University of Tehran

COMPUTER SCIENCE DEPARTMENT

COMPUTATIONAL NEUROSCIENCE

REPORT 6

DOG and GABOR FILTER

Student Name	Student ID
AMIR NADERI	610398126

Lecturer in charge:

M.Ganjtabesh

Submission Date : 30/06/2023

1 Visual Cortex

The visual cortex of the brain is the area of the cerebral cortex that processes visual information. Sensory input originating from the eyes(Retina) travels through the lateral geniculate nucleus(LGN) in the thalamus and then reaches the visual cortex. The area of the visual cortex that receives the sensory input from the LGN is the primary visual cortex, also known as V1. The rest of areas consist of V2, V3, V4, and IT. Both hemispheres of the brain include a visual cortex; the visual cortex in the left hemisphere receives signals from the right visual field, and the visual cortex in the right hemisphere receives signals from the left visual field.
This is the picture of visual cortex of the brain:

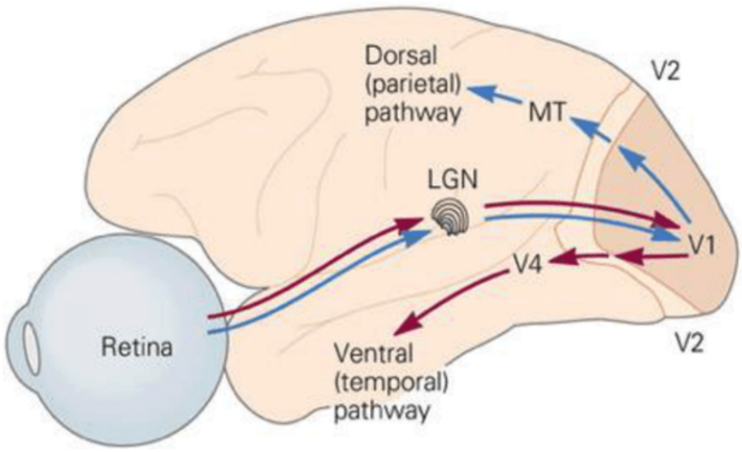


Figure 1: Visual Cortex

As you see V1 transmits information to two primary pathways, called the ventral pathway and the dorsal pathway. The ventral pathway begins with V1, goes through visual area V2, then through visual area V4, and to the inferior temporal cortex (IT). The ventral pathway, sometimes called the "What Pathway", is associated with form recognition and object representation. It is also associated with storage of long-term memory. The dorsal pathway begins with V1, goes through Visual area V2, then to the medial temporal area (MT). The dorsal pathway, sometimes called the "Where Pathway", is associated with motion, representation of object locations, and control of the eyes and arms, especially when visual information is used to guide saccades or reaching.
If we want to go through details we first have to know about retina organization. First let's see a picture of retina:

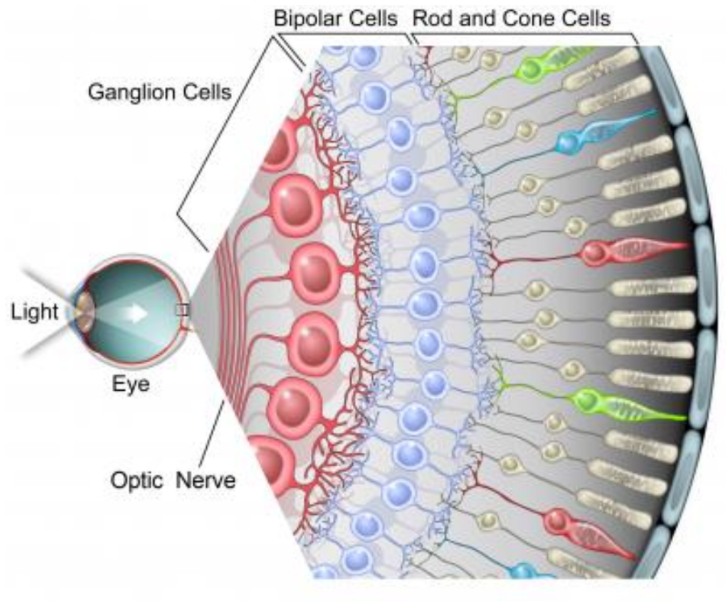


Figure 2: Retina Organization

Well, As you see the light enters the eye and goes to the retina. Retina gets the light and processes it and then transmits the information through optic nerves. The retina itself consists of some cells that process the information:

1. Photoreceptors: Photoreceptors are specialized neurons found in the retina that convert light into electrical signals that stimulate physiological processes. We have two kinds of photoreceptors:
   - Rod:
     - Scotopic (dark-adapted)
     - High sensitivity
     - Low spatial resolution
     - Achromatic (release rhodopsin)
   - Cone:
     - Photopic (light-adapted)
     - Lower sensitivity
     - High resolution
     - Chromatic (release chromatin: Red, Green, and Blue cones)
2. Bipolar Cell: Responds to the release of glutamate from photoreceptors with graded potentials (i.e., by hyperpolarizing or depolarizing). There are two types of bipolar cells:
   - On-type bipolar cells:
     - Hyperpolarized by glutamate
     - Detect light points in a darker background
     - Depolarizes when photoreceptors are in the light
   - Off-type bipolar cells:
     - Depolarized by glutamate

– Detect dark points in a lighter background
– Depolarizes when photoreceptors are in the dark

3. Ganglion cells: The retinal ganglion cells are the final retinal elements in the direct pathway from the eye to the brain. The retinal ganglion cells that synapse with on bipolar cells will have "on-center-off-surround" receptive fields. Also The retinal ganglion cells that synapse with off bipolar cells will have "off-center-on-surround" receptive fields. There are two major types of retinal ganglion cells:

   - Parvo-cellular (P cells):
     – they make synaptic contact with one to a few cone bipolar cells in the fovea.
     – they are color sensitive.
     – they have a small receptive field.
     – they Produce weak responses to stimuli that move across its receptive field.
   - Magno-cellular (M cells):
     – they are much larger than P ganglion cells.
     – Synapses with many bipolar cells.
     – they are color-insensitive.
     – they have a large concentric receptive field.
     – they respond maximally to stimuli moving across its receptive field.

We can simulate the ganglion cells using DoG(difference of gaussian) filters. We see this process in next section.
So far we learned that the ganglion cells detect light/dark points in the darker/-lighter background. Now LGN transmits this information to the area V1. This part has two major types of cell. One is V1 simple cell and the other is V1 complex cell. V1 simple cells respond to the lines with different orientations. In other words there are some ganglion cells that respond to points in a particular receptive fields and if we put these cells in a line in a way that they have synapses with another neuron(V1 simple cell), we can present a line. These ganglion cells will activate this V1 simple cell when this line with this orientation is detected. You can see a picture of this process in below:

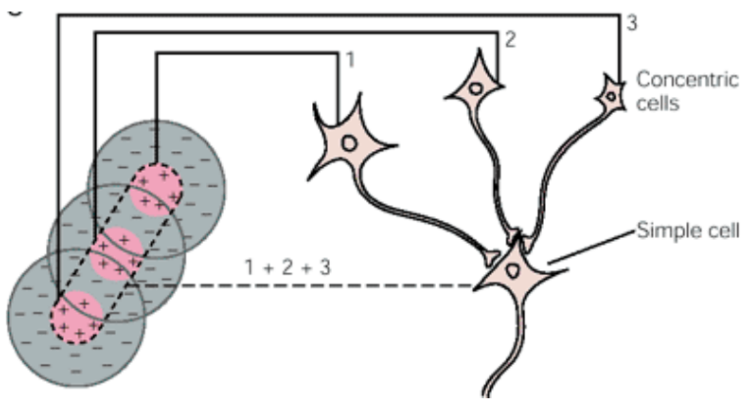


Figure 3: V1 Simple Cell

We can simulate this process using Gabor filters which we'll see in the next section.
So we can detect edges using V1 simple cells. These are some pre-processing of the image that we see using our eyes. In this project we are going to simulate the ganglion and V1 simple cells using DoG and Gabor filters.

2 DoG Filter

As we said before, the DoG filter simulates ganglion cells which detect points in a background with different color. Difference of Gaussian(DoG) is calculated as the difference between two smoothed versions of an image obtained by applying two Gaussian kernels of different standard deviations on that image. In other words, DoG transformation of an image requires subtracting one highly blurred version of an original image from another less blurred version to preserves a specific spatial frequency. DoG removes high-frequency spatial components representing noise in the image through the blurring and low-frequency components. DoG generally serves as an edge enhancement algorithm that delineates the high-frequency content of the image free from noise. The formula of DoG in 2D is:

$$DoG_{\sigma_1,\sigma_2}(X,Y) = \frac{1}{\sqrt{2\pi}}\left(\frac{1}{\sigma_1}e^{\frac{-(x^2+y^2)}{2\sigma_1^2}} - \frac{1}{\sigma_2}e^{\frac{-(x^2+y^2)}{2\sigma_2^2}}\right)$$

We can see an example of 1D difference of gaussian in below:

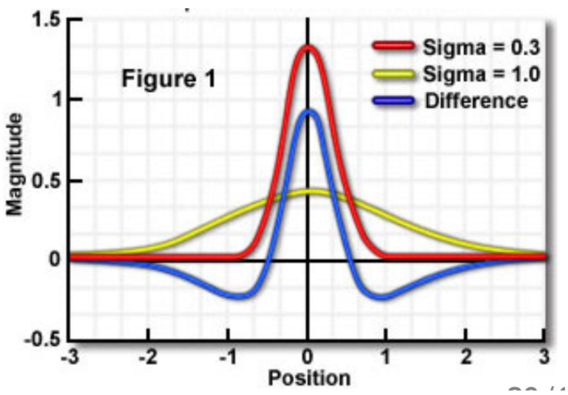


Figure 4: 1D difference of gaussian

The volume under each Gaussian is first normalized to one, so that their difference has a mean of zero. So when we convolve this filter on a solid region of an image, the result should be zero, because we have no points in a background with different color. Now let's see an example of DoG filter and convolving it on an image:

2.1 Ex 1

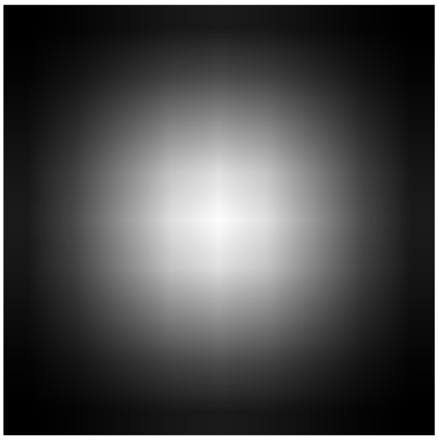
Our original image is:

Figure 5: Original Image

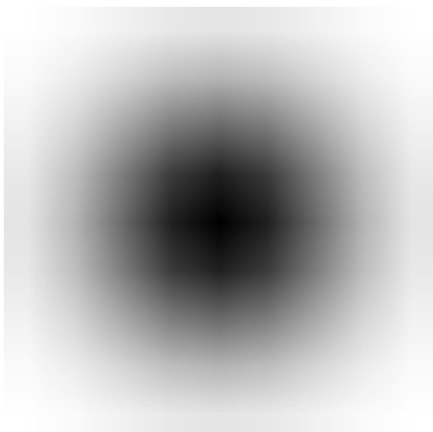
We read this image as black and white way. So the shape of it is 2D. The parameters chose for filter in this part are:

- sigma1 = 2
- sigma2 = 9
- kernel size = 9

The resulting image of this filter is:



(a) On-center-off-surround DoG Filter



(b) Off-center-on-surround DoG Filter

Figure 6: DoG Filters

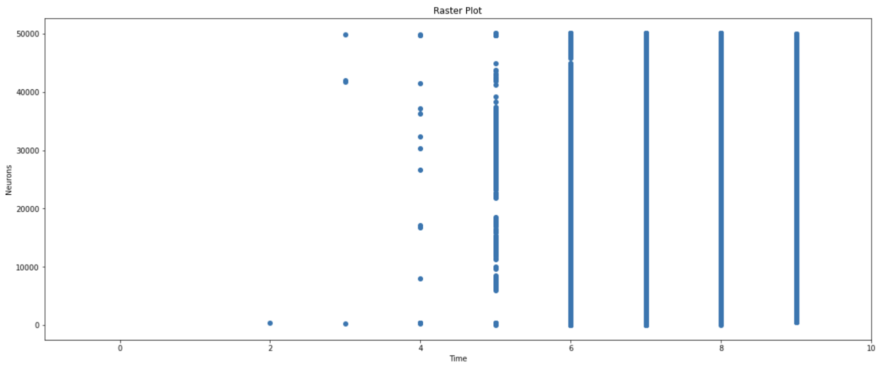
As you see in (a), this is on-center-off-surround filter which we have positive values in the middle of filter and negative values in the surround. We have the opposite of this matrix for off-center-on-surround that you can see in the part (b). Now we convolve these filters on the image, the result is:

(a) On-center-off-surround

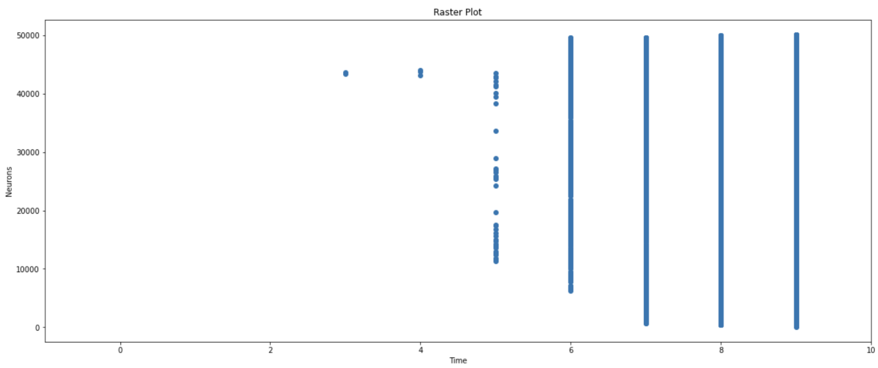
(b) Off-center-on-surround

Figure 7: Convolved Image

As you see by convolving this filter on the image, we detect edges and remove noises. Now let's see the time to first spike (TTFS) encoding for these convolved images:



(a) On-center-off-surround



(b) Off-center-on-surround

Figure 8: TTFS Encoding

TTFS encoding shows the time that neurons spike for their first time and after that they don't spike. Now let's see in each iteration which neurons spike in their corresponding receptive field:

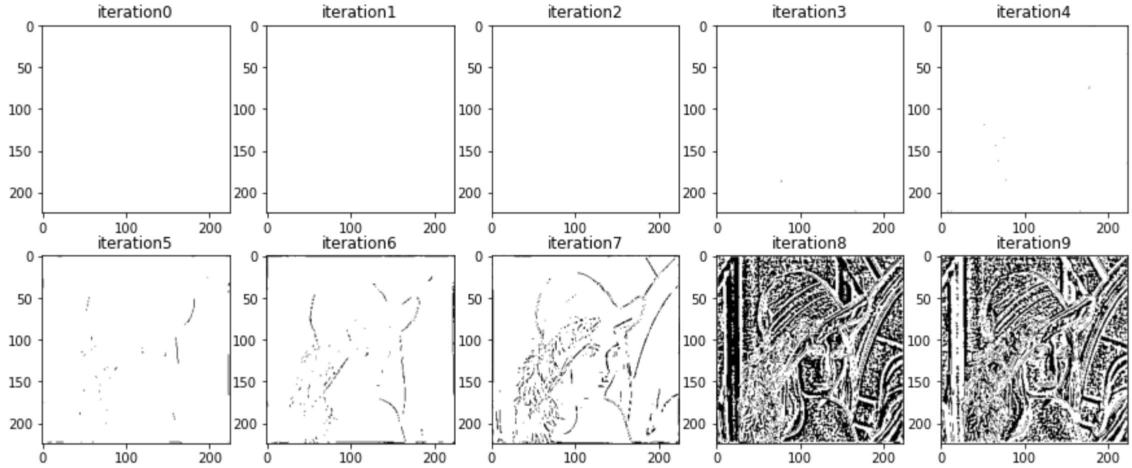


Figure 9: On-center-off-surround neurons spike in each iteration of ttfs encoding

As you see in first iteration neurons don't spike because their value (pixel intensity) are low. after some iterations we see some neurons spike because they have the biggest intensity in pixels and in the iteration 7 we see that the neurons which indicate edges, spike and in iteration 8 and 9 other neuron with less intensity spike as well.
Now let's see the process of ttfs encoding for off-center-on-surround neurons:

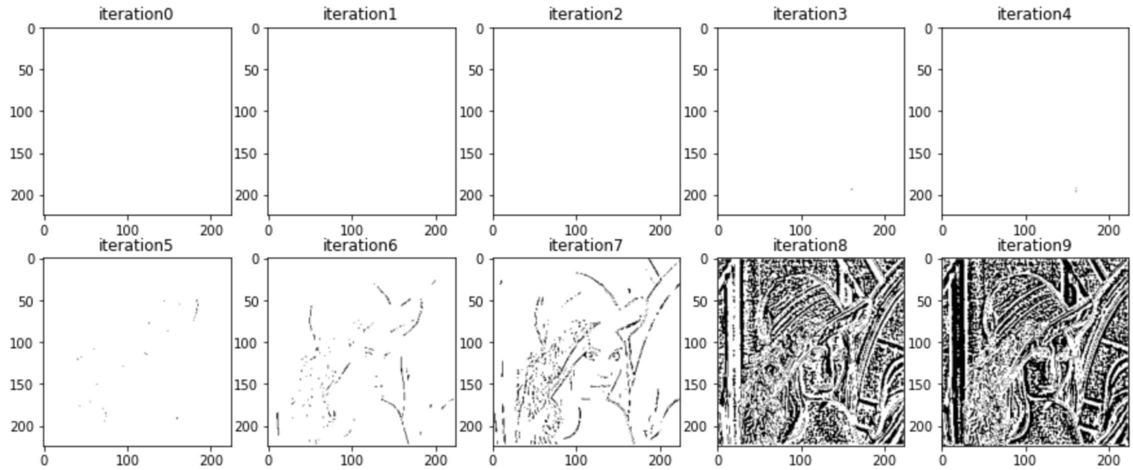


Figure 10: Off-center-on-surround neurons spike in each iteration of ttfs encoding

Well, here the neurons finds darker points in the lighter background, because they are off-center ganglion cells and they connect to the off-center bipolar cells, so they detect dark point in the lighter background, but in figure 9, neurons detect light point in the darker background.
Now check the poisson encoding of convolved image obtained by on-center-off-surround filter:

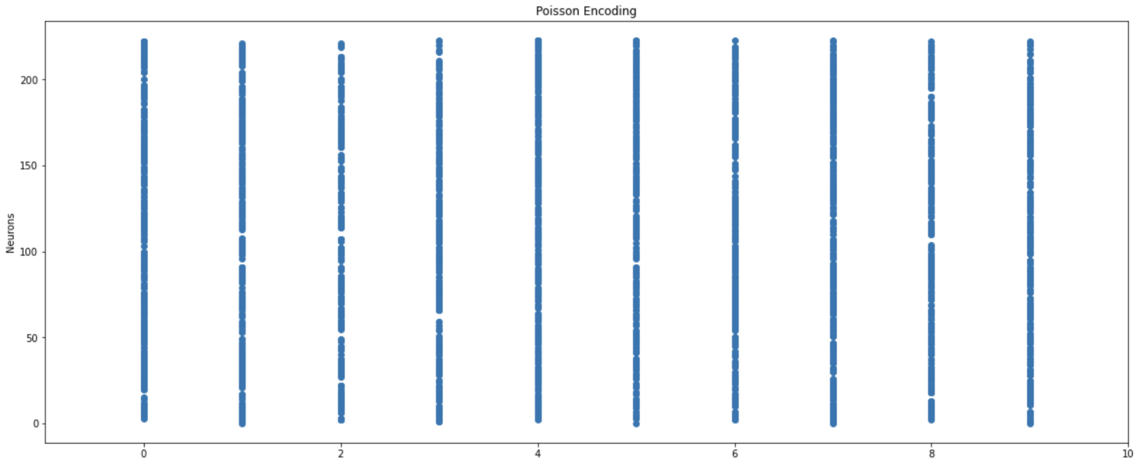
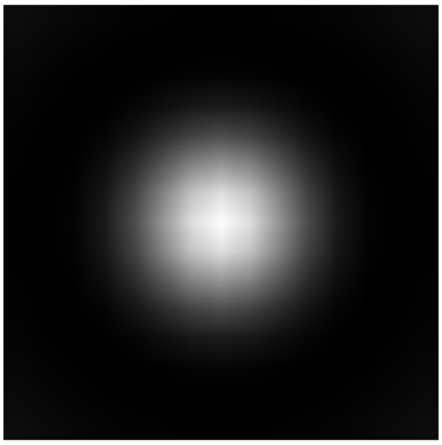


Figure 11: Off-center-on-surround neurons spike in each iteration of poisson encoding

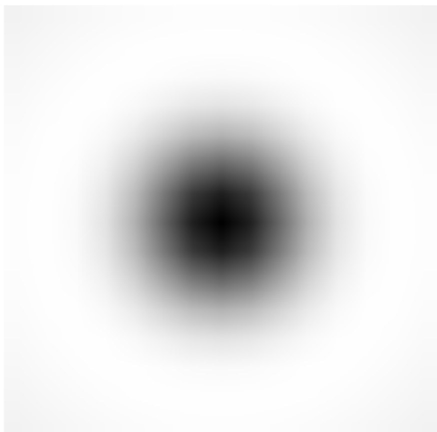
This encoding represent spikes of neurons regarding to their pixel values. It means if a value is larger, the spikes are more.

2.2 Ex 2

Assume the picture we plot in figure 5. In this example we increase the size of kernel to 15 and the values of sigma remains the same. See the picure of filters:



(a) On-center-off-surround DoG Filter



(b) Off-center-on-surround DoG Filter

Figure 12: DoG Filters

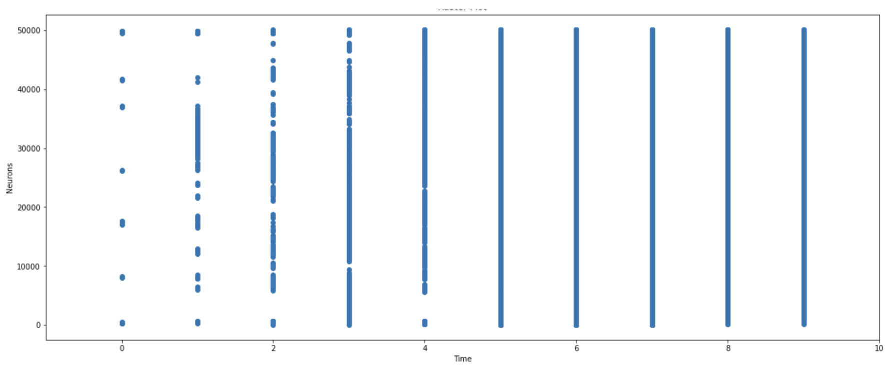
For example in the part (a) when we have on-center filter, the neurons will inhibit more when the surround of the point is darker. Also its receptive field increases. Now let's see the convolved images using these filters:

(a) On-center-off-surround

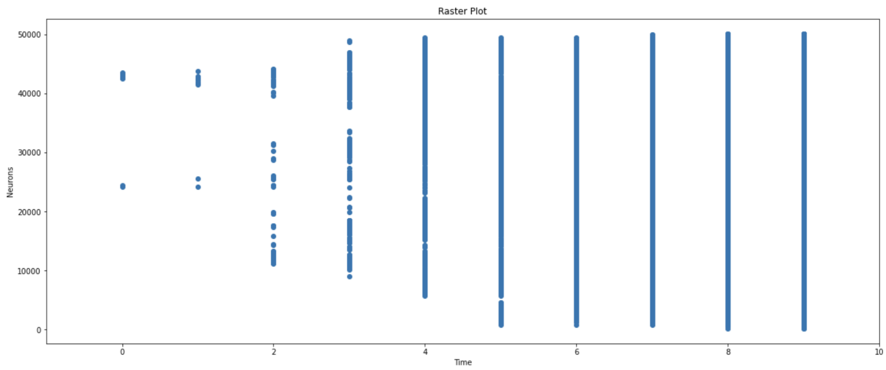
(b) Off-center-on-surround

Figure 13: Convolved Image

As you see we have more blurred images and the pixel intensities increases. So the effect of this filter on TTFS encoding would be:



(a) On-center-off-surround



(b) Off-center-on-surround

Figure 14: TTFS Encoding

Here we have bigger intensity values so some neurons spike on their first iterations but most of them spike at last two iterations. Let's see the steps of the TTFS encoding:

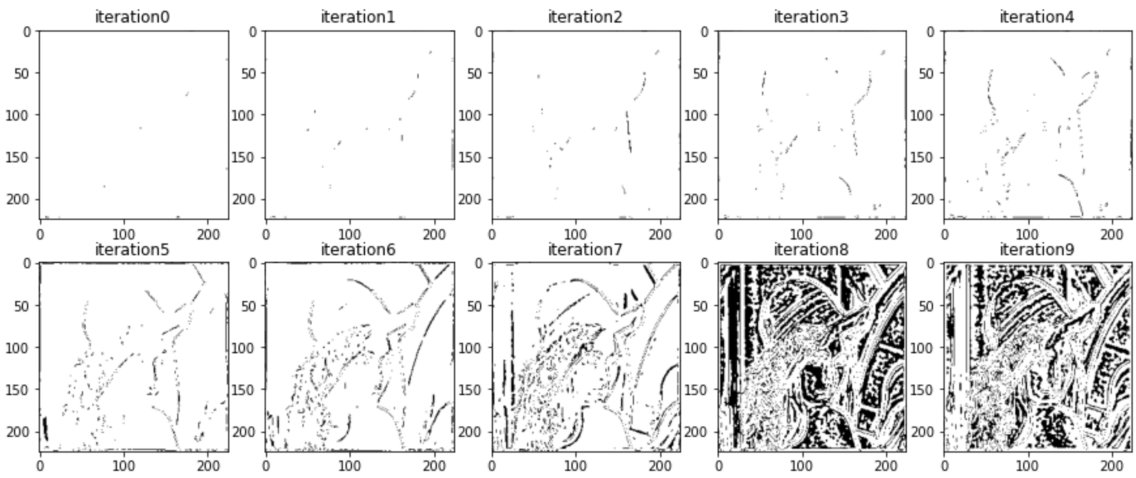


Figure 15: On-center-off-surround neurons spike in each iteration of ttfs encoding

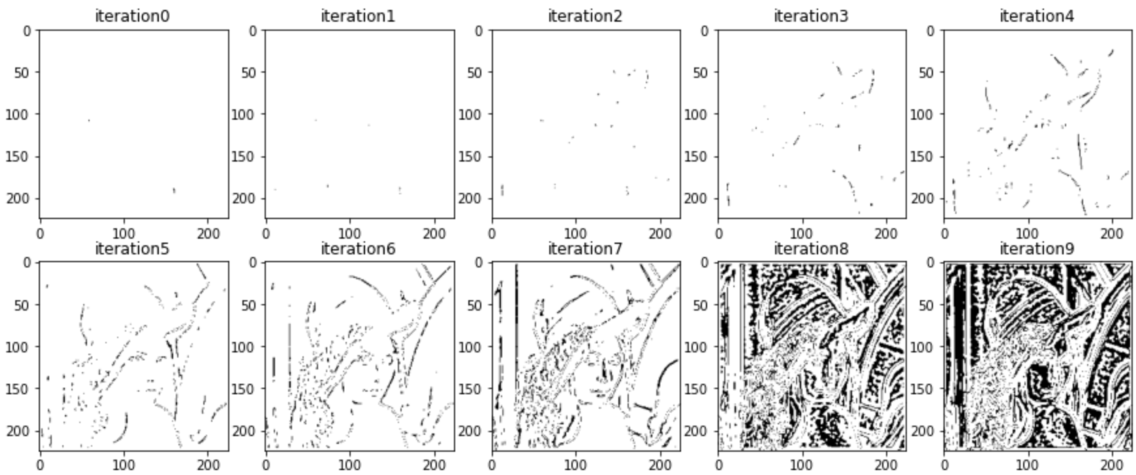


Figure 16: Off-center-on-surround neurons spike in each iteration of ttfs encoding

Now check the poisson encoding of convolved image obtained by on-center-off-surround filter:

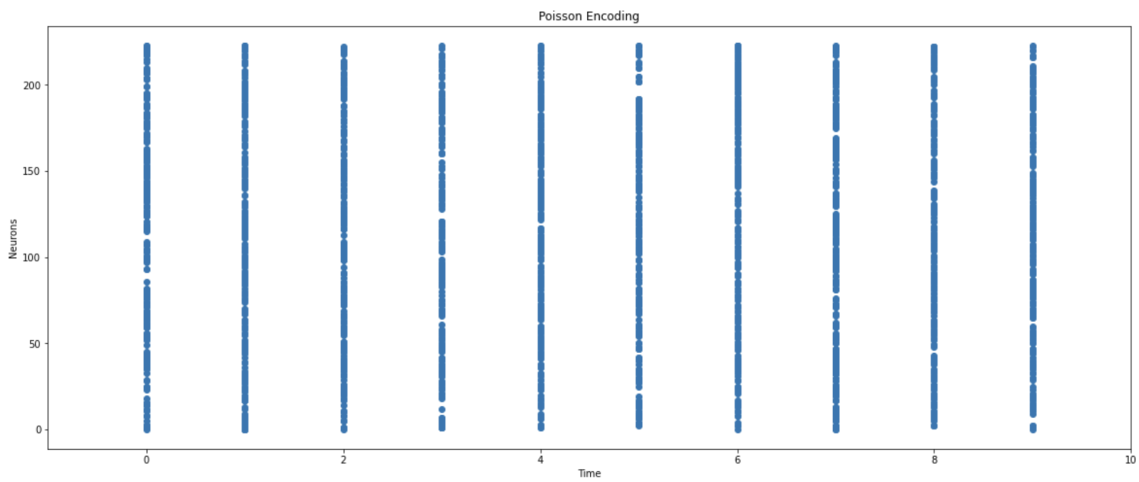


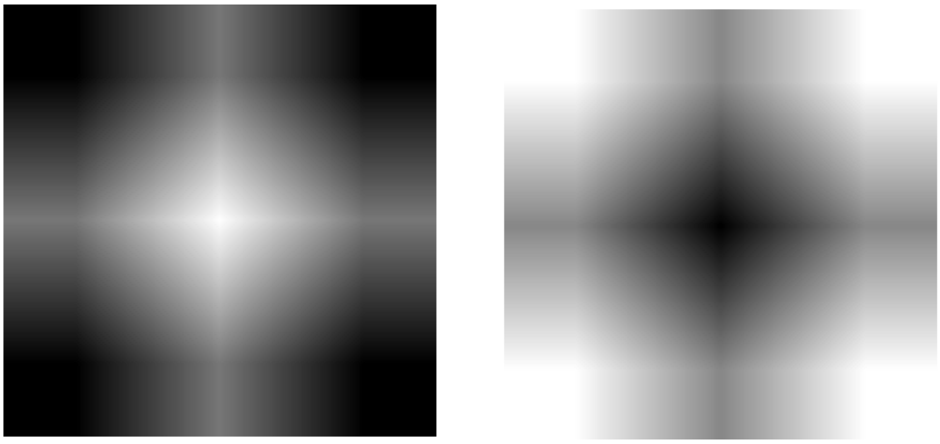
Figure 17: Off-center-on-surround neurons spike in each iteration of poisson encoding

2.3 Ex 3

In this example we decrease the size of the kernel to 3. So the parameters of the filter are:

- sigma1 = 2
- sigma2 = 9
- kernel size = 3

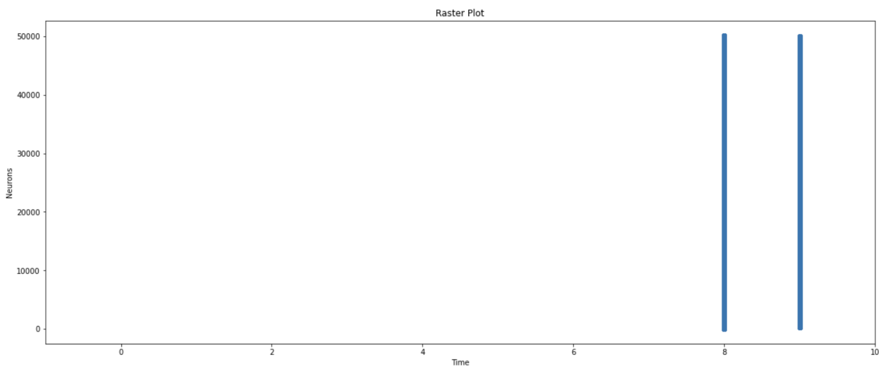
Check the filters:



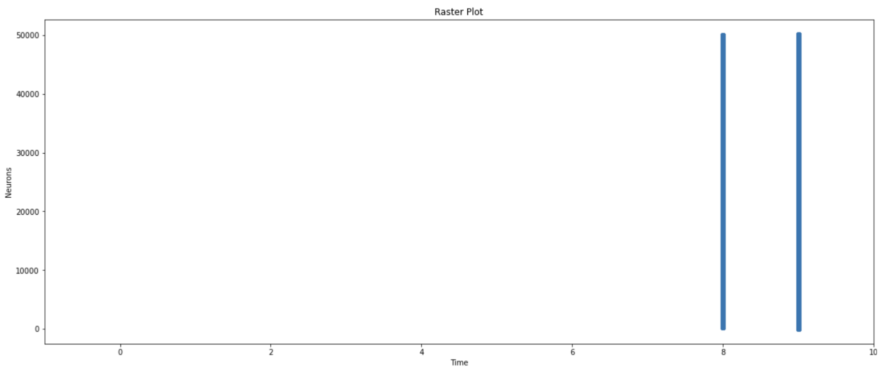
(a) On-center-off-surround DoG Filter (b) Off-center-on-surround DoG Filter

Figure 18: DoG Filters

Remember if we decrease the kernel size to a lower value like 3, the convolved image will have less intensity for their pixels and the receptive field decreases as well, therefore the TTFS encoding would look like this:



(a) On-center-off-surround



(b) Off-center-on-surround

Figure 19: TTFS Encoding

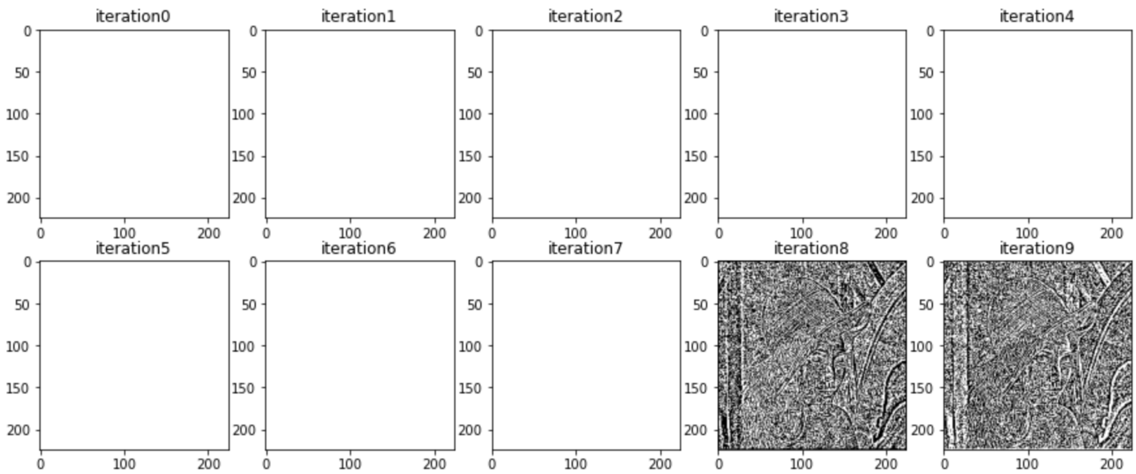


Figure 20: On-center-off-surround neurons spike in each iteration of ttfs encoding

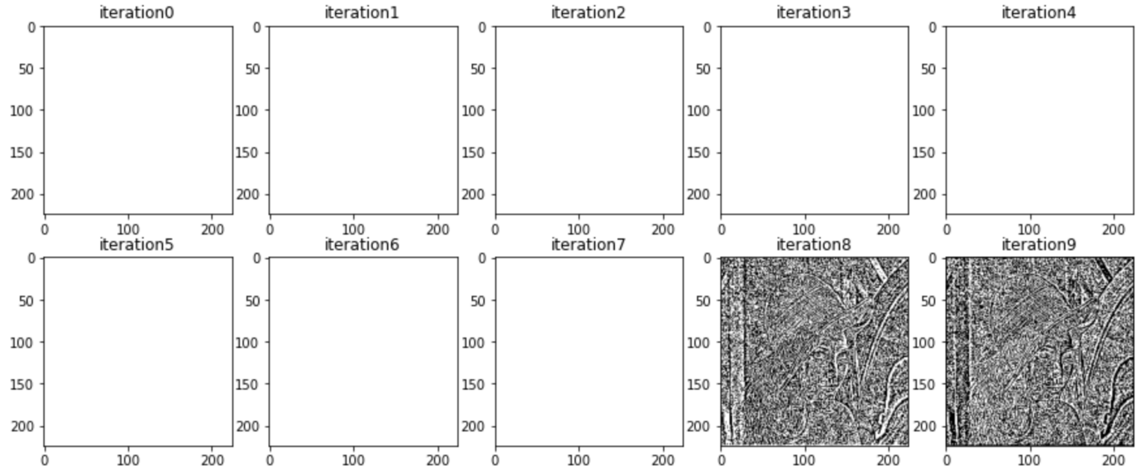


Figure 21: Off-center-on-surround neurons spike in each iteration of ttfs encoding

As you see in figure 20, first the light points in the darker background were detected by their corresponding on-center neurons and in the figure 21, the dark points in the lighter background were detected first. Also see the convolved images obtained by the filters:

(a) On-center-off-surround

(b) Off-center-on-surround

Figure 22: Convolved Image

These images are less blurred and the noises are less more compared to the last examples.
Now check the poisson encoding of convolved image obtained by on-center-off-surround filter:

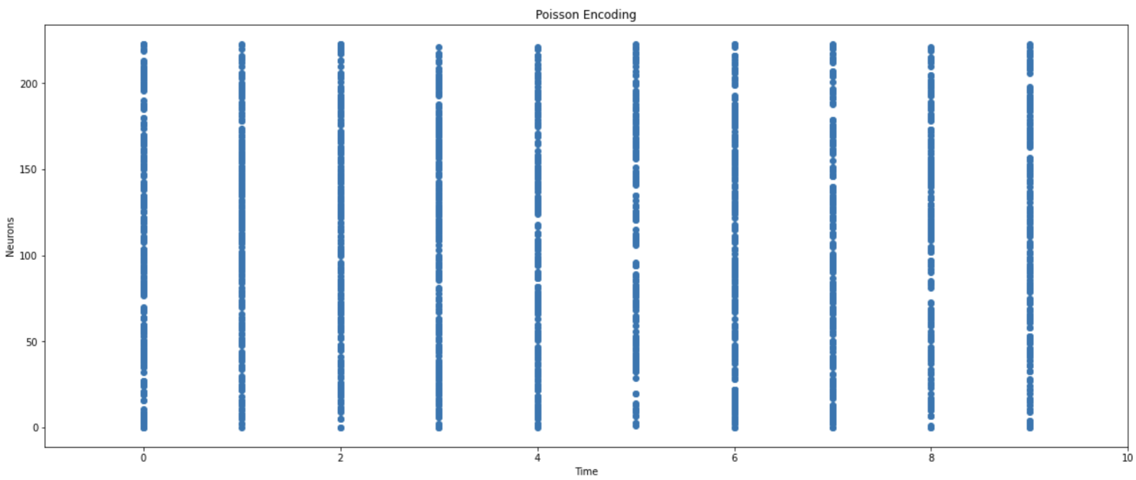


Figure 23: Off-center-on-surround neurons spike in each iteration of poisson encoding

2.4 Ex 4

Now we decrease the size of sigma1 to 0.2. So the parameters are:

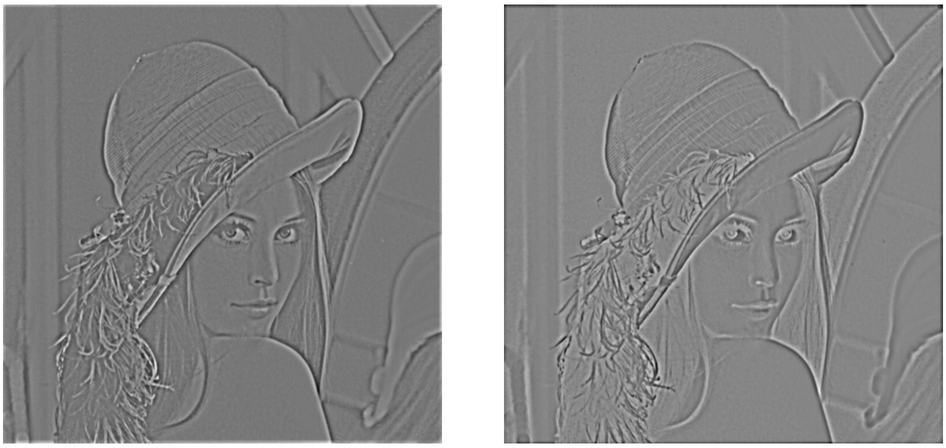
- sigma1 = 0.2
- sigma2 = 9
- kernel size = 9

the difference between two gaussian distribution increases and the filter would look like:

(a) On-center-off-surround DoG Filter (b) Off-center-on-surround DoG Filter

Figure 24: DoG Filters

Here the negative values are more and receptive field didn't change compared to Ex 1. So what do we expect ?
First let's see the convolved images:



(a) On-center-off-surround (b) Off-center-on-surround

Figure 25: Convolved Image

As you see we have less noises in picture. The reason that happened is we have more negative values in the surround and less positive values in the middle (for on-center neurons). So we have greater pixel intensity and therefore the neurons spike sooner in the iteration of TTFS encoding compared to figure 8. The steps of the spikes for TTFS encoding are:

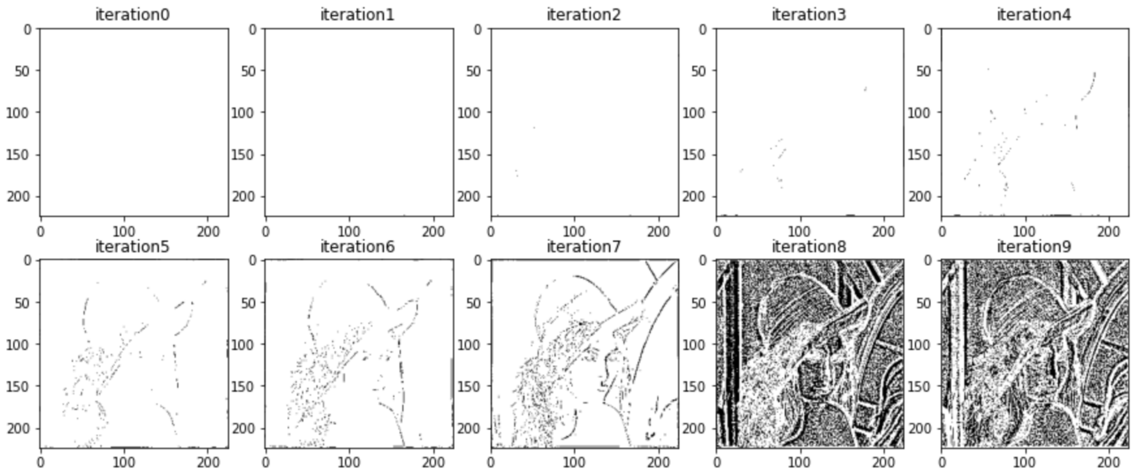


Figure 26: On-center-off-surround neurons spike in each iteration of ttfs encoding

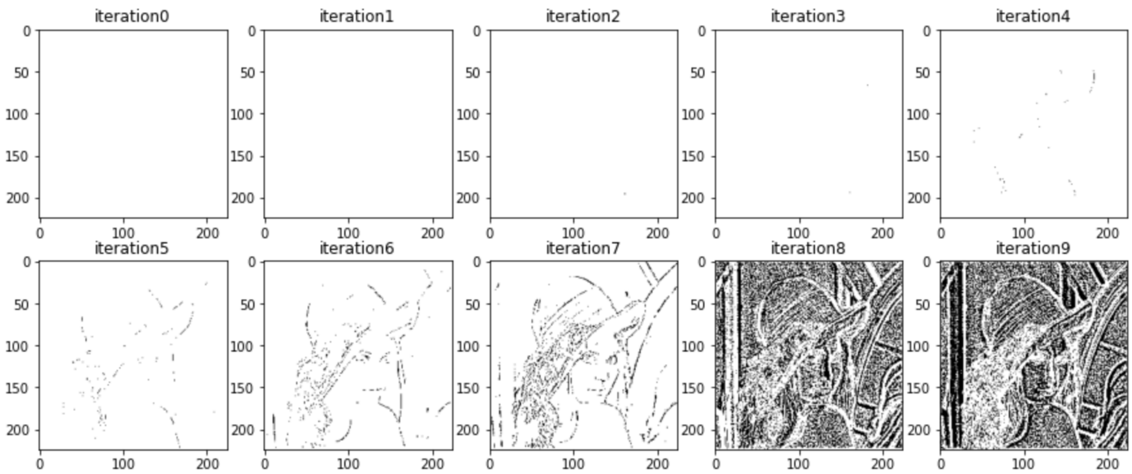


Figure 27: Off-center-on-surround neurons spike in each iteration of ttfs encoding

Now check the poisson encoding of convolved image obtained by on-center-off-surround filter:

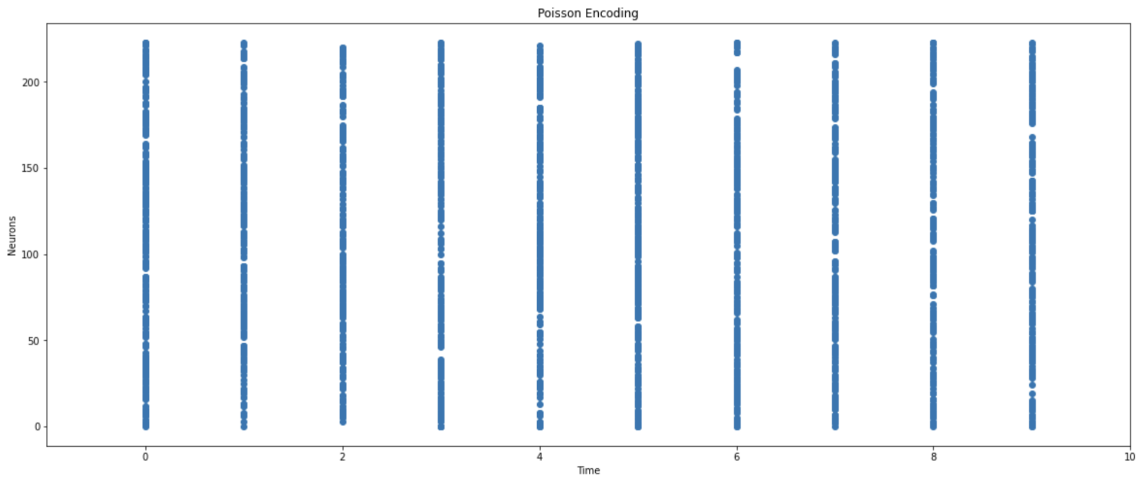
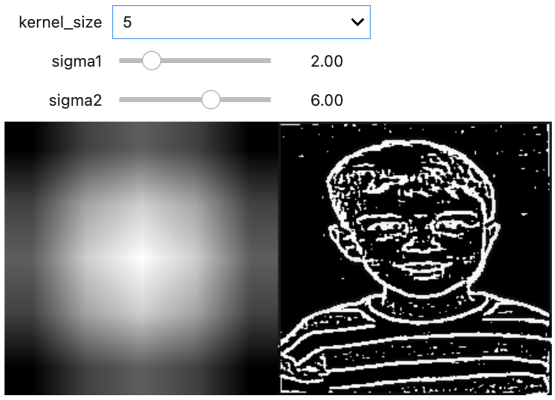
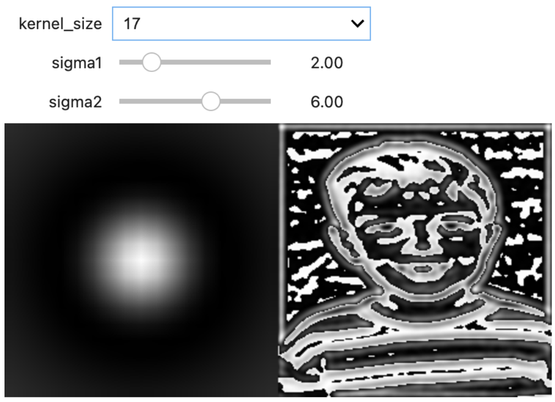


Figure 28: Off-center-on-surround neurons spike in each iteration of poisson encoding

As you see the neuron spike sooner compared to figure 9 and figure 10. So if the difference between gaussian network get larger and if we increases the size of our kernel, we have more spikes in the first iterations of TTFS encoding. Remember if the size of the kernel is too large, small features within the image may get suppressed, and the image may look blurred. Hence, the quality of the details of the image will be affected. Also If the kernel size is too small, eliminating the noises within the image will be compromised.
choosing our kernel size depends on the complexity of the image that we are trying to extract information. For an example, if we are trying to segment the parts from an whole sceneric image, as the objects will be big enough, then we can use big sized kernels. If its like texture, it's preferred to use smaller size. If the kernel size possibly be low, we will engage with texture.
See the picture below for:



(a) Kernel size is 5



(b) Kernel size is 17

Figure 29: Effect of Kernel Size

3 Gabor Filter

As we said before, the gabor filter simulates V1 simple cells which detect line with a particular orientation. These filters have been shown to possess optimal localization properties in both spatial and frequency domains and thus are well-suited for texture segmentation problems. Gabor filters are special classes of band-pass filters, i.e., they allow a certain band of frequencies and reject the others. A gabor filter can be viewed as a sinusoidal signal of particular frequency and orientation, modulated by a Gaussian wave. The formula is:

$$g(x, y, \lambda, \theta, \sigma, \gamma) = exp\left(-\frac{X^2 + \gamma^2 Y^2}{2\sigma^2}\right).cos\left(\frac{2\pi}{\lambda}X\right),$$

$$X = xcos(\theta) + ysin(\theta),$$

$$Y = -xsin(\theta) + ycos(\theta)$$

which:

- λ: Wavelength of the sinusoidal component.
- θ: The orientation of the normal to the parallel stripes of the Gabor function.
- σ: sigma/standard deviation of the Gaussian envelope

- γ: The spatial aspect ratio and specifies the ellipticity of the support of the gabor function.

You can see the filter in below:

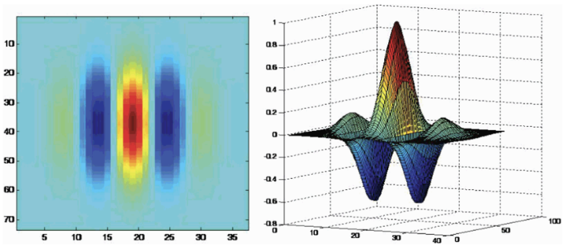


Figure 30: Gabor Filter

Now like last section we analyze the parameters of this filter and convolve it on images and see the results. Also we see the TTFS and Poisson encoding obtained from these convolved images.

3.1 Theta Parameter

In this part we analyze the theta parameter which indicates the orientation of the parallel stripes of the gabor function. Assume a filter with these parameter:

- kernel size = 5
- $\lambda = 5$
- $\gamma = 0.1$
- $\sigma = 4$

Now our original image is a white circle in a dark background:

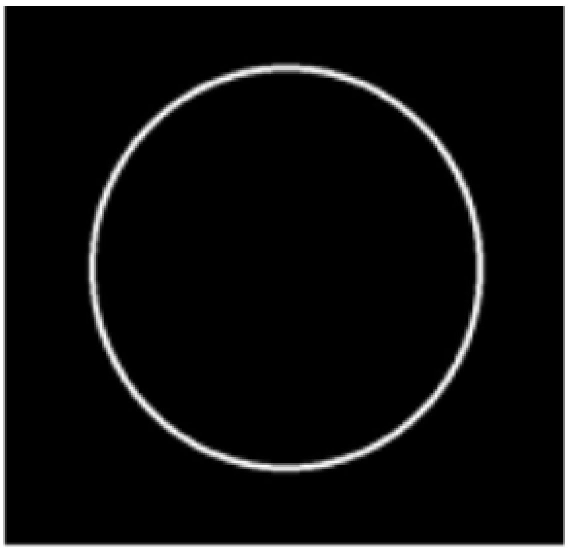


Figure 31: Original Image

Now we test 16 gabor filter with different orientations (theta):

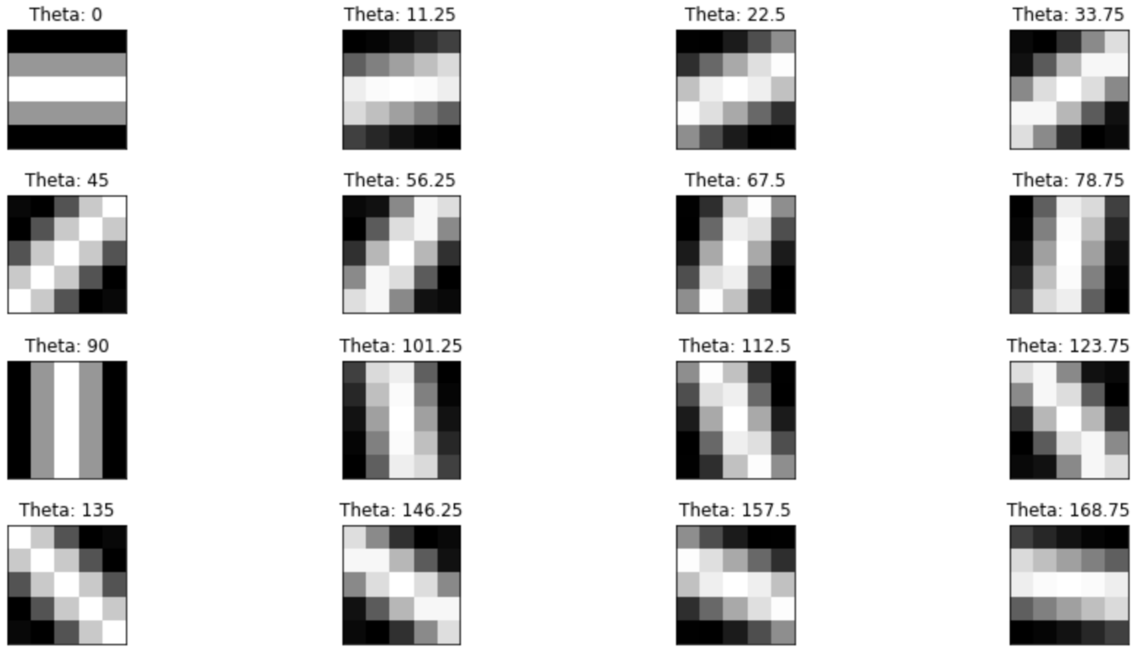


Figure 32: 16 Gabor Filters With Different Orientations

When we convolve these filters on the original images, we will have this result:

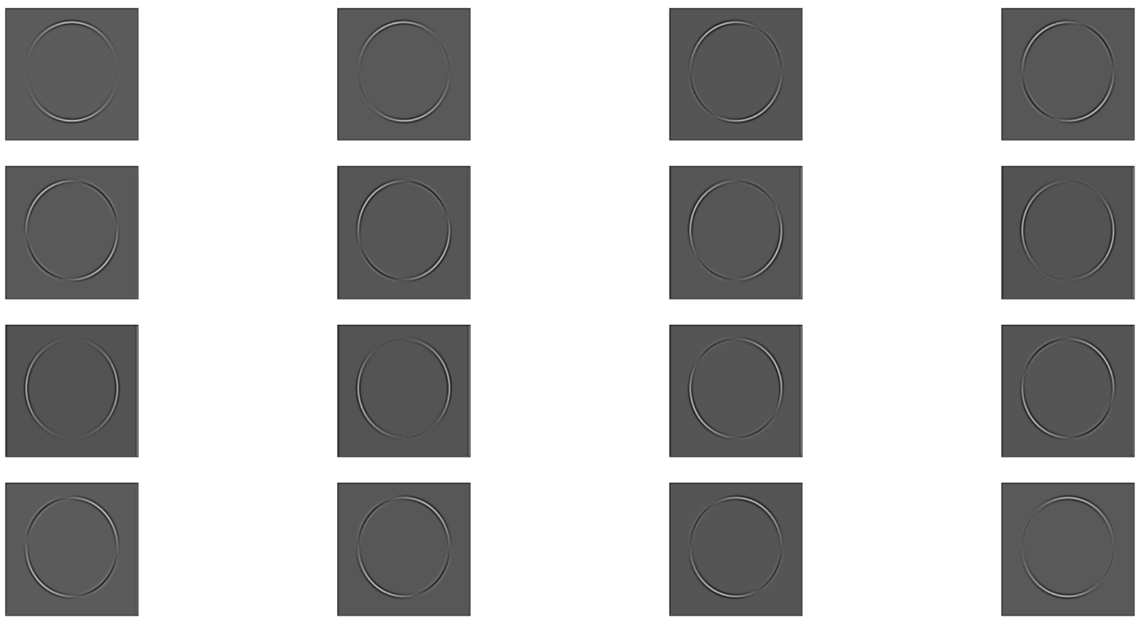


Figure 33: 16 Convolved Images

As you see the features are extracted in a specific direction for example horizontal and vertical features are extracted with θ=0 and θ=90, respectively.
Now we want to encode two gabor filters with θ=0 and θ=90 using TTFS and Poisson encoding:

1. θ=0: The TTFS encoding with the time window of 10 is:

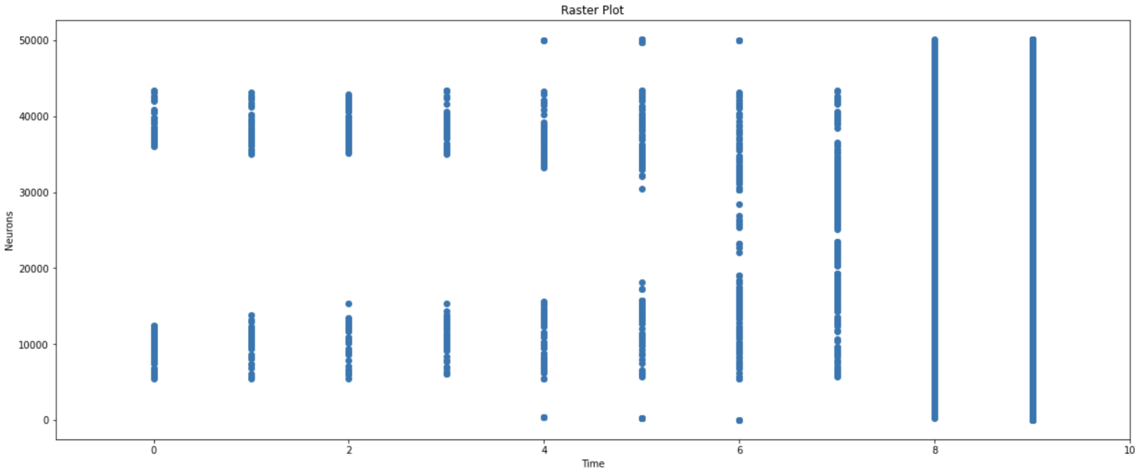


Figure 34: TTFS encoding

Also in below, we can see the raster plot of poisson encoding with the time window of 10 and r = 0.001:

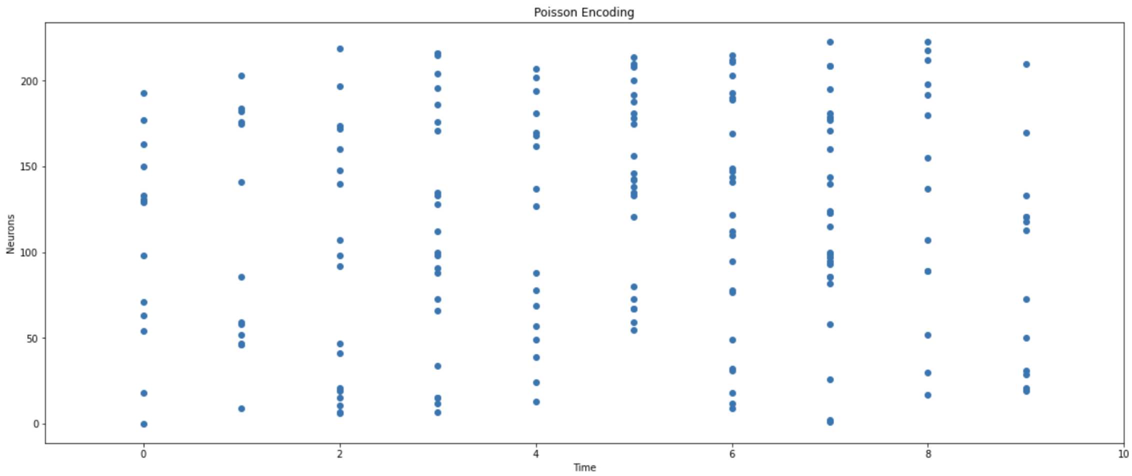


Figure 35: Poisson encoding

2. θ=90: Now the orientation of the gabor filter is 90 degree and it extracts vertical lines. The TTFS encoding is:

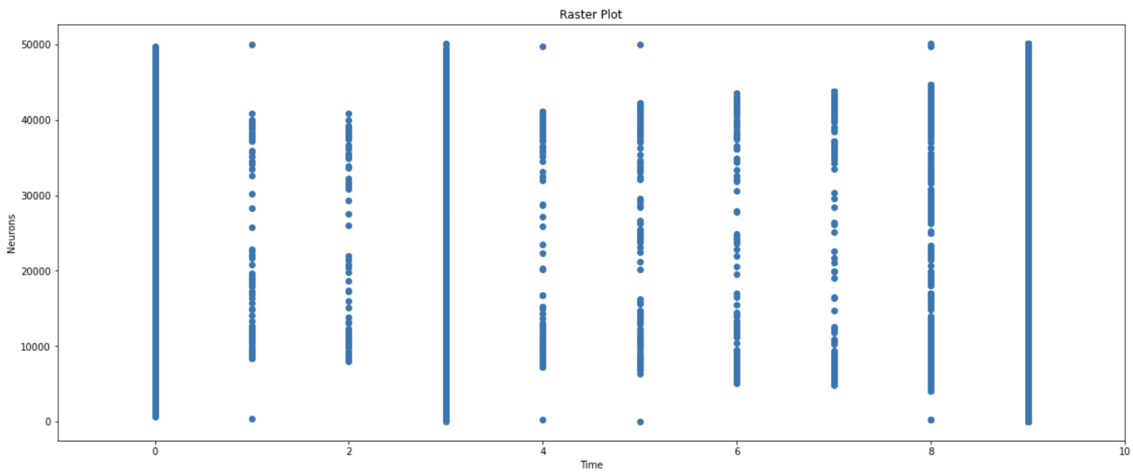


Figure 36: TTFS encoding

If you compare this encoding to figure 34, we see more neurons spike in the first iteration. That's because we have larger values when we apply 90 degree filter on image. If we want to see the spikes of neurons in each spike, it would be:

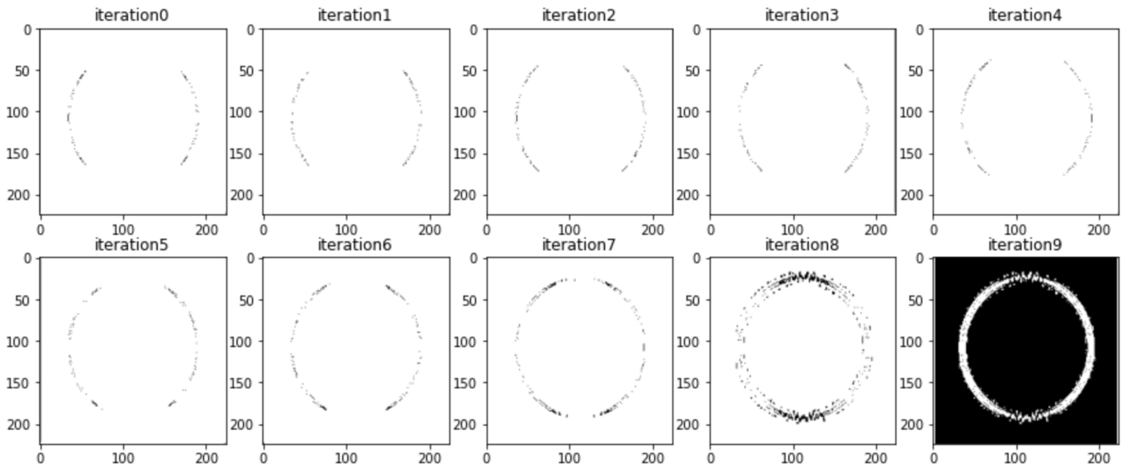


Figure 37: neurons spike in each iteration of ttfs encoding

As you see first the neurons which their receptive fields are in the line with 90 degree spike.
Also in below, we can see the raster plot of poisson encoding:

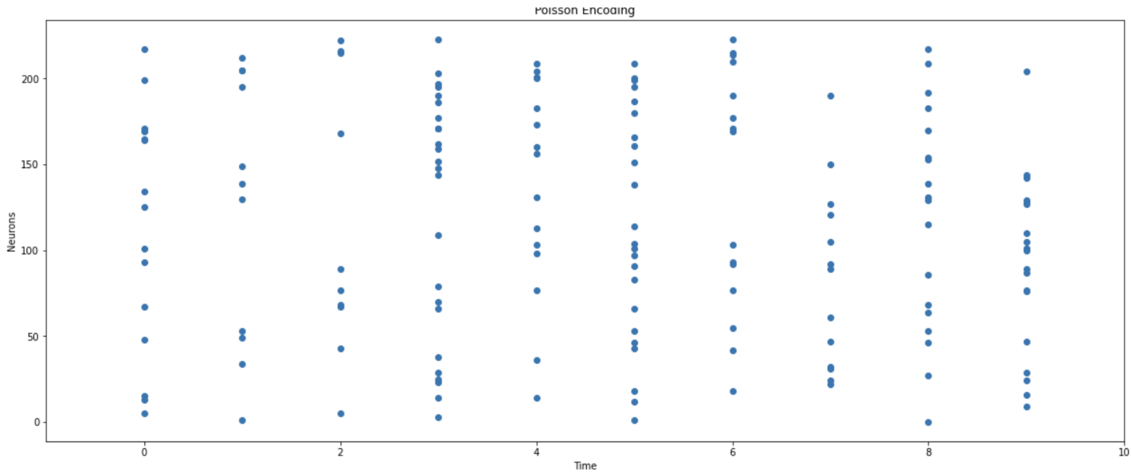


Figure 38: Poisson encoding

3.2 Lambda parameter

As we said before the wavelength governs the width of the strips of the gabor function. Increasing the wavelength produces thicker stripes and decreasing the wavelength produces thinner stripes.
Our original image for this section is:

Figure 39: Original Image

Here we construct a filter with these parameters:

- kernel size = 17
- $\sigma = 2$
- $\gamma = 0.5$

Now in two orientation (0 and 90 degree), we construct this filter and test the parameter of λ in extracting features. See the picture below, this is the result of our work:

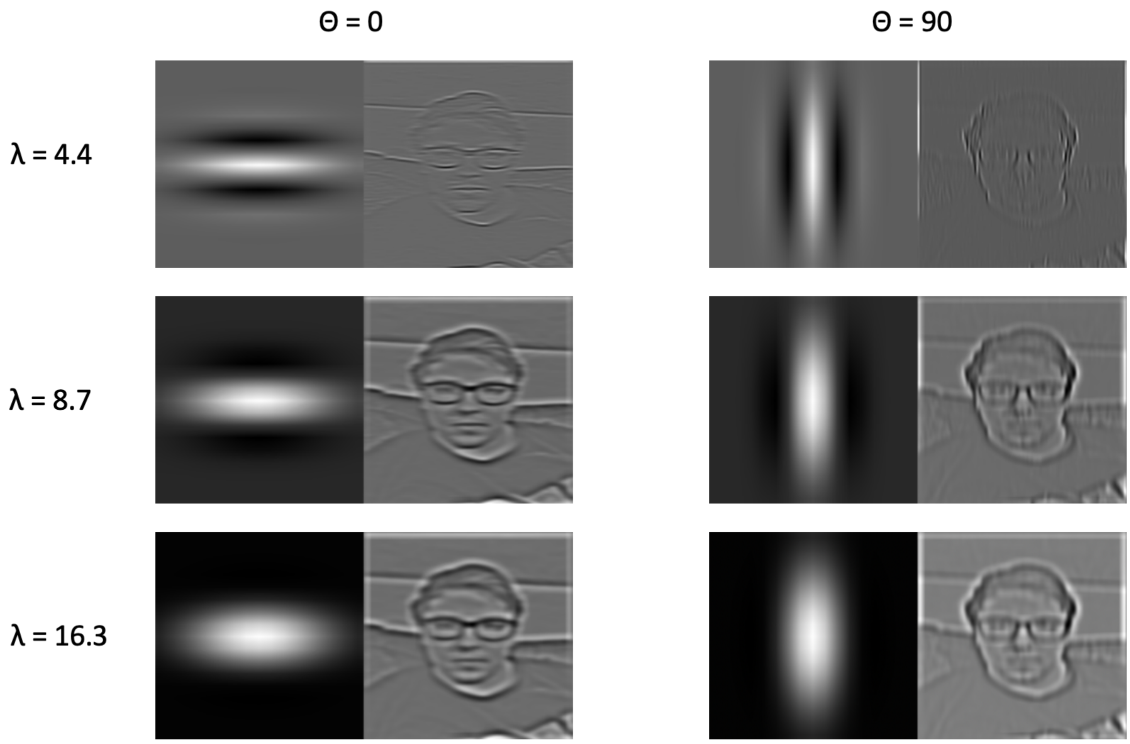
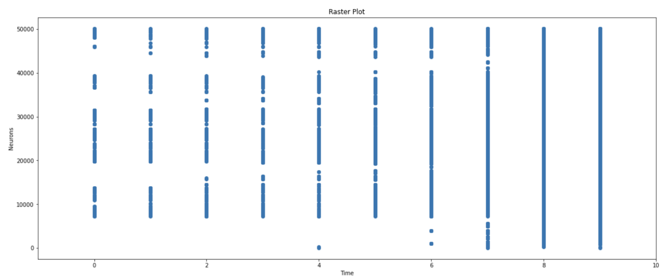
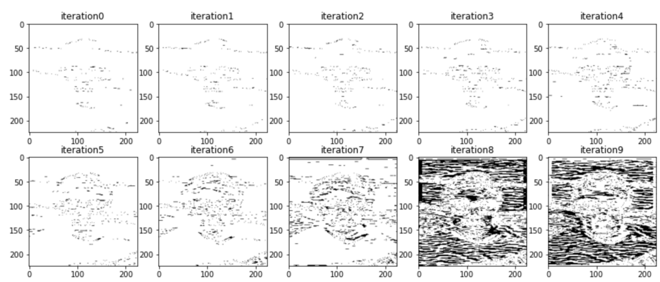


Figure 40: Effect of λ parameter

As you see when θ is 0, the horizontal lines will extracted and when θ is 90, the vertical lines will extracted. The other point is we can extract finer details with lower wavelength values.
Now Let's compare the TTFS encoding of the convolved images when θ is 0 and λ is 4.4 0r 16.3:

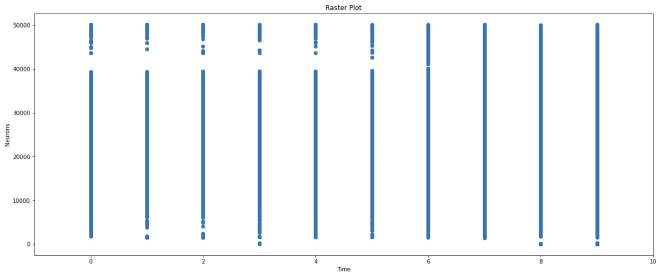


(a) TTFS encoding when λ is 4.4

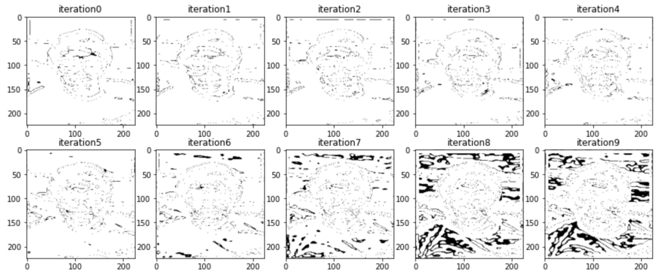


(b) steps of TTFS encoding when λ is 4.4

Figure 41: λ is 4.4 and θ is 0



(a) TTFS encoding when λ is 16.3



(b) steps of TTFS encoding when λ is 16.3

Figure 42: λ is 16.3 and θ is 0

As we said, decreasing the wavelength produces thinner stripes, so when λ is smaller lower pixel intensities will be produced and we have TTFS encoding like figure 41 part (a) and figure 42 part (a). Because of this we have more spikes in the first iteration of the convolved image obtained by the gabor filter with λ 16.3. You can see this by comparing figure 41 part (b) with figure 42 part (b).

4 Interactive Filters

In this project, we made it happen to play with the parameters of Dog and Gabor filter and see the convolved images obtained by these filter in real time. You can see this in the jupyter notebook of this project.
Now see a preview of this interactive part:

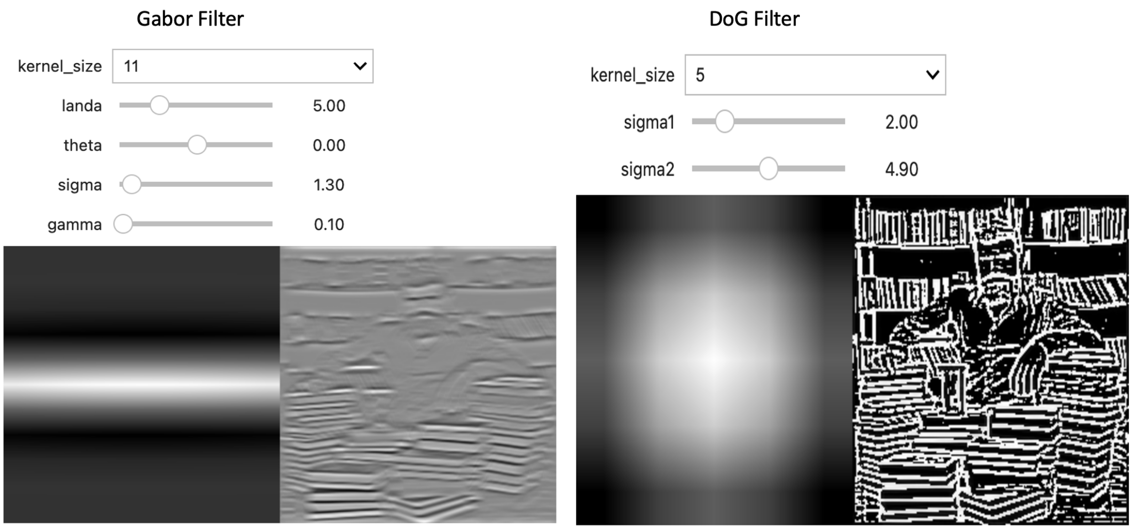


Figure 43: Interactive Filters

5 References

[1] Visual Cortex https://en.wikipedia.org/wiki/Visual_cortex

[2] Retina-LGN https://maktabkhooneh.org/course/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D8%B1%D8%A7%DB%8C%DA%AF%D8%A7%D9%86-%D8%B9%D9%84%D9%88%D9%85-%D8%A7%D8%B9%D8%B5%D8%A7%D8%A8-%D9%85%D8%AD%D8%A7%D8%B3%D8%A8%D8%A7%D8%AA%DB%8C-mk744/%D9%81%D8%B5%D9%84-%D8%A7%D9%88%D9%84-%D8%B9%D9%84%D9%88%D9%85-%D8%A7%D8%B9%D8%B5%D8%A7%D8%A8-%D9%85%D8%AD%D8%A7%D8%B3%D8%A8%D8%A7%D8%AA%DB%8C-ch2077/%D9%88%DB%8C%D8%AF%DB%8C%D9%88-%D8%B4%D8%A8%DA%A9%D9%87-%D9%87%D8%B3%D8%AA%D9%87-%D8%AE%D9%85%DB%8C%D8%AF%D9%87-%D8%AC%D8%A7%D9%86%D8%A8%DB%8C/

[3] Visual Cortex https://maktabkhooneh.org/course/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D8%B1%D8%A7%DB%8C%DA%AF%D8%A7%D9%86-%D8%B9%D9%84%D9%88%D9%85-%D8%A7%D8%B9%D8%B5%D8%A7%D8%A8-%D9%85%D8%AD%D8%A7%D8%B3%D8%A8%D8%A7%D8%AA%DB%8C-mk744/%D9%81%D8%B5%D9%84-%D8%A7%D9%88%D9%84-%D8%B9%D9%84%D9%88%D9%85-%D8%A7%D8%B9%D8%B5%D8%A7%D8%A8-%D9%85%D8%AD%D8%A7%D8%B3%D8%A8%D8%A7%D8%AA%DB%8C-ch2077/%D9%88%DB%8C%D8%AF%DB%8C%D9%88-%D8%B9%D9%85%D9%84%DA%A9%D8%B1%D8%AF-%D9%82%D8%B4%D8%B1-%D8%A8%DB%8C%D9%86%D8%A7%DB%8C%DB%8C-%D9%85%D8%BA%D8%B2/

[4] Difference of Gaussian https://en.wikipedia.org/wiki/Difference_of_Gaussians

[5] Gabor Filter https://en.wikipedia.org/wiki/Gabor_filter