DOG AND GABOR FILTERS

August 28, 2023

Amin Zeinali
University of Tehran
Faculty of Mathematics, Statistics, and Computer Sciences

1 Introduction

Gabor filters and Difference of Gaussians (DoG) filters are commonly used for feature extraction in image processing and computer vision tasks. Gabor filters capture orientation and spatial frequency information in local regions, making them useful for texture analysis. DoG filters mimic the receptive field properties of cells in the retina and act as an edge detector. In this project, I have implemented these two types of filters and tested them on sample images.

2 DoG Filter

The DoG filter simulates ganglion cells which detect points in a background with different colors. The 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, the DoG transformation of an image requires subtracting one highly blurred version of an original image from another less blurred version to preserve 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 Gaussian kernel in 2D is:

$$G_\sigma(x,y) = \frac{e^{-(x^2+y^2)/2\sigma^2}}{\sigma\sqrt{2.\pi}}$$

The DoG kernel is defined as follows:

$$DoG_{\sigma_1,\sigma_2} = G_{\sigma_1}(x,y) - G_{\sigma_2}(x,y)$$

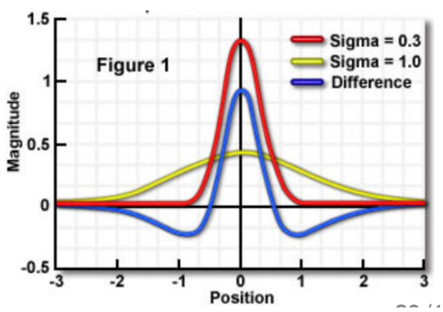

Figure 1: 1D difference of Gaussian

3 Gabor Filter

The Gabor filter simulates V1 simple cells which detect lines 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:

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

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

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

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.

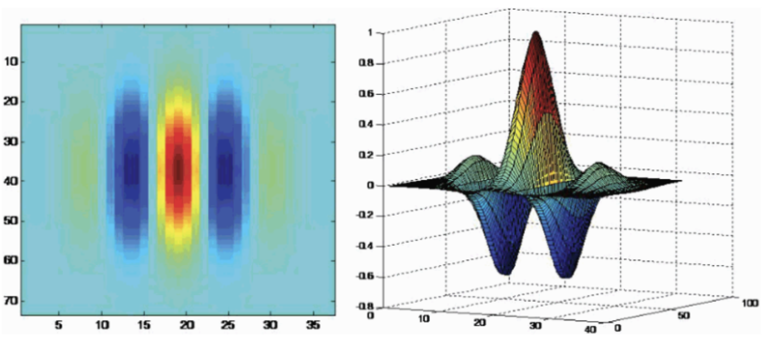

Figure 2: Gabor filter

4 Methodology

I defined Gabor and DoG filter classes in Python using PyTorch for efficient numerical computations on GPU. For the Gabor filter, I implemented the 2D Gabor kernel function that takes parameters like wavelength, orientation, standard deviation, etc., and generates a filter response. For the DoG filter, I defined two Gaussian kernels with different standard deviations and subtracted them to get the DoG kernel.

A convolution layer is implemented that can apply these filters on an input image. Options like padding and stride are passed to the convolution layer. The input is zero-padded for 'same' padding to retain the original size after convolution. The sliding window approach is used to convolve the filters across the image.

Max pooling is implemented afterward to downsample the feature maps. It divides the image into non-overlapping rectangular patches and outputs the maximum value within each patch.

I also use Time-to-First-Spike encoding in order to convert images to spike trains. If you are not familiar with TTFS, check my Computational-Neuroscience repository.

5 Results

In this section, we are going to apply different kernels on sample images and generate their spike trains.

5.1 DoG Filters

In the on-center-off-surround filter, we have positive values in the middle of the filter and negative values in the surround. We have the opposite of this matrix for off-center-on-surround.

5.1.1 on-center-off-surround

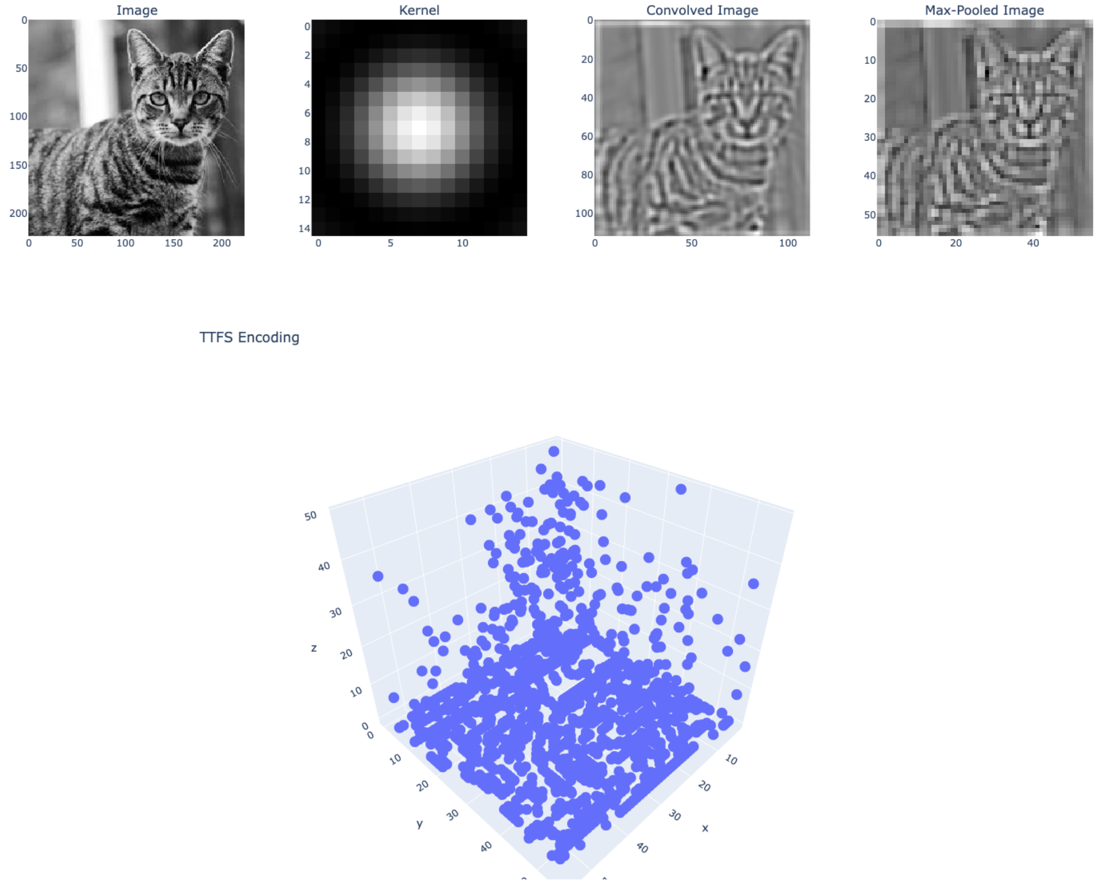

Figure 3: DoG filter and its results with parameters: $\sigma_1 = 5, \sigma_2 = 7$

5.1.2 off-center-on-surround

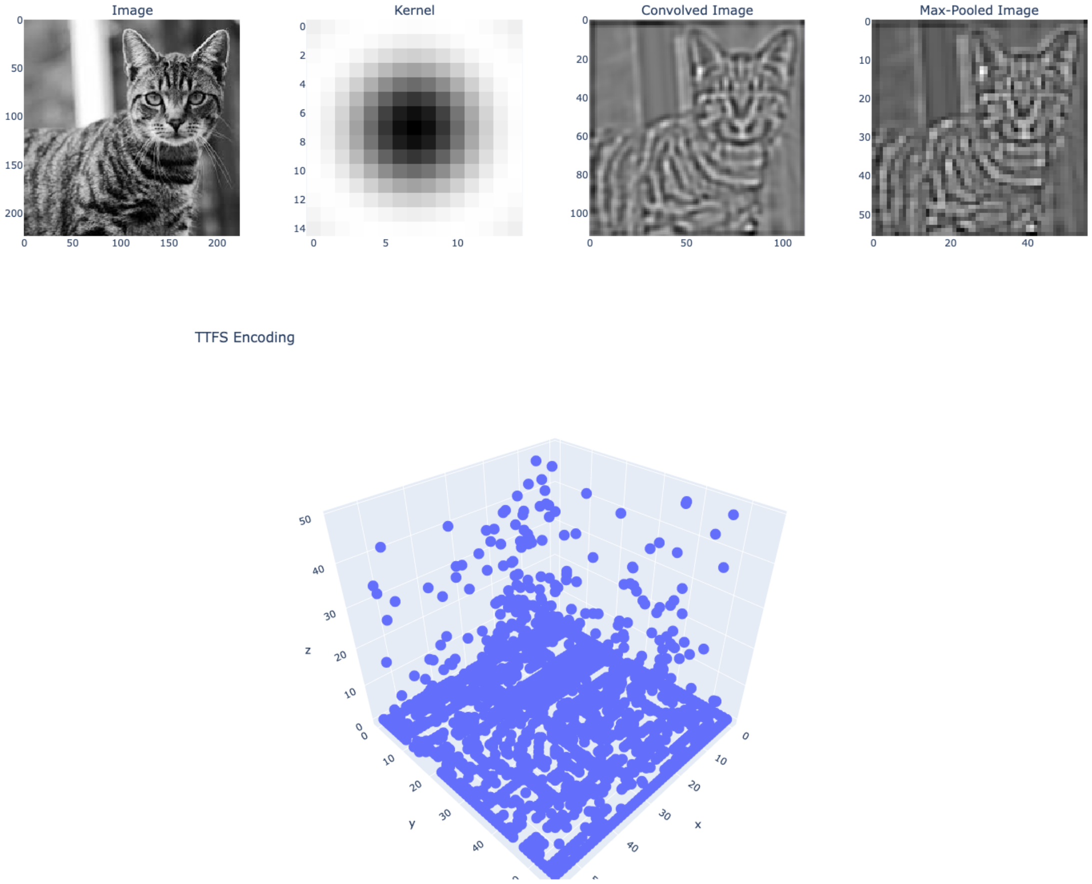

Figure 4: DoG filter and its results with parameters: $\sigma_1 = 7, \sigma_2 = 5$

5.2 Gabor Filters

Figure 5 shows a Gabor filter applied on a sample image in both forms of an image and spike train. In the following parts, we are going to change the parameters of the Gabor filter one by one and see the effects.

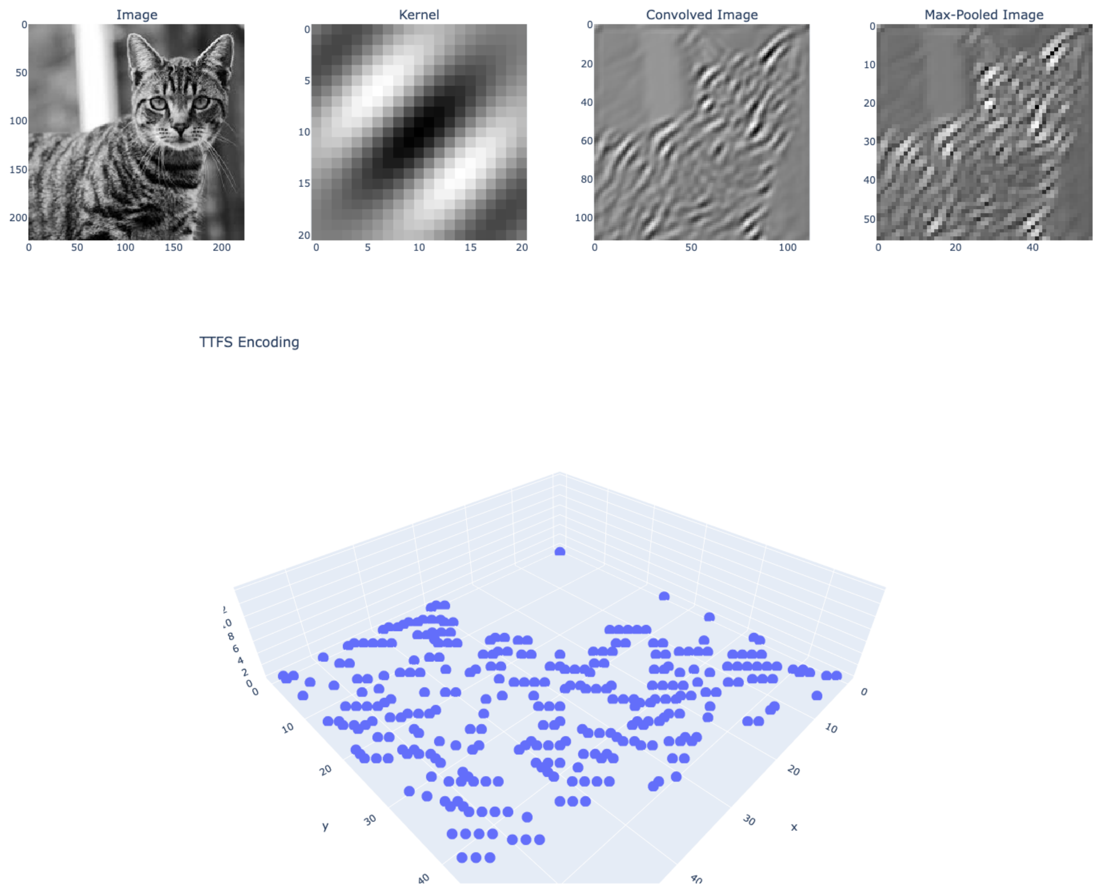

Figure 5: Gabor filter and its results with parameters: $\sigma = 10,\ \gamma = 1.6,\ \lambda = 12,\ \theta = \frac{\pi}{4}$

5.2.1 γ parameter

The aspect ratio or gamma controls the height of the Gabor function. For a very high aspect ratio, the height becomes very small and for a very small gamma value, the height becomes quite large. On increasing the value of gamma to 2.4, keeping other parameters unchanged, the height of the Gabor function reduces.

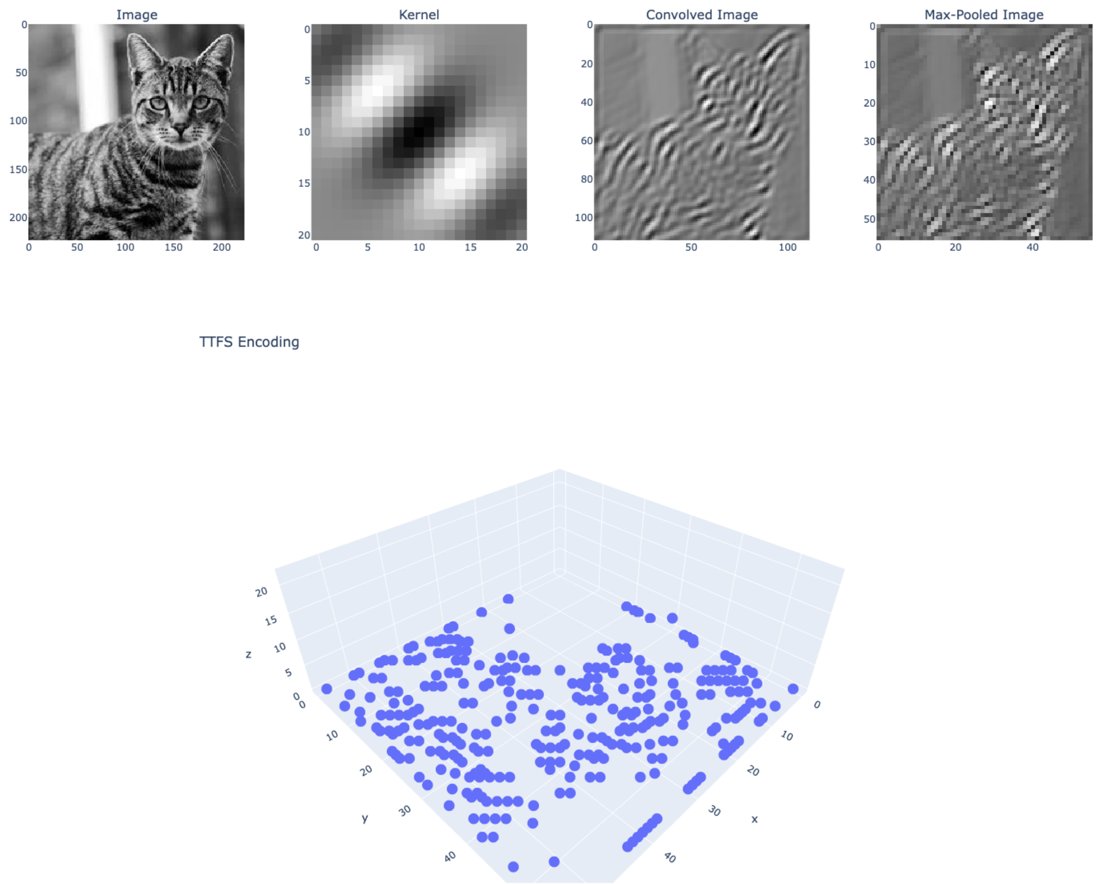

Figure 6: Gabor filter and its results with parameters: $\sigma = 10,\ \gamma = 2.4,\ \lambda = 12,\ \theta = \frac{\pi}{4}$

5.2.2 λ parameter

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. Keeping other parameters unchanged and changing the lambda to 15, the stripes get thicker.

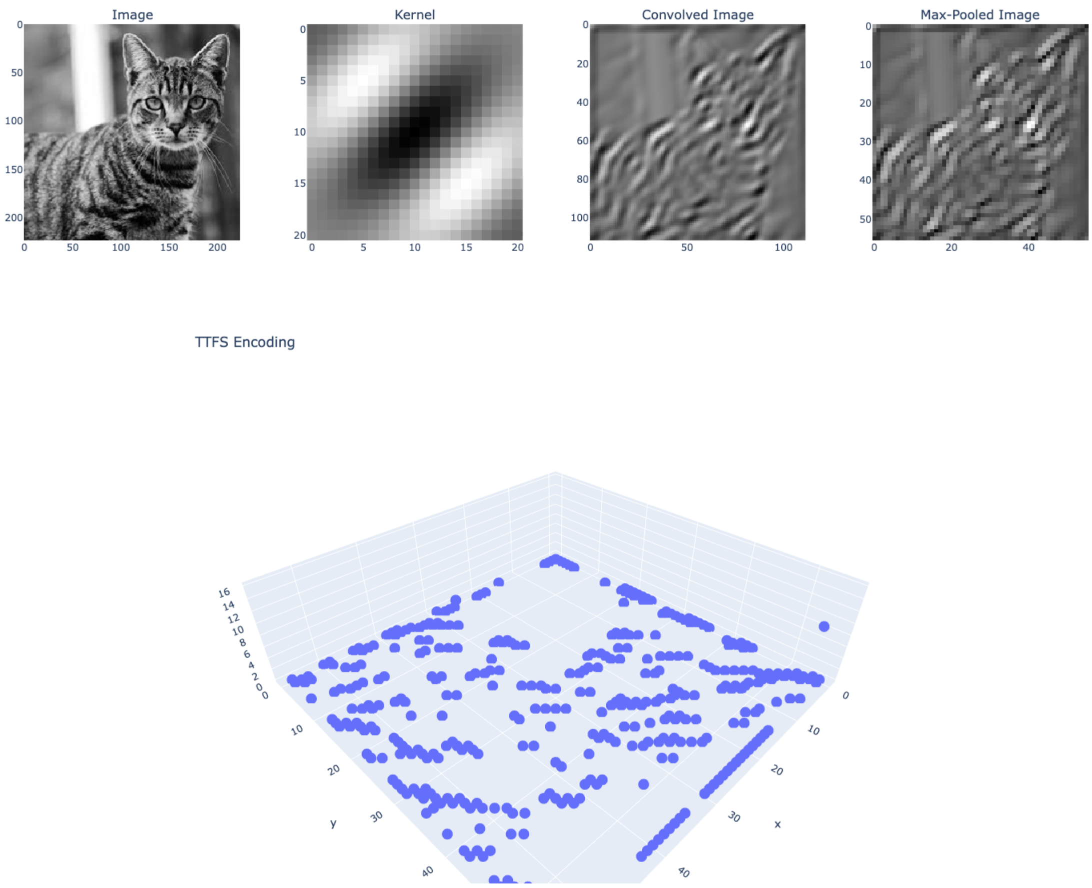

Figure 7: Gabor filter and its results with parameters: $\sigma = 10$, $\gamma = 2.4$, $\lambda = 15$, $\theta = \frac{\pi}{4}$

5.2.3 θ parameter

The θ controls the orientation of the Gabor function.

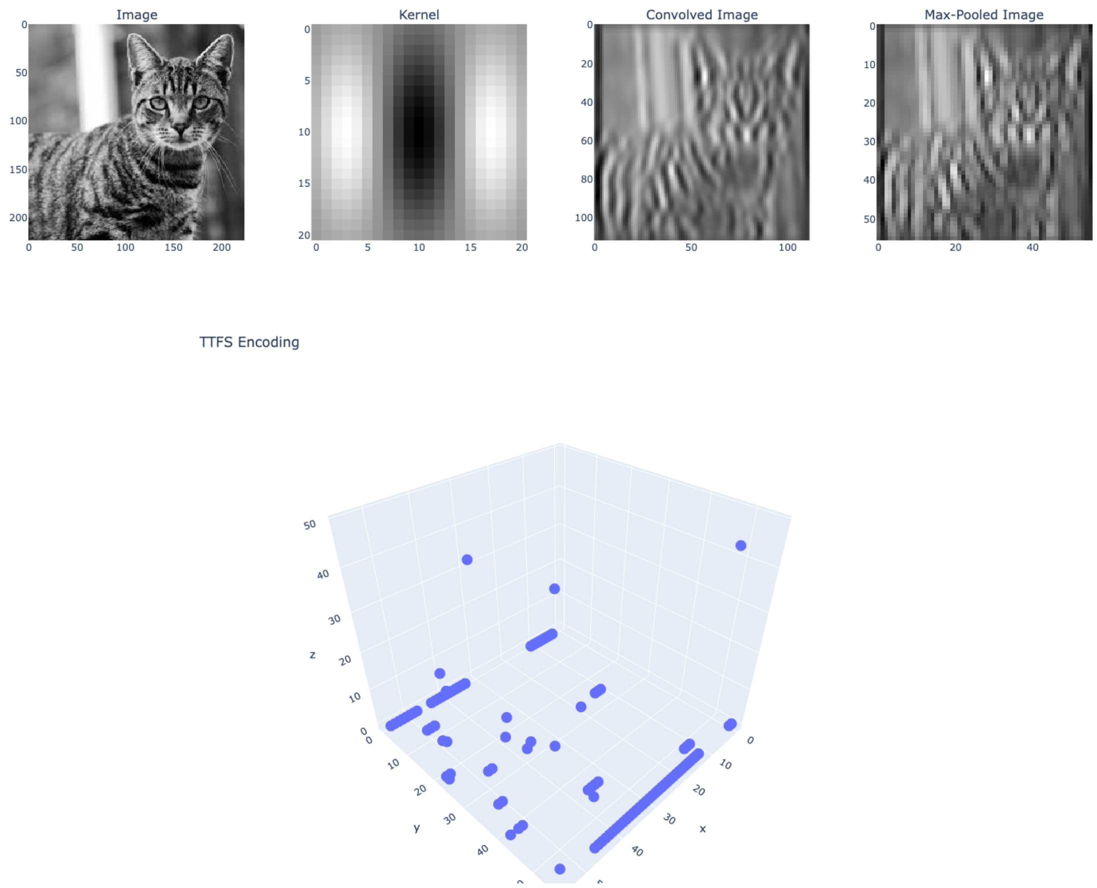

Figure 8: Gabor filter and its results with parameters: $\sigma = 10,\ \gamma = 2.4,\ \lambda = 15,\ \theta = \frac{\pi}{2}$

5.2.4 σ parameter

The bandwidth or sigma controls the overall size of the Gabor envelope. For larger bandwidth the envelope increase allowing more stripes and with small bandwidth the envelope tightens.

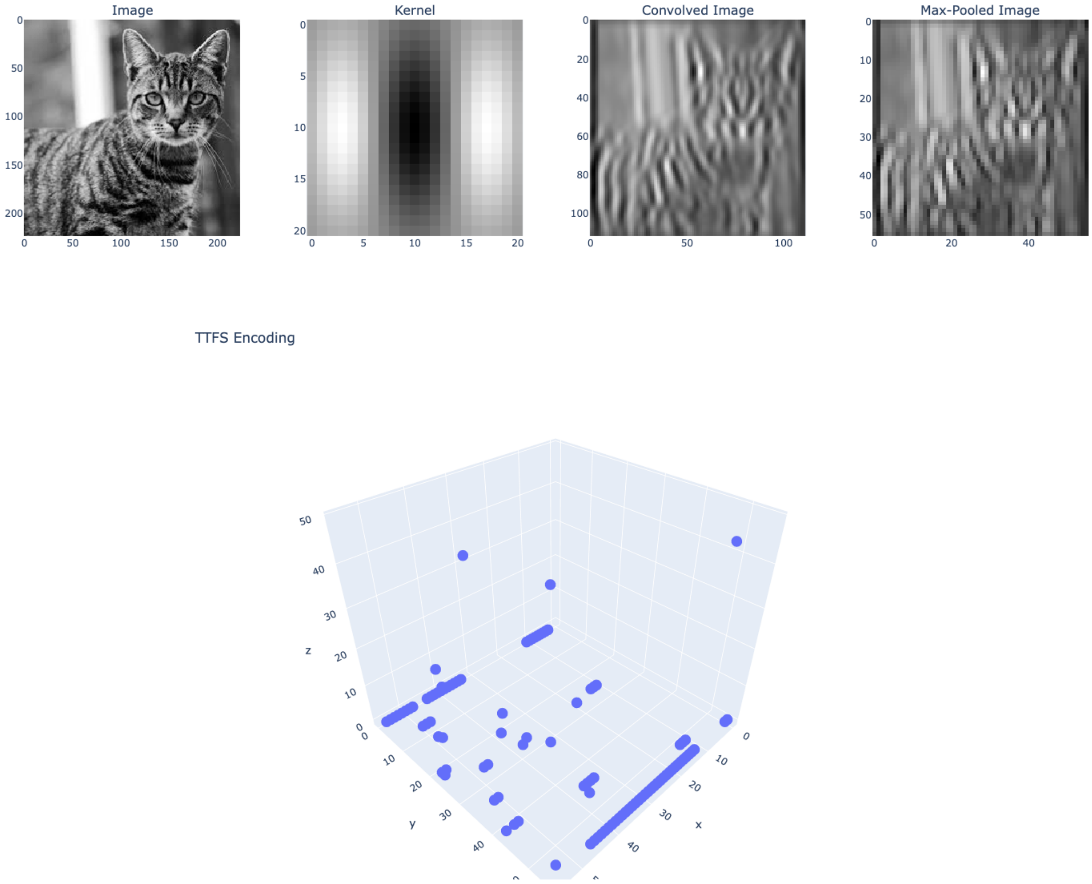

Figure 9: Gabor filter and its results with parameters: $\sigma = 18,\ \gamma = 2.4,\ \lambda = 15,\ \theta = \frac{\pi}{2}$

Bibliography

[1] Difference of Gaussian URL

[2] Gabor Filter URL

[3] Visual Cortex URL

[4] Retina-LGN URL