well as the box filter with size 3 removes the gaussian noise. PSNR gaussian noise box filter size 3 = 26.2422; PSNR salt & pepper noise median filter size 3 =27.82

4.2 Question 8

1. & 2. (left to right)



3. & 4. (left to right)



4.3 Question 9

1. Test your function using image2.jpg and visualize your results using the three methods. 2. Discuss the difference between applying the three methods. 3. In the first method, why is it important to convolve an image with a Gaussian before convolving with a Laplacian? 4. In the third method, what is the best ratio between sigma 1 and sigma 2 to achieve the best approximation of the LoG? 5. Do you notice a visible difference with the gradient magnitude of the first-order method? 6. What else is needed to improve the performance and isolate the road?

1.

Method 1 & 2 (from left to right) & 3 (below):



2.For the 1st and second methods we should get good identical results, unfortunately due to using the built in log we found a image with higher magnitude on edges. On the other hand, method 3 performs a little worse than 1, as expected because it is only an approximation.
3. Because it is important to smooth before taking the derivatives to detect edges. This way we remove the noise first and then find the edges.
4. In the third method we found that the greater the ratio the better and past $\tilde{1}00$ it does not make a difference anymore.
5. Yes there is a clear strong difference. Also it seems that the 1st order Y derivative does a much better job at detecting the road.
6. In order to further isolate the road, more complex feature that would be specific to the road would be necessary or some method that would be depth invariant.

4.4 Question 10

1. We can observe that the image is partitioned into 2 segments. As such, the main object in the pictures are located and boundaries around them are added.

2. We have experimented with various parameters in order to achieve good results.

As such for 'Kobi', a resize factor of 0.3, lambda factor of 8, sigma of 27, theta of 0:pi/8:(3*pi/4)).

For 'Polar' a resize factor of 0.75, a lambda factor of 4.5, sigma [1,2], and theta of 0:pi/4:(3*pi/4).

For 'Robin-2', a resize-factor of 0.5, lambda factor 3.5, sigma [1,2] and theta 0:pi/8:(7*pi/8).

For 'Cows', a resize-factor of 0.5, lambda factor of 4, sigma [4,10] and theta 0:pi/4:(3*pi/4)

3. The objects are no longer properly segmented. We always have to smooth in order to remove the noise and because of the unremoved noise now the segmentation no longer properly works.