

project-8

June 4, 2021

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>                    | <b>2</b>  |
| <b>2</b> | <b>Part 1</b>                          | <b>3</b>  |
| 2.1      | DoG Filter . . . . .                   | 3         |
| 2.1.1    | <i>FilterSize = 5</i> . . . . .        | 3         |
| 2.1.2    | <i>FilterSize = 15</i> . . . . .       | 4         |
| 2.2      | Gabor Filter . . . . .                 | 4         |
| 2.2.1    | <i>FilterSize = 7</i> . . . . .        | 4         |
| 2.2.2    | <i>FilterSize = 15</i> . . . . .       | 5         |
| <b>3</b> | <b>Part 2</b>                          | <b>8</b>  |
| 3.1      | DoG Filters . . . . .                  | 8         |
| 3.2      | Gabor Filters . . . . .                | 12        |
| 3.2.1    | Effect of $\sigma$ . . . . .           | 12        |
| 3.2.2    | Effect of $\theta$ . . . . .           | 13        |
| 3.2.3    | Effect of $\gamma$ . . . . .           | 16        |
| 3.2.4    | Effect of $\lambda$ . . . . .          | 17        |
| 3.3      | Effect of Convolution Stride . . . . . | 19        |
| <b>4</b> | <b>Summary</b>                         | <b>21</b> |

# Chapter 1

## Introduction

In this project, we implement `DoG` and `Gabor` filters and use different filter parameter sets and convolve them on two images, then we clip the negative values of convolution's output. Finally, we encode the output of previous steps using `Time2FirstSpike` and `Poisson Encoders`.

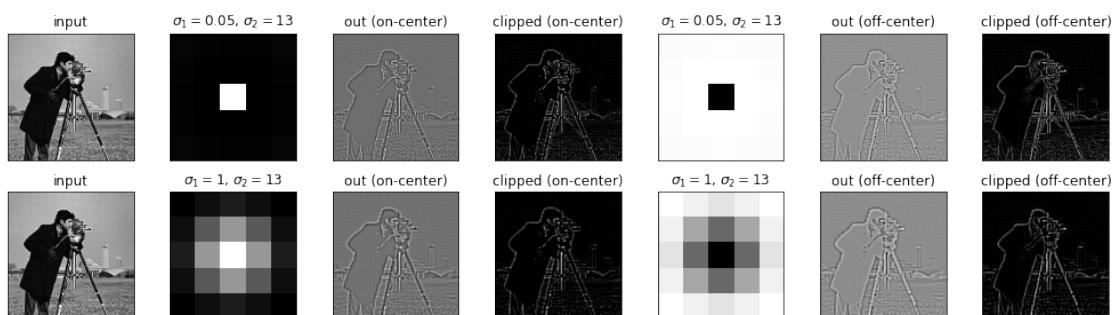
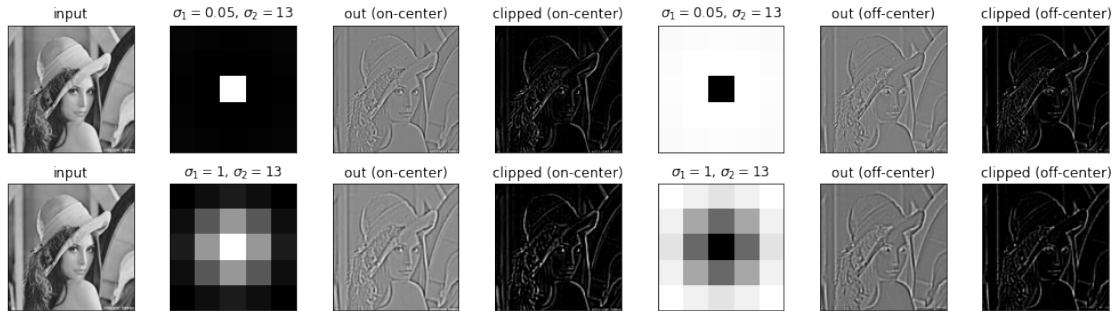
# Chapter 2

## Part 1

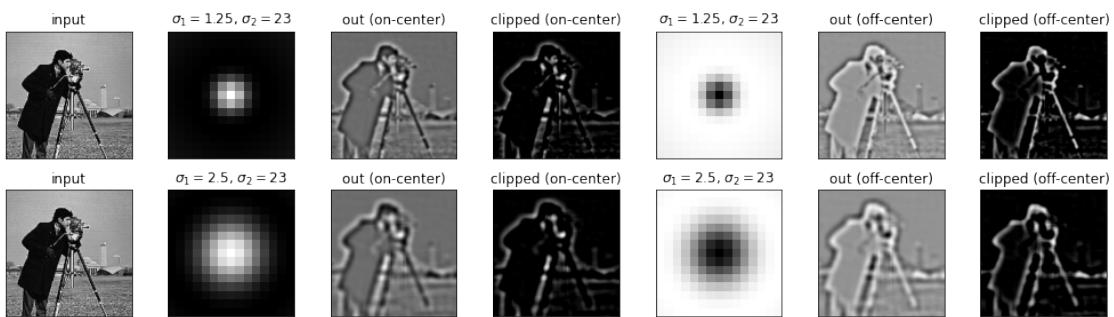
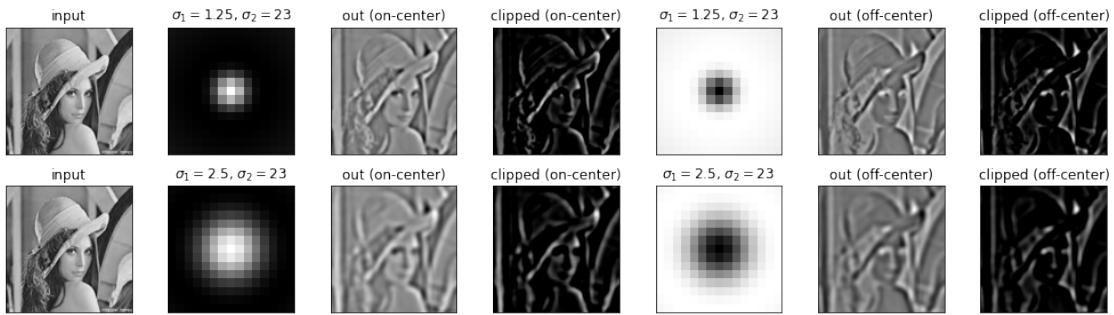
### 2.1 DoG Filter

We use two different filter sizes: 5 and 13.

#### 2.1.1 $\text{FilterSize} = 5$



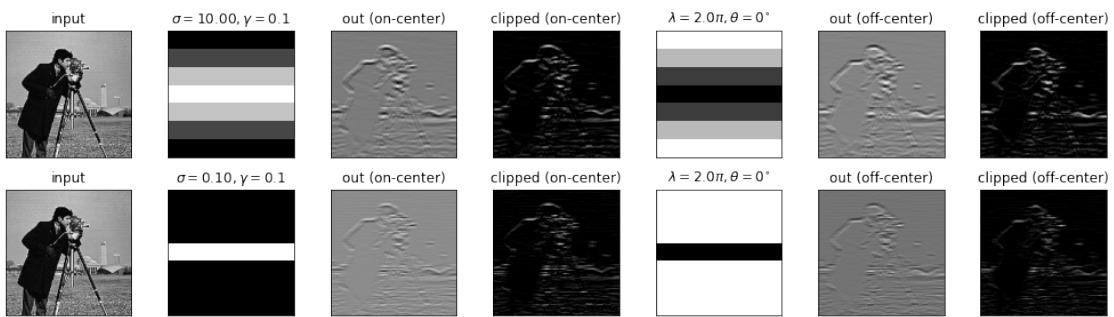
### 2.1.2 FilterSize = 15

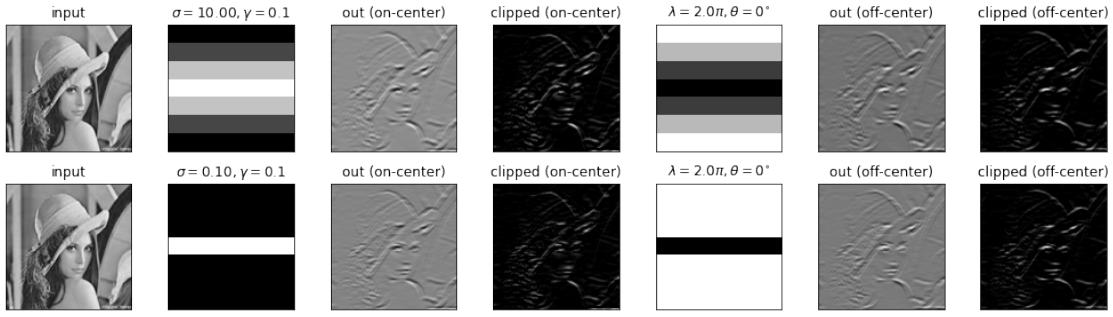


## 2.2 Gabor Filter

### 2.2.1 FilterSize = 7

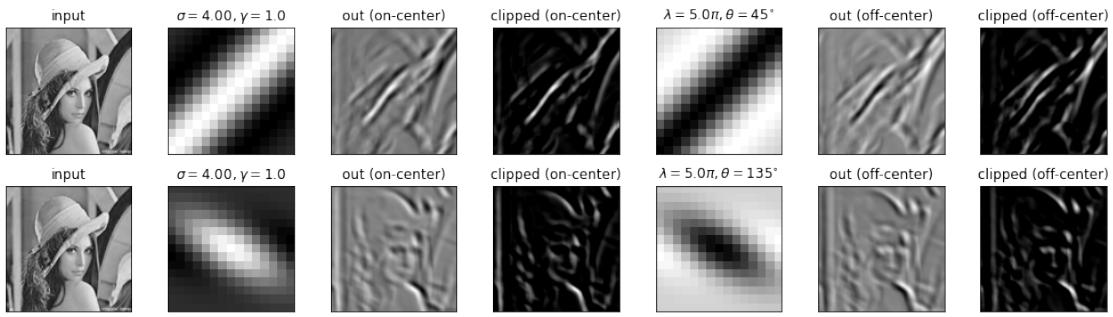
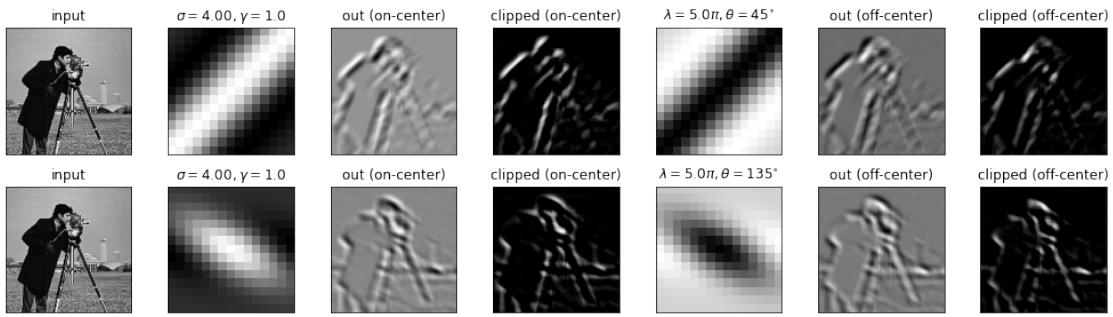
$\sigma$  Effect



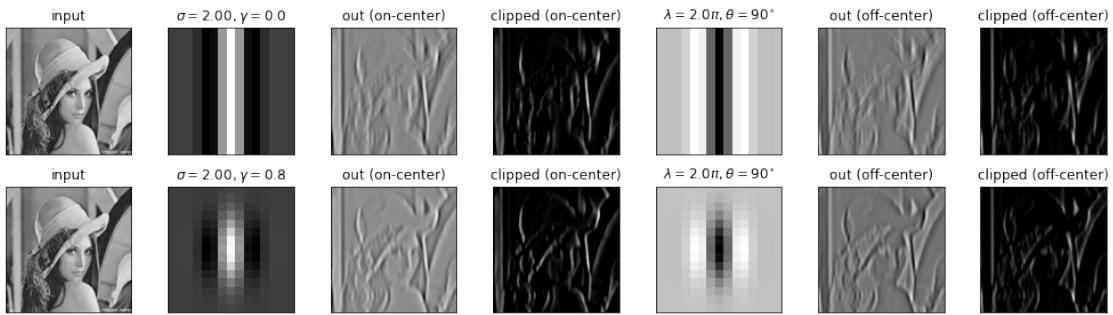
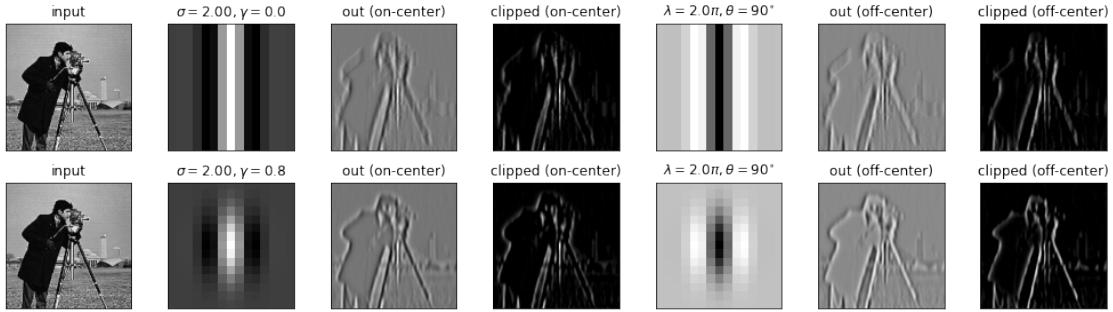


## 2.2.2 $FilterSize = 15$

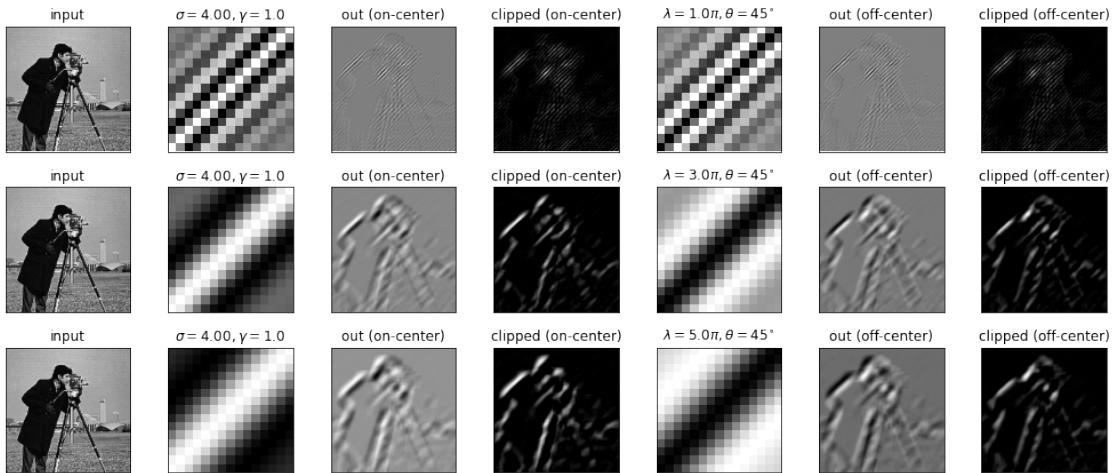
$\theta$  Effect

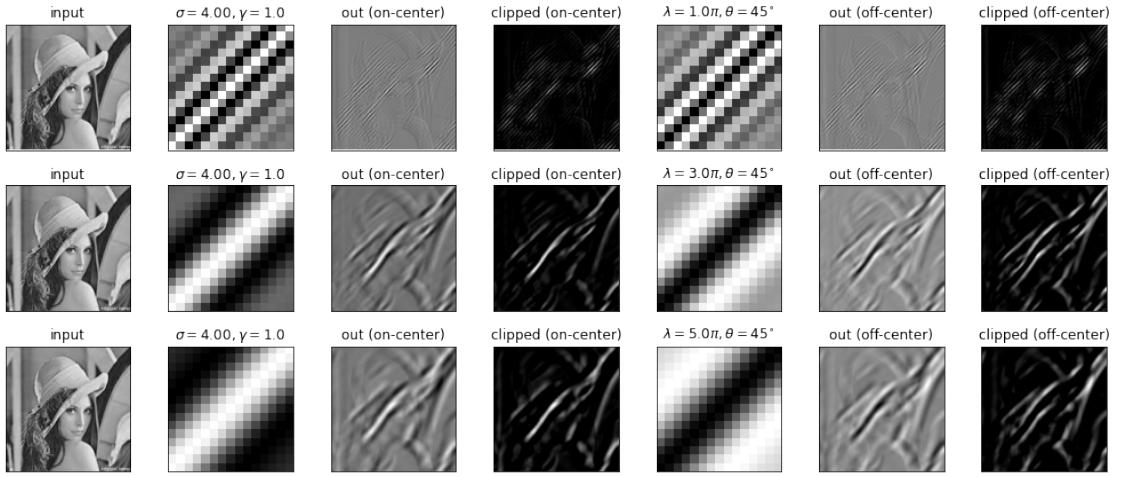


## $\gamma$ Effect



## $\lambda$ Effect





# Chapter 3

## Part 2

In this section we draw encoding (using `Time2FirstSpike` and `Poisson`) outputs of the clipped feature maps.

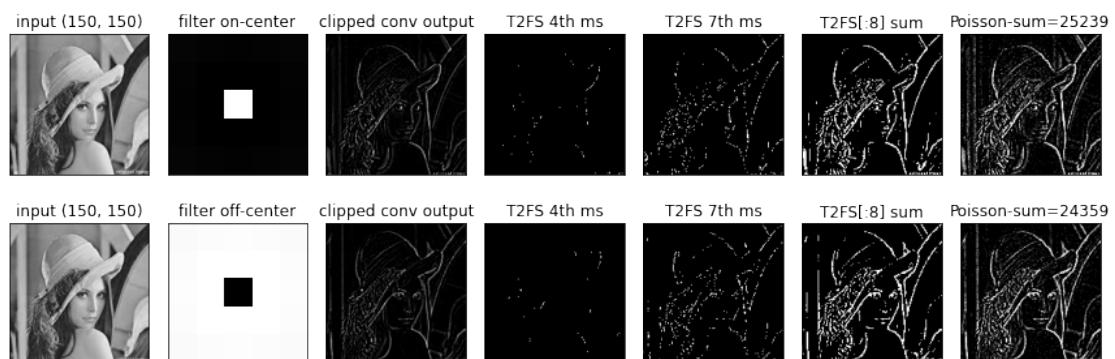
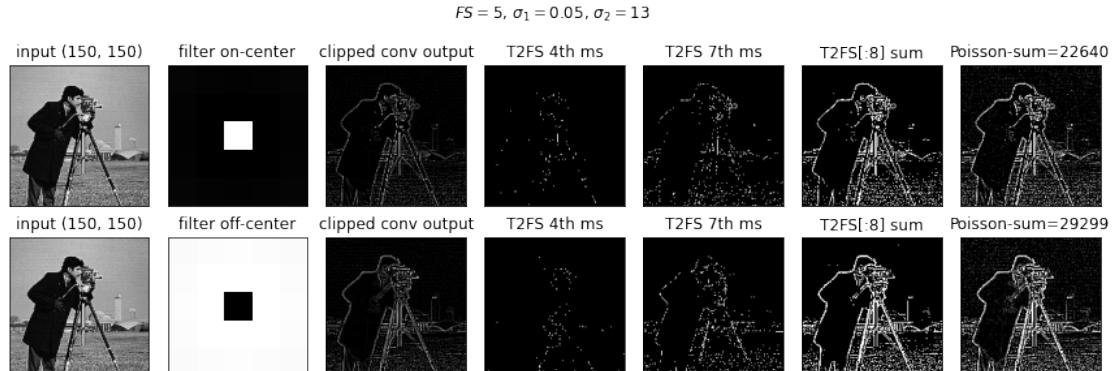
Parameters of encoders:

**Time2FirstSpike**: *EncodingTime* : 10ms

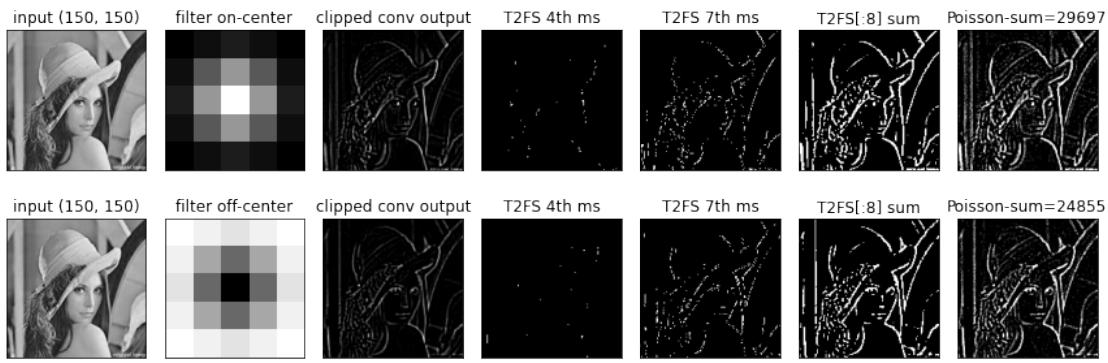
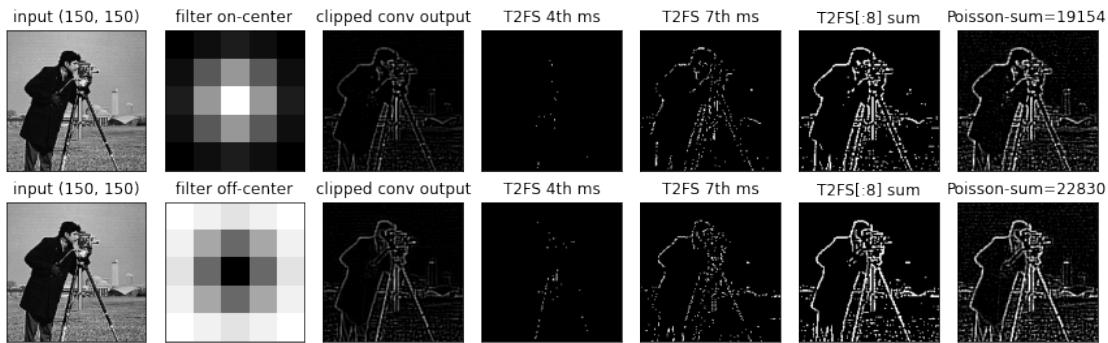
**PoissonEncoder**: *EncodingTime* : 10ms, *RateMax* = 25

**Note**: The *RateMax* = 25 is selected for `PoissonEncoder` to make the output of the encoder more visible. If we set *RateMax* = *EncodingTime*, the sum of outputs in all time-steps will be almost the same as the input.

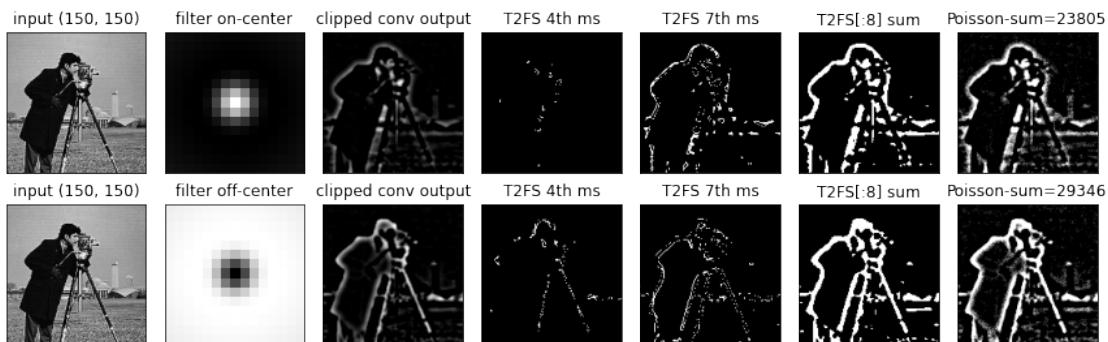
### 3.1 DoG Filters

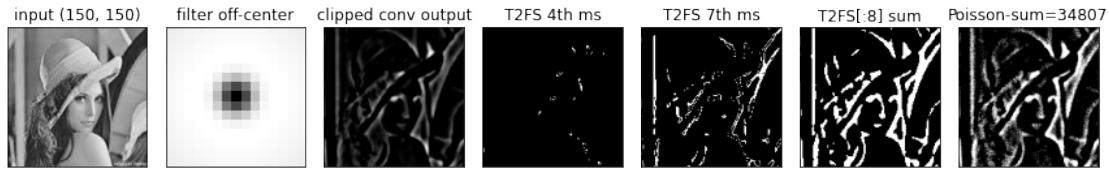
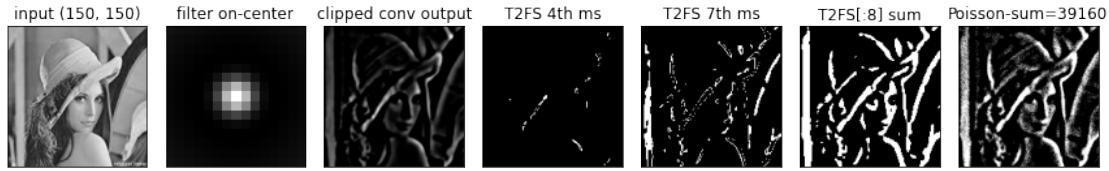


$$FS = 5, \sigma_1 = 1, \sigma_2 = 13$$

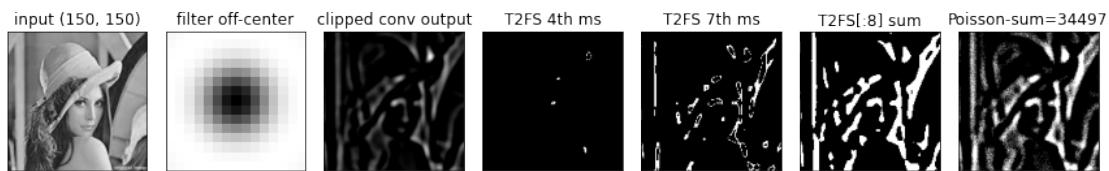
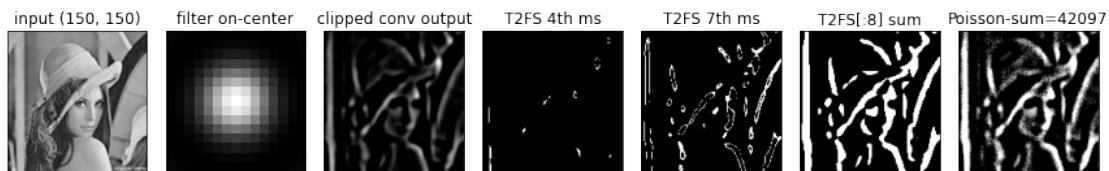
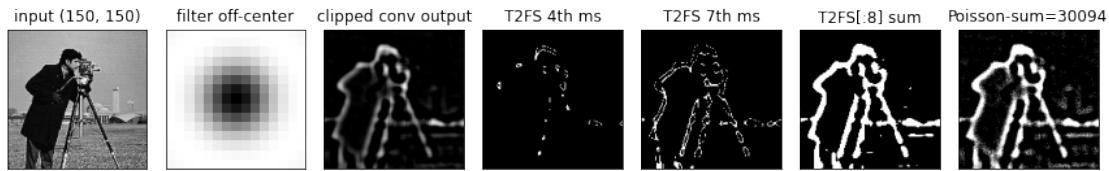
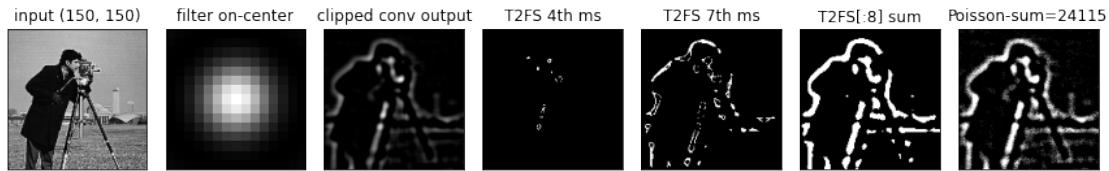


$$FS = 15, \sigma_1 = 1.25, \sigma_2 = 23$$

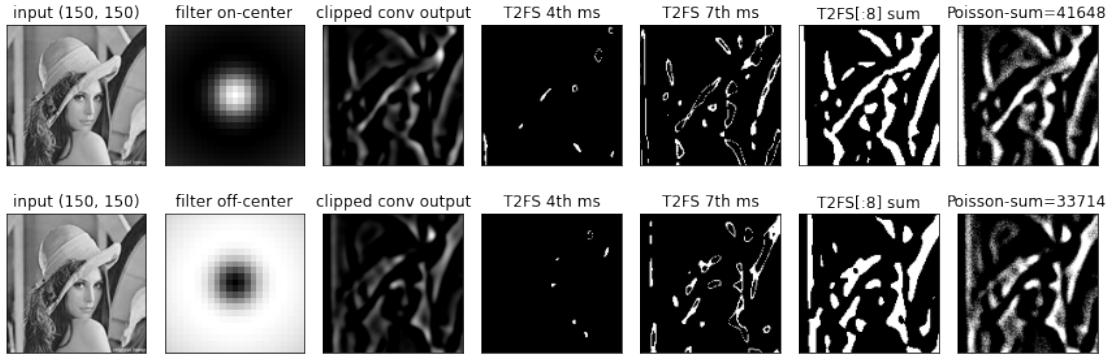
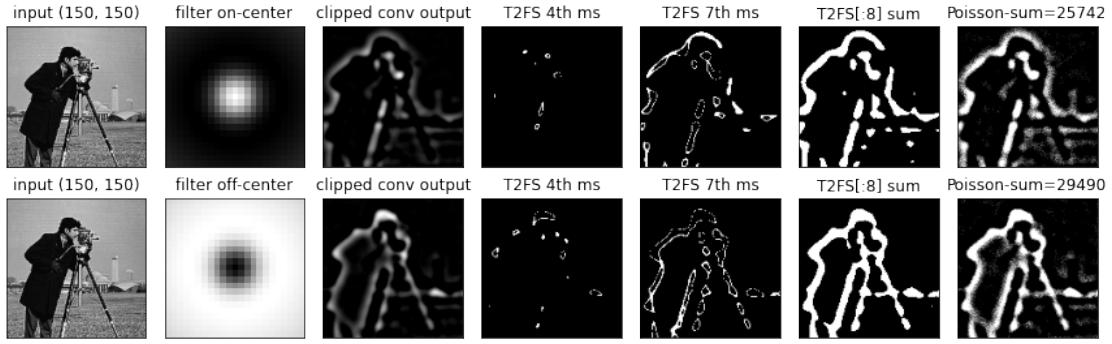




$FS = 15, \sigma_1 = 2.5, \sigma_2 = 23$



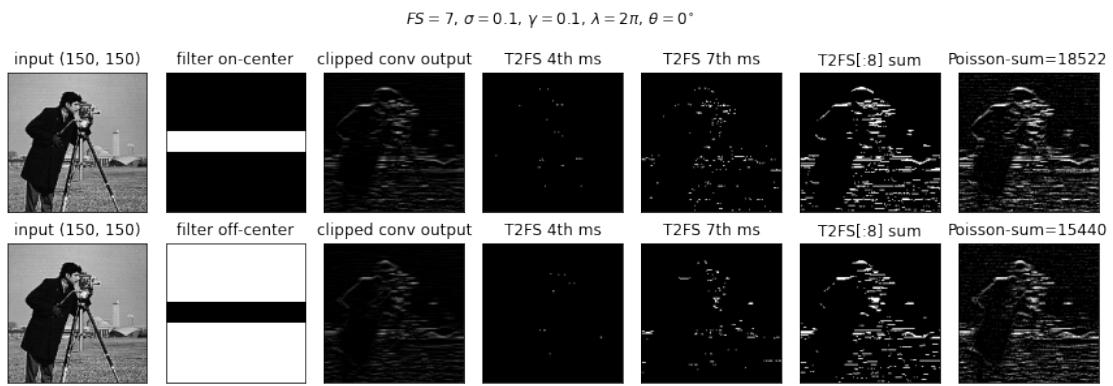
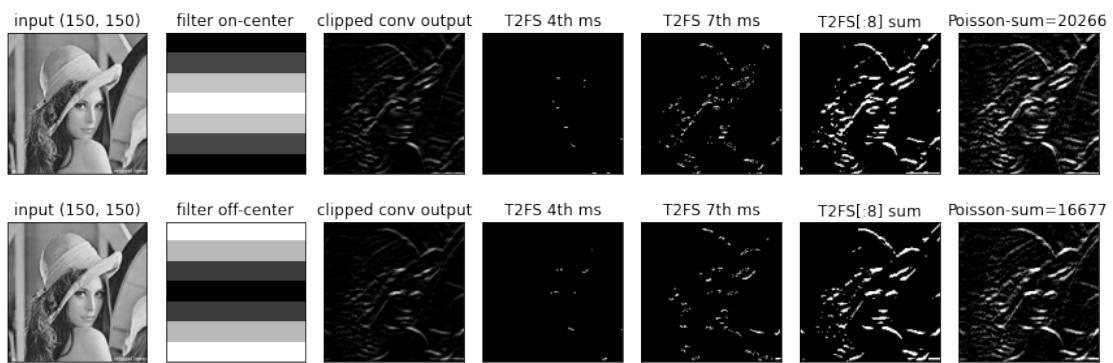
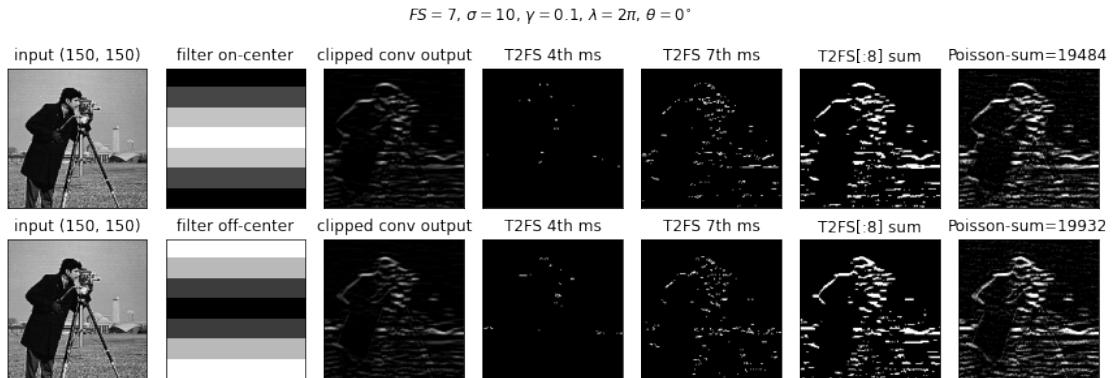
$$FS = 23, \sigma_1 = 2.5, \sigma_2 = 23$$

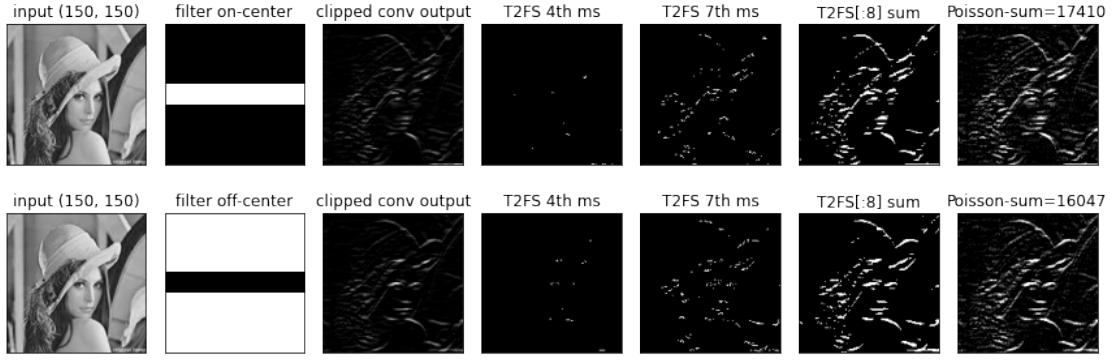


From the above figures, we can conclude that Time2FirstSpike encoder is encoding the parts of the images where it is more similar to the filter earlier, and the parts where it is less similar to the filter in the later time-steps. The parts of the image that is not similar to the filter will be encoded in the last time-steps, and they contain no useful information. If we sum the output of T2FS through time until the 8th time-step, we see that the actual input (the clipped output of convolution) is being created. In Poisson encoder, all the information is present in all time steps, but the intensity is low. If we sum the outputs of the Poisson Encoder in all time-steps, we see that the input is being created. Compared to the T2FS encoder, Poisson encoder's output contains slightly more details which might not be very useful for tasks like classification.

## 3.2 Gabor Filters

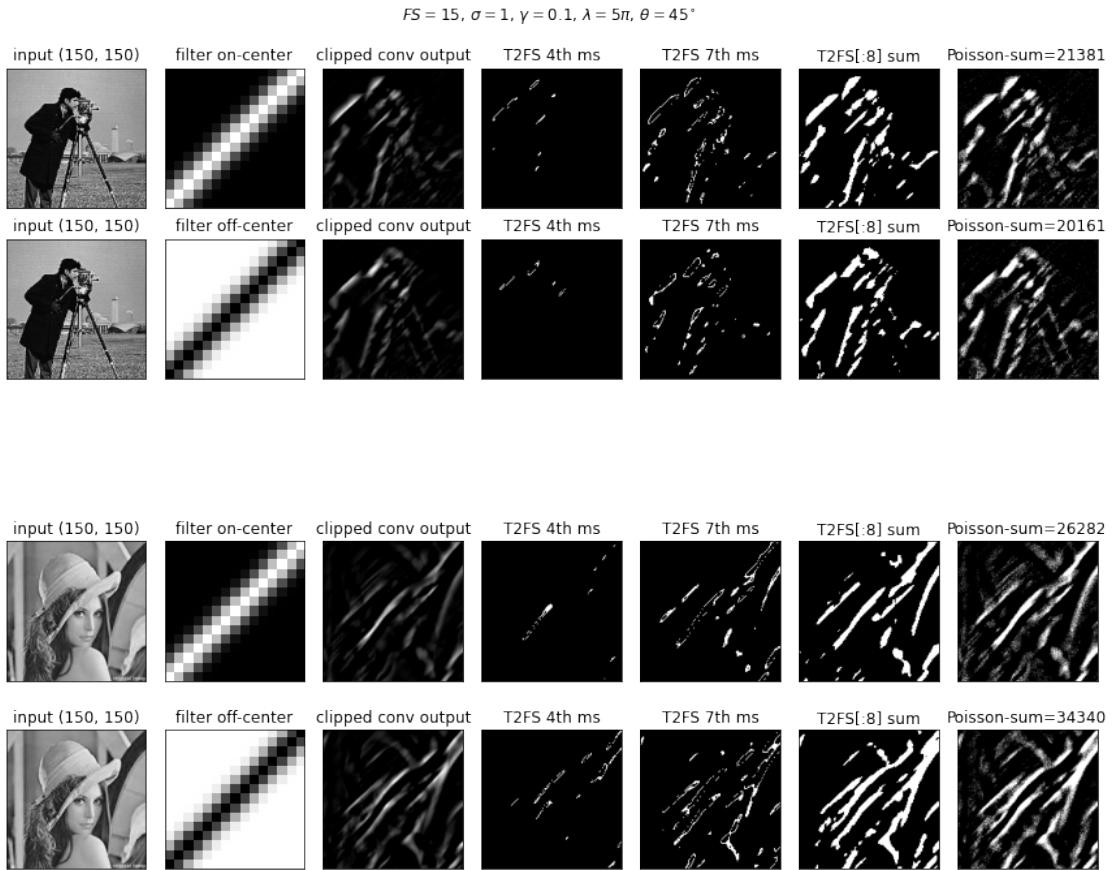
### 3.2.1 Effect of $\sigma$



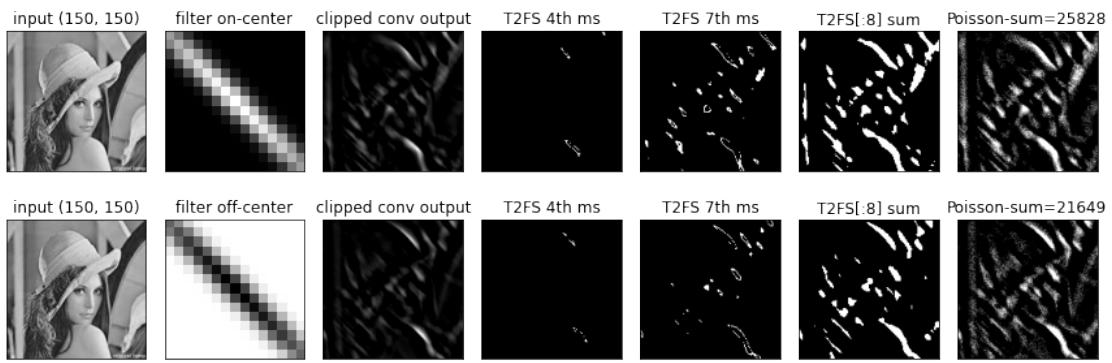
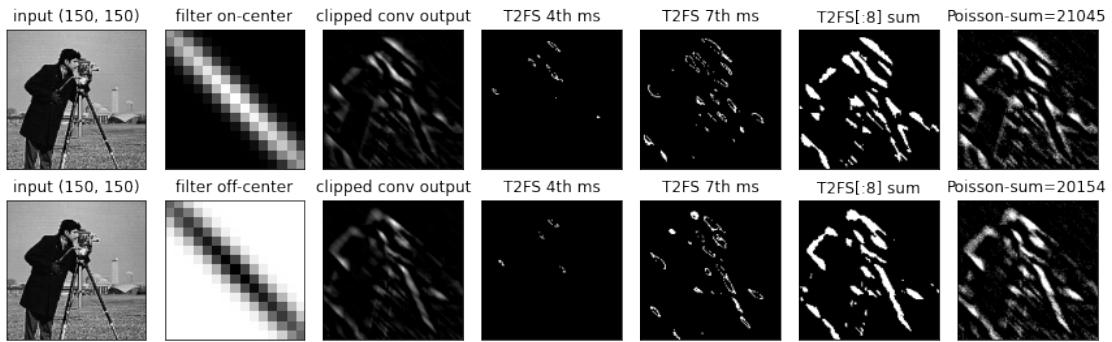


In the above plots, we see the effect of  $\sigma$ . Higher sigma indicates smoother changes in the filter. Since in real images the lines are not as sharp as what you see in the second filter, the output of the first filter has more spikes as shown in poisson encoder output. Also, none of the images have very intense horizontal lines, so we do not see many spikes in T2FS encoder output in the early time steps.

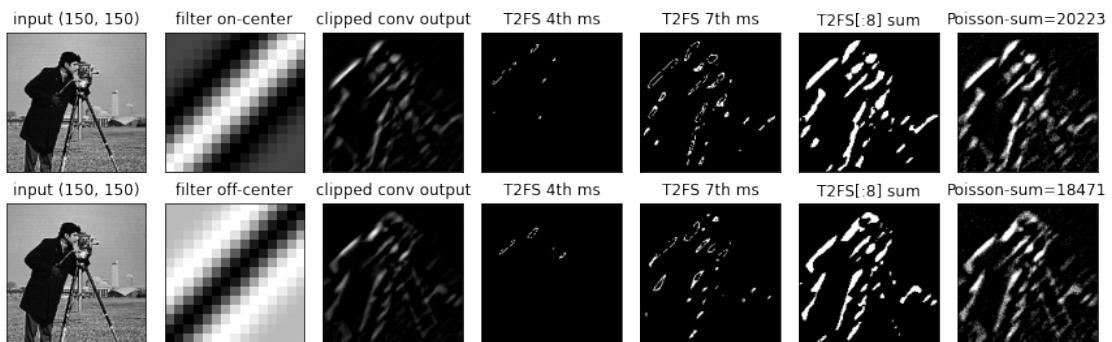
### 3.2.2 Effect of $\theta$

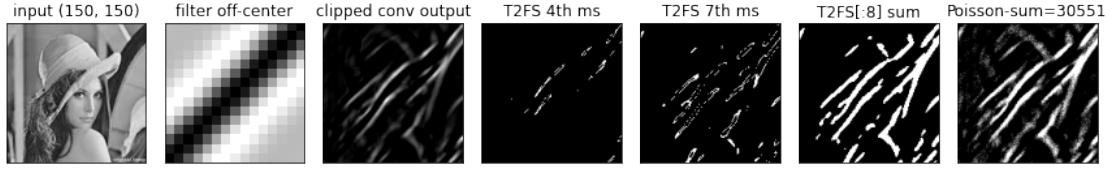
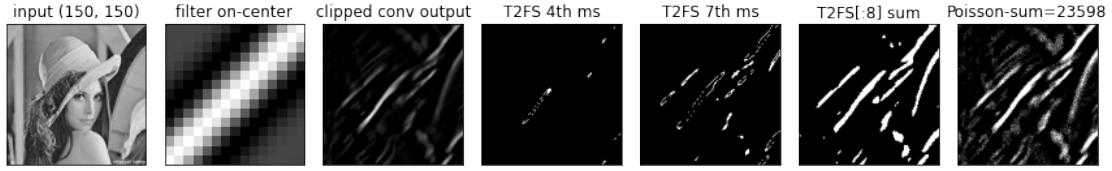


$$FS = 15, \sigma = 1, \gamma = 0.1, \lambda = 5\pi, \theta = 135^\circ$$

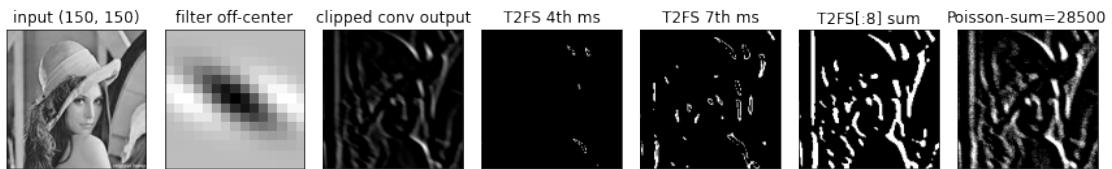
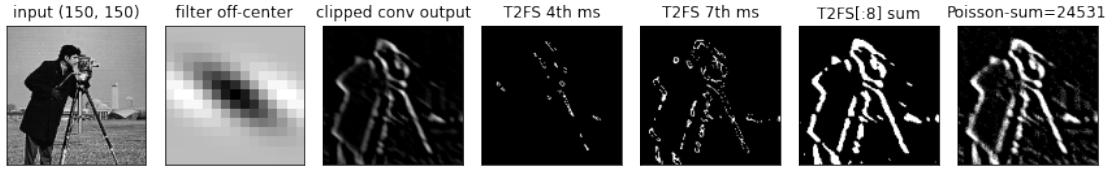
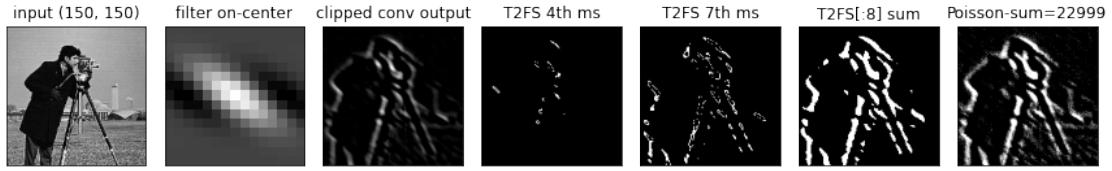


$$FS = 15, \sigma = 3, \gamma = 1, \lambda = 3\pi, \theta = 45^\circ$$





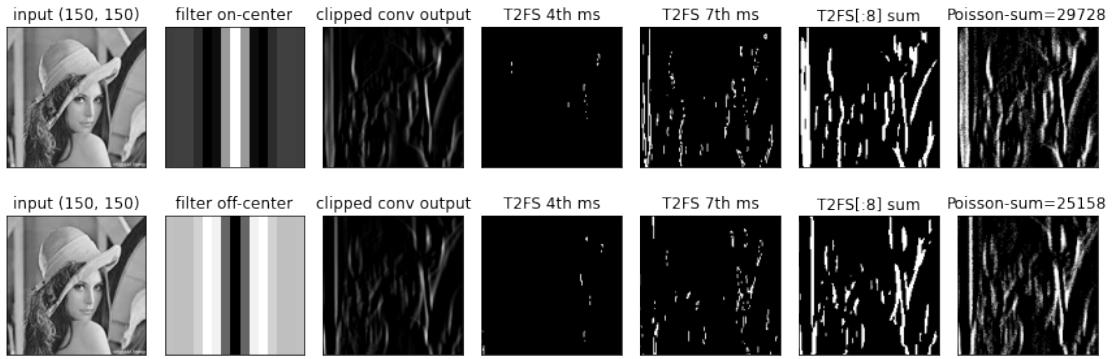
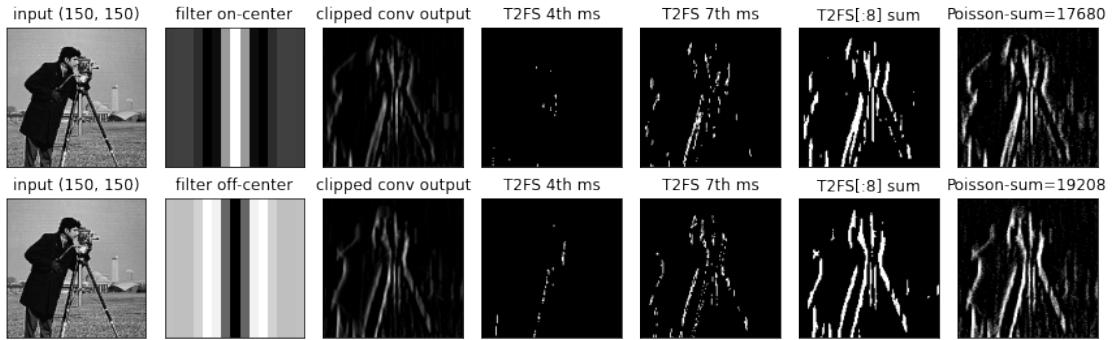
$FS = 15, \sigma = 3, \gamma = 1, \lambda = 3\pi, \theta = 135^\circ$



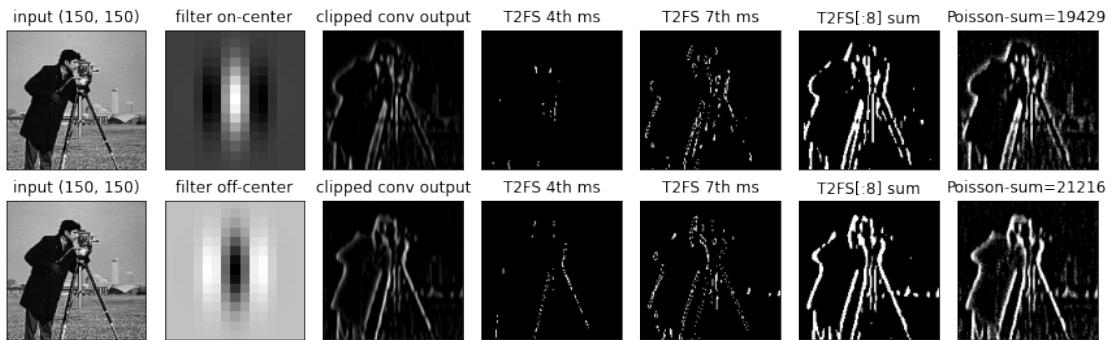
When we use a  $135^\circ$  filter on the woman image, we see a low amount of spikes in the early time-steps of T2FS encoder. The reason is that there are not great matches for that filter in the image.

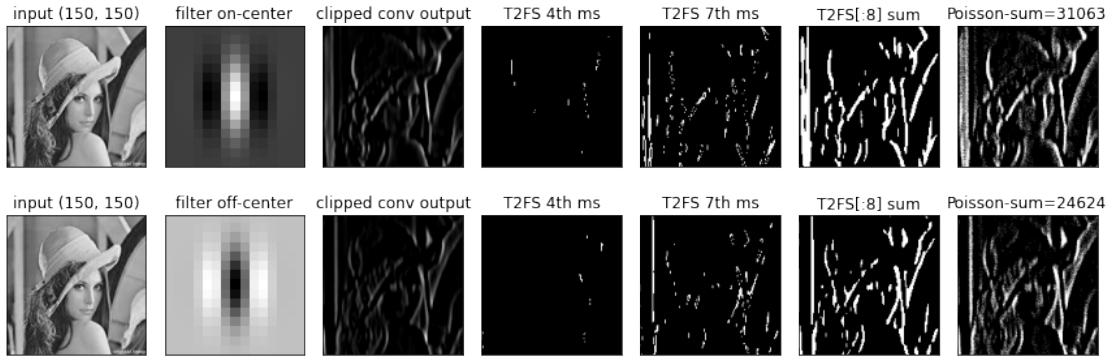
### 3.2.3 Effect of $\gamma$

$$FS = 15, \sigma = 2, \gamma = 0.01, \lambda = 2\pi, \theta = 90^\circ$$

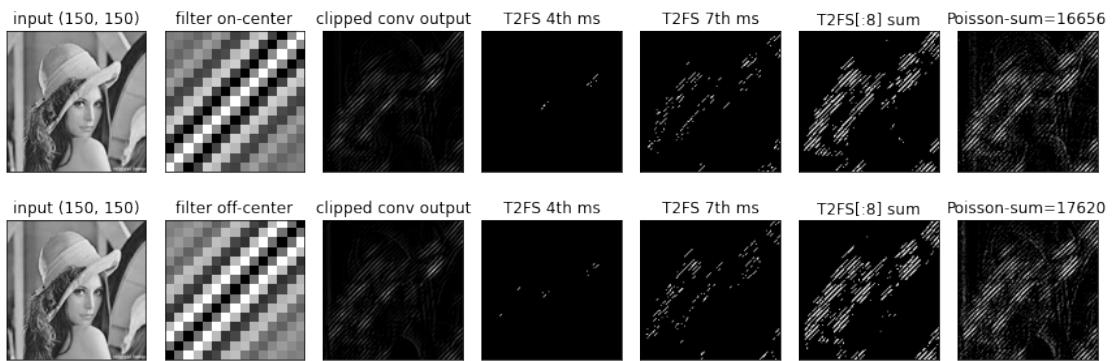
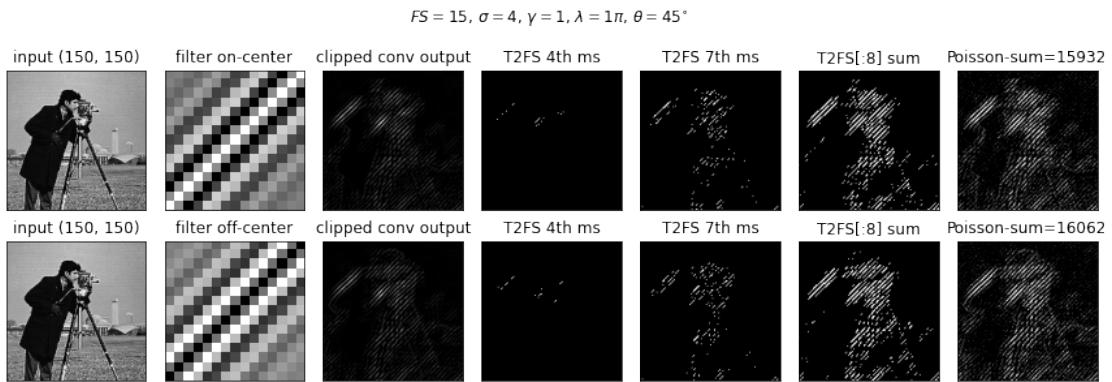


$$FS = 15, \sigma = 2, \gamma = 0.75, \lambda = 2\pi, \theta = 90^\circ$$

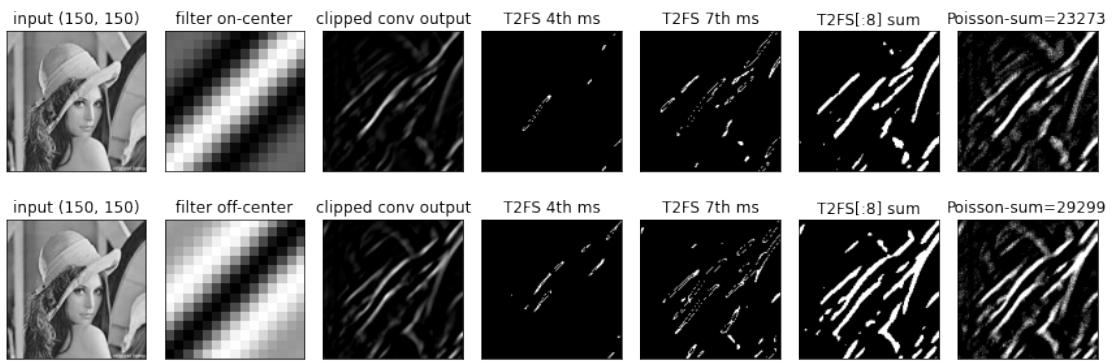
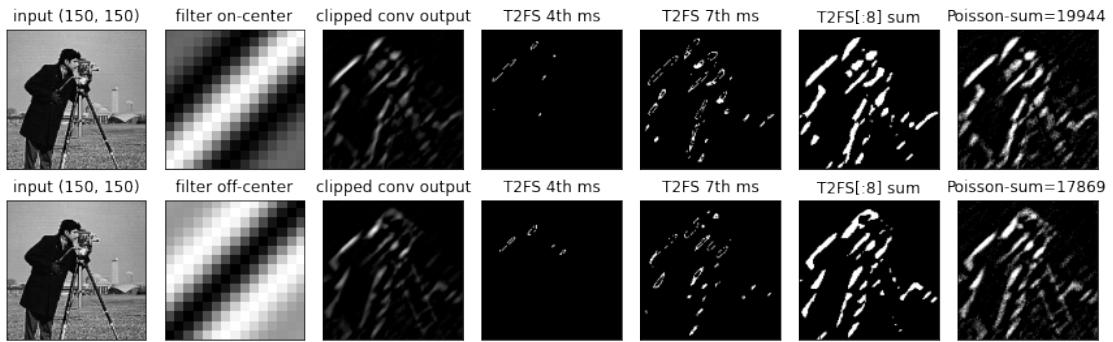




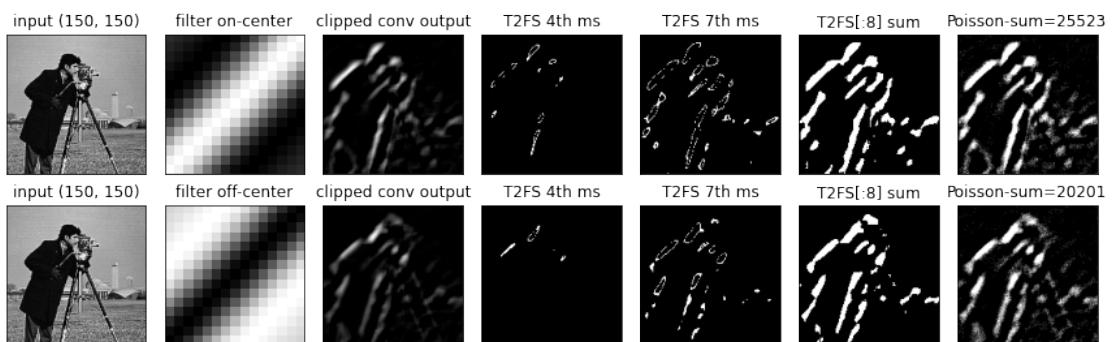
### 3.2.4 Effect of $\lambda$

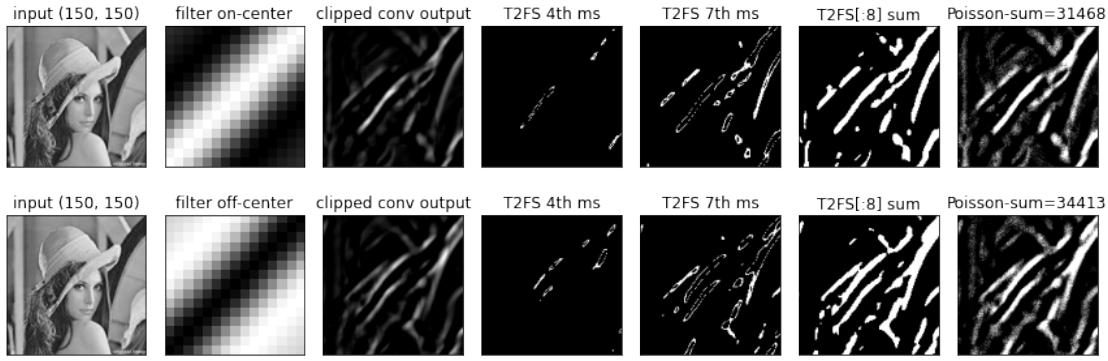


$$FS = 15, \sigma = 4, \gamma = 1, \lambda = 3\pi, \theta = 45^\circ$$

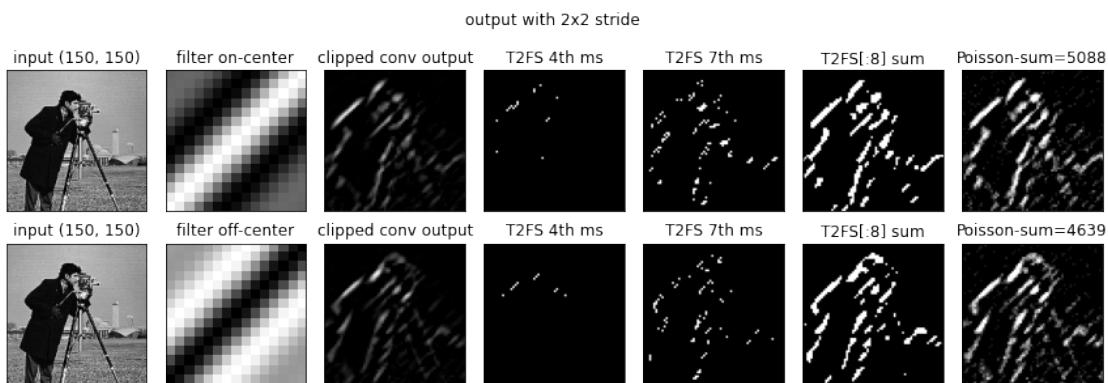
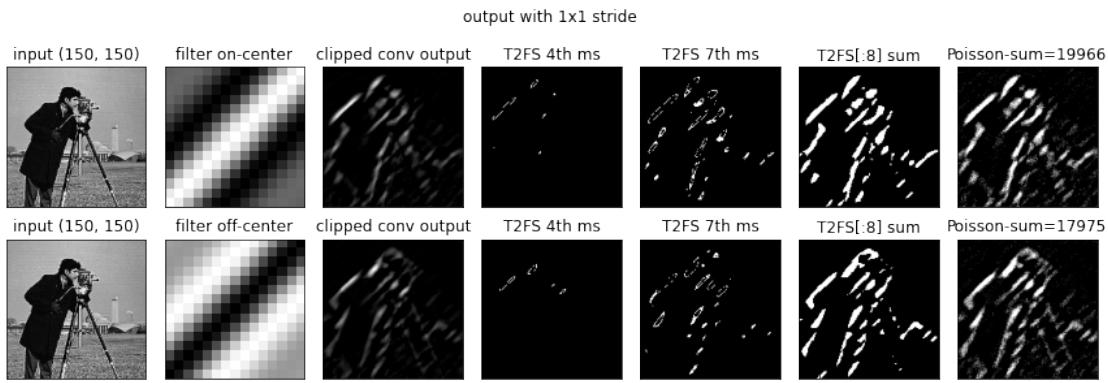


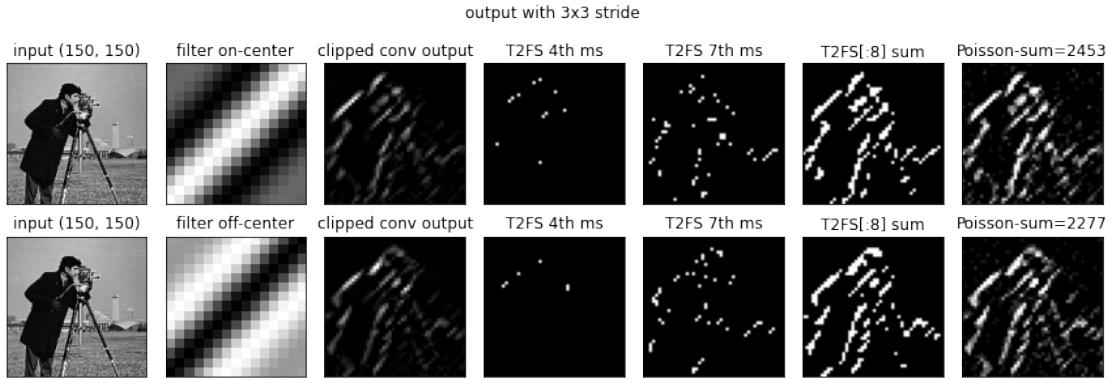
$$FS = 15, \sigma = 4, \gamma = 1, \lambda = 5\pi, \theta = 45^\circ$$





### 3.3 Effect of Convolution Stride





By applying different strides, we see that the output's pattern is not changing significantly. By increasing the strides, some sort of down-sampling is taking place, so output's shape is reduced. Only some unimportant details will be lost by increasing stride.

# Chapter 4

## Summary

1. By using filters with larger size, we are increasing receptive field and patterns (dots and lines) with larger size will be found in images. If we increase the filter size too much, the output will not be informative.
2. Both encoders tend to encode the image through time and if we sum the encoding outputs through time we will get the encoding input.
3. `Time2FirstSpike` encoder encodes parts of the image where its more similar to the filter in earlier time steps and parts with less similarity in later time-steps. Parts of the image that are not alike the filter will be encoded in the last time steps, so they can be ignored completely. The reason for this behaviour is the fact that parts of image with more similarity to the filter will have more intensity in the filtering output and `T2FS` encoder will encode pixels with more intensities in earlier time-steps.
4. `Poisson` encoder encodes the filtering output in all time steps almost equally, but it is a bit noisy in each time-step. By summing encoding output in all time-steps, the noise's effect will be diminished, and we can see the input clearly in the encoding output.
5. Total number of spikes using Poisson encoder depends on two factors:
  - Input size: larger input size will produce more spikes
  - Similarity of the filter with the input image: If the input image is similar to the filter, the output of convolution contains pixels with more intensity. This high intensity increases the number of spikes in the encoding output.
6. By increasing the strides, the output of the encoding will be affected significantly as long as the input size is big enough. Convolving with strides larger than 1, downsamples the input.