project-5

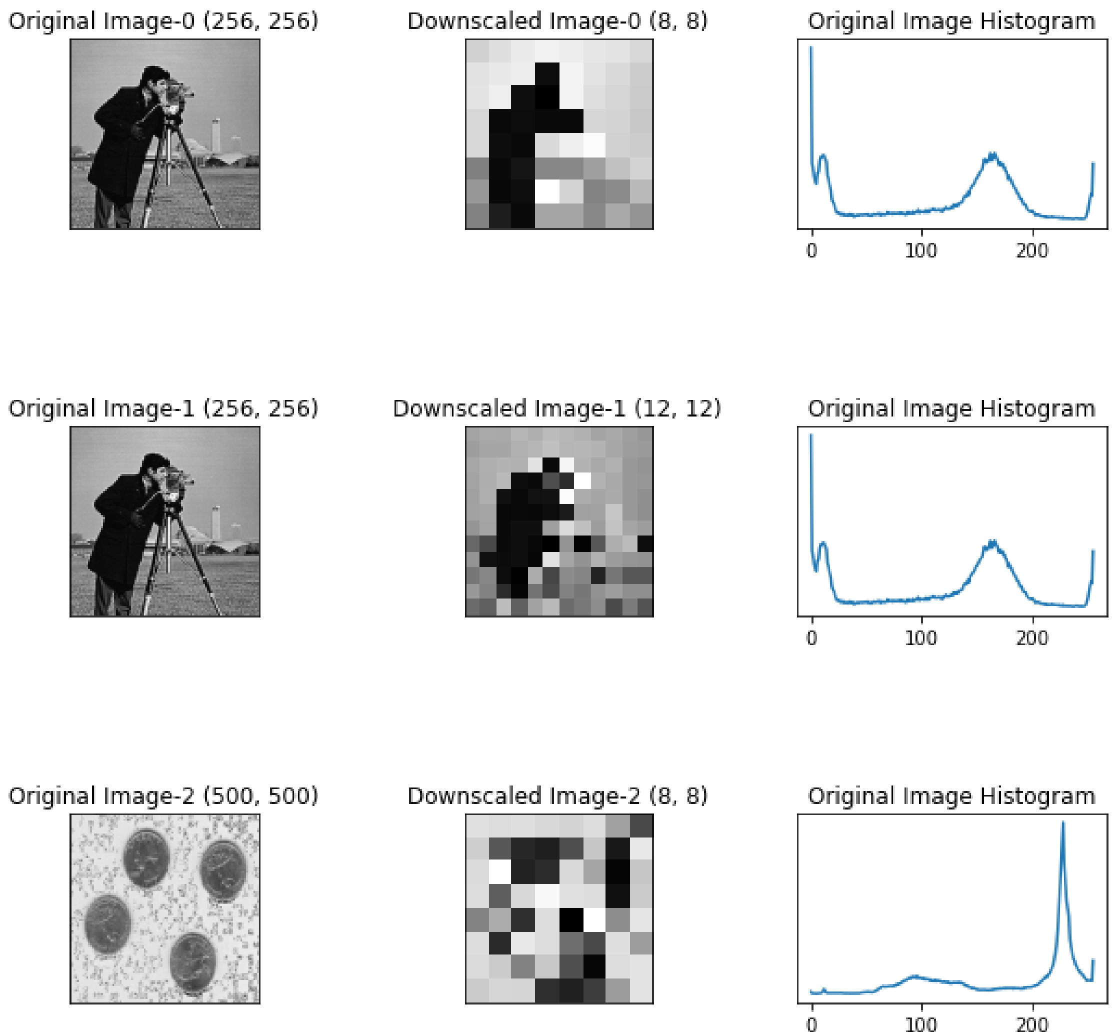
May 7, 2021

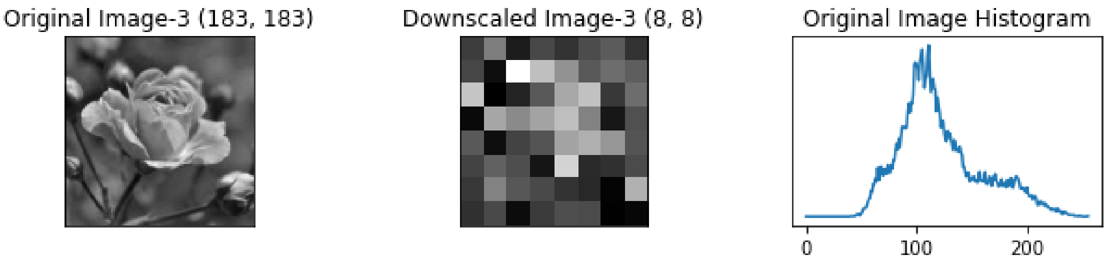
Contents

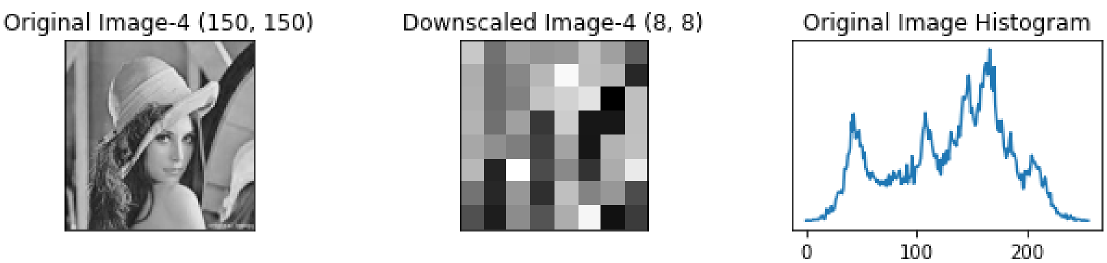
Chapter 1

Introduction

In this project we will demonstrate and compare different encoding methods on a set of 5 images as shown below. The first and second images are only different in their size to inspect the image scale effect. The images' histograms are plotted as well.







In the following we test three different encoding methods:

1. **Time to First Spike Encoding**
2. **Poisson Encoding**
3. **Numerical Encoding**

Chapter 2

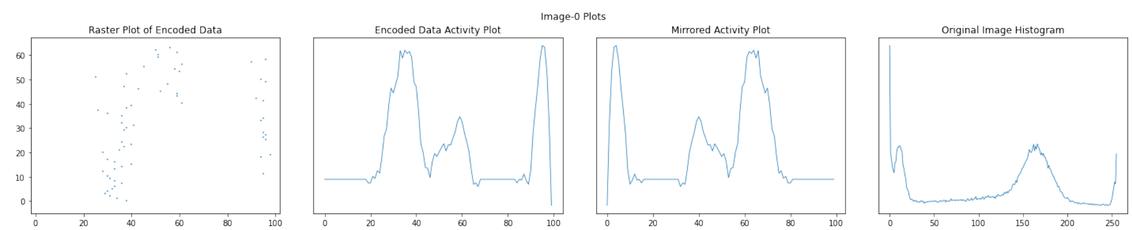
Time2FirstSpikeEncoder

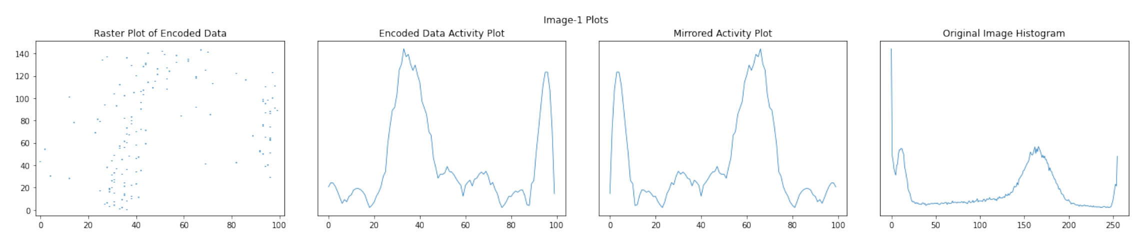
2.1 Experiment #1 (Overall Behaviour)

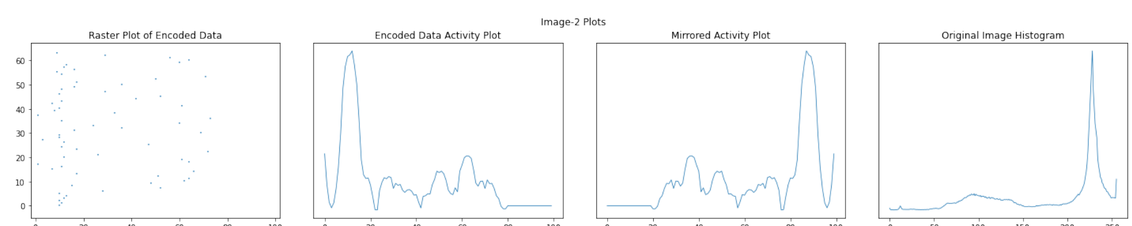
$Time_{encode} = 100ms$

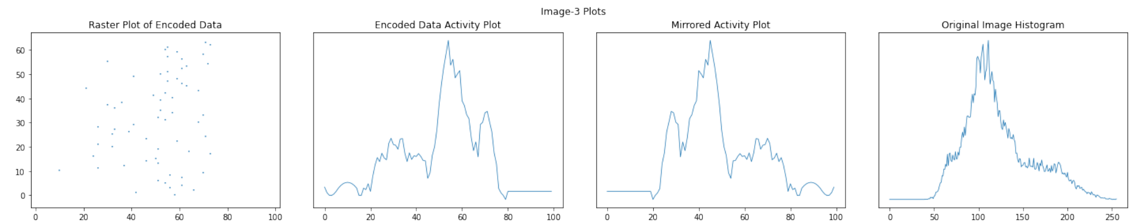
In the first experiment we additionally draw the mirrored activity plot, and the original image's histogram to show that the mirrored activity plot (extracted from a 8x8 image) is very similar to the image's histogram with original size. This is quite interesting; although we are using a very small 8x8 image, the whole image's statistics are encoded into a sparse set of spikes. However, a large amount of information is lost in the process.

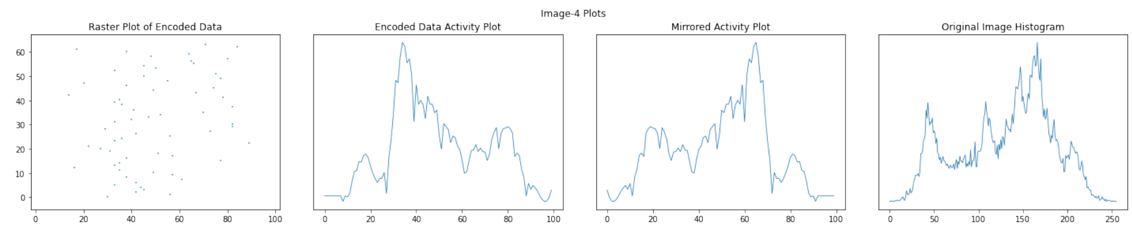
Note: In the following plots, each row represents an image and columns are either showing the effect of changing an encoder's parameter or different plots of that image.









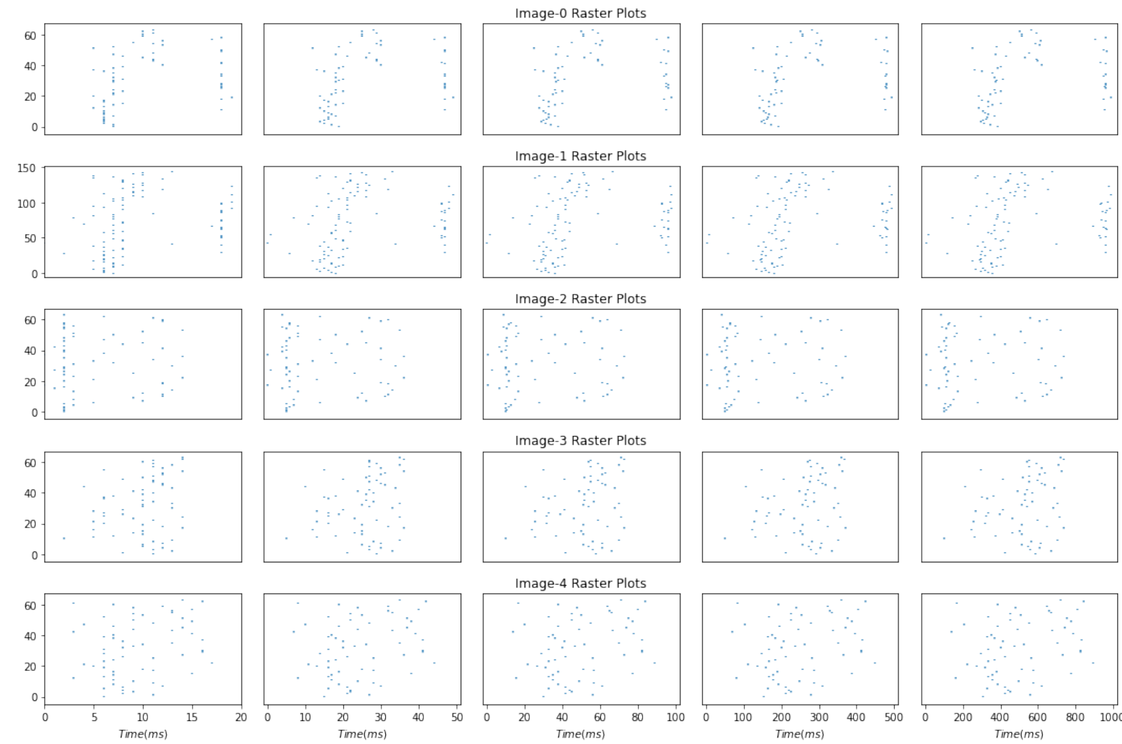


In the first image, dark pixels (values close to 0) are prevalent. So, in the raster plot we see that a large portion of neurons spike at the end of the raster plot. The other peak in the image's histogram could be seen in the raster plot as well. This explanation is also true for the other images.

2.2 Experiment #2 (Encoding Time)

Now, we test different values for $Time_{encode}$: $[20, 50, 100, 500, 1000]$

Each column represent one T_{encode} and each row represents one image.



From the above plots we conclude that if the encoding time becomes greater than a certain value (about $20ms$ in our experiments), the output of encoding will have the same shape regardless of the encoding time. To select a particular encoding time, we should consider the sparsity of the activity we need in the next layer of our network. By increasing the encoding time, the same number of sparks will be distributed in more time. If the time becomes too long, the effect of the spikes might not be sensed in the following layers, as neurons will decay to their resting potentials exponentially.

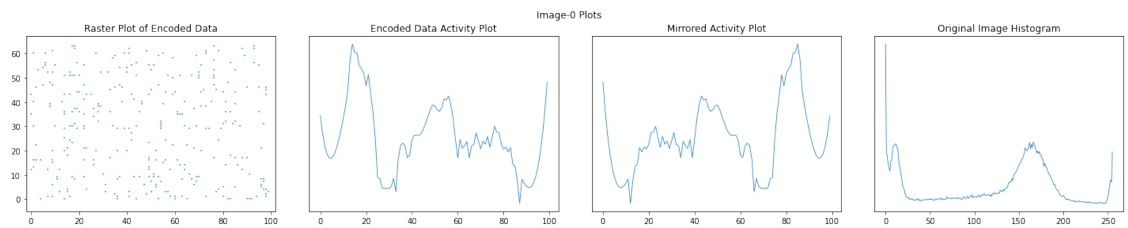
Chapter 3

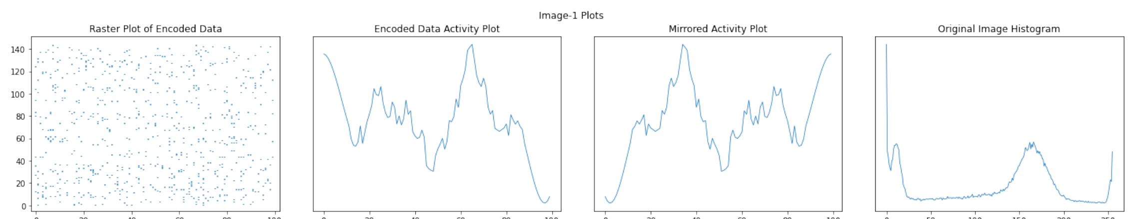
PoissonEncoder

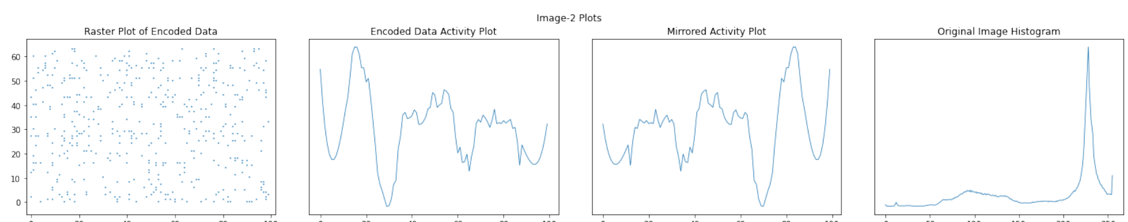
In experiment #1 and #2, $r_{max} = 10$, and in experiment #3, the effect of changing this parameter is discussed.

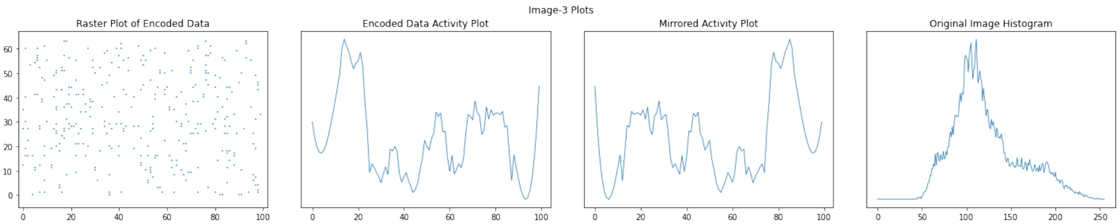
3.1 Experiment #1 (Overall Behaviour)

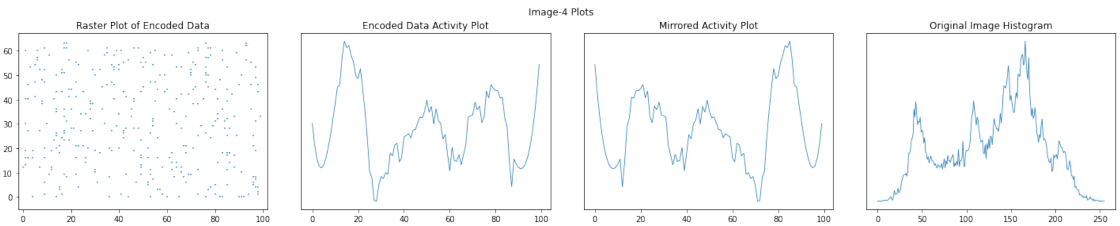
As we did for Time2FirstSpikeEncoder, we draw activity plot and its mirror, along with the original's image histogram.







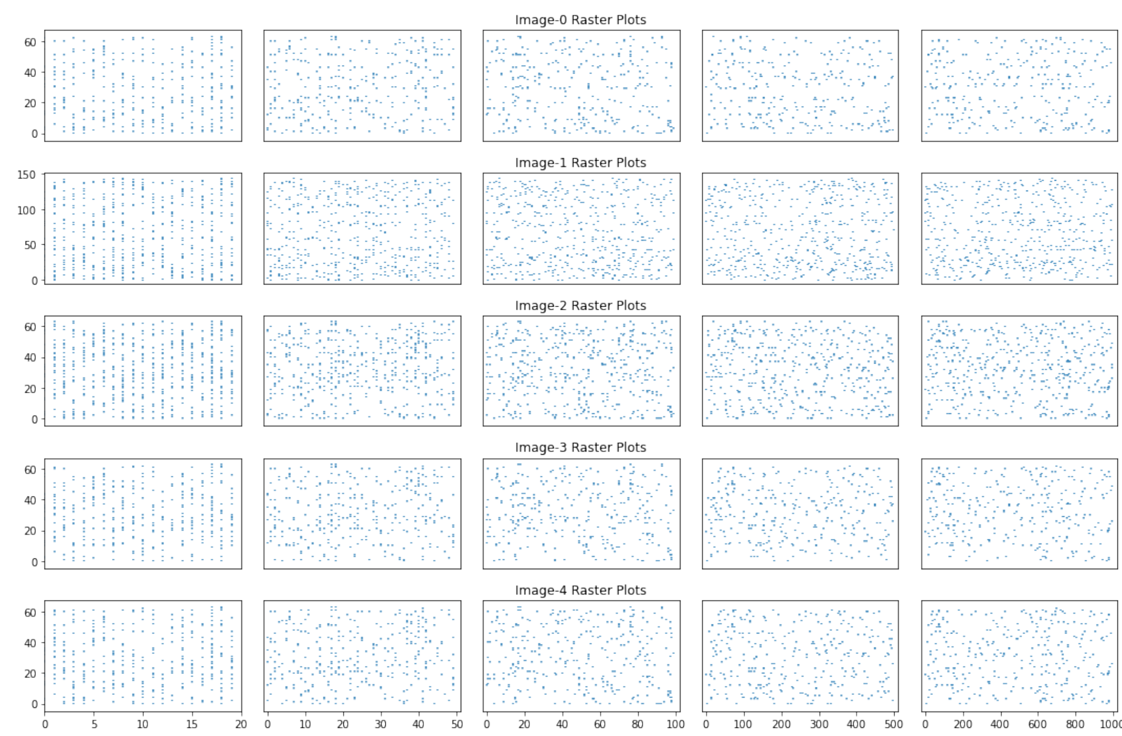




By observing the above plots we see that the raster plots are not as sparse as they are in Time-to-First-Spike-Encoding. As a result, the activity plots are more noisy compared with the previous encoding method and less similar to the original images' histograms. The good thing however, is that the spikes are distributed in time more uniformly; so, the amount of lost information is lower than the previous encoding method.

3.2 Experiment #2 (Encoding Time)

Now, we test different values for $Time_{encode}$: $[20, 50, 100, 500, 1000]$

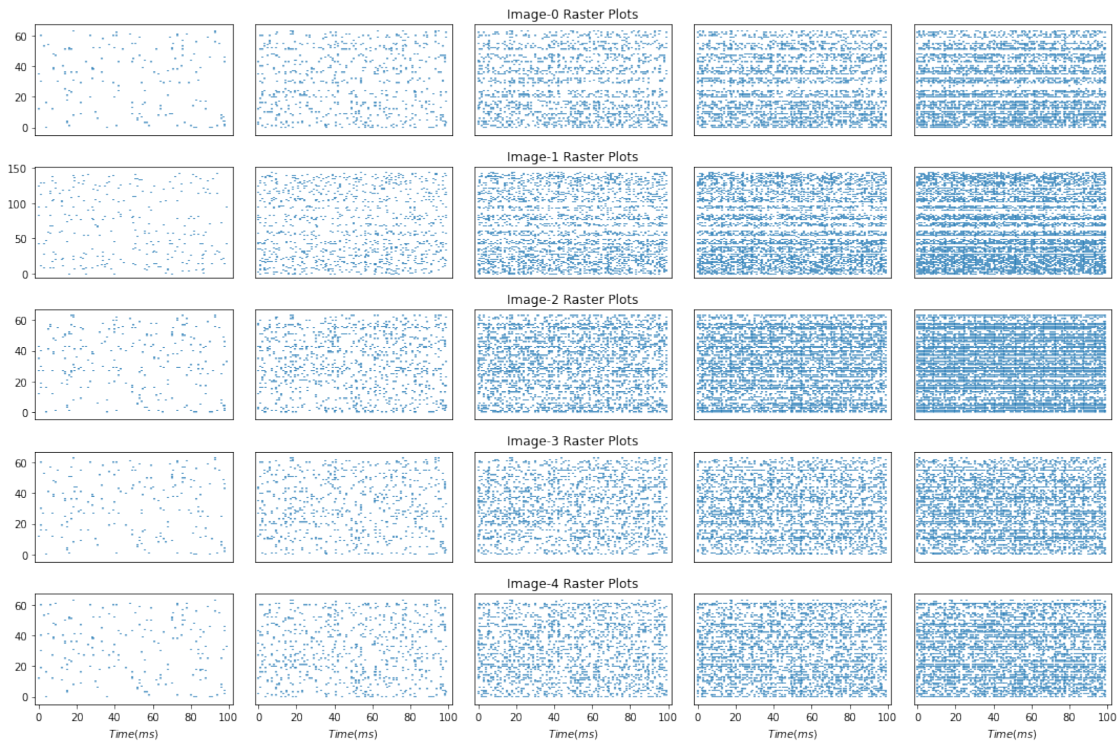


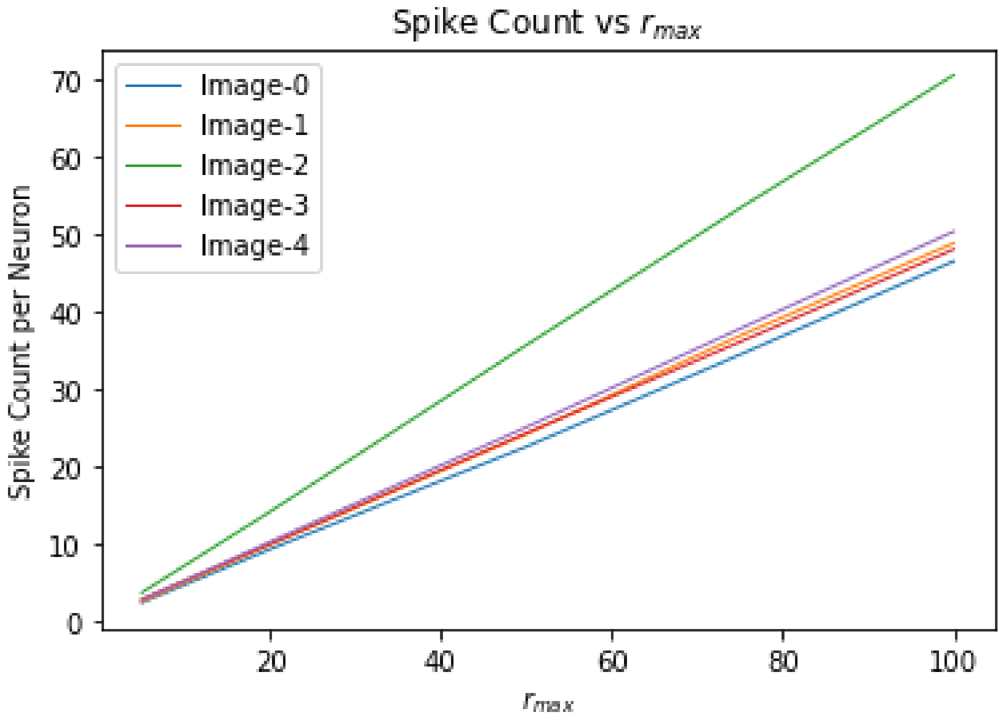
Again by increasing the encoding time, after a certain encoding time, the overall shape of the raster plots are preserved, and only the sparsity of the spikes increases. If the encoding time is too short, the encoder does not generate a good representation of data.

3.3 Experiment #3 (r_{max})

Here, we fix the encoding time to $100ms$, and test the results of the encoding with the following values of r_{max}: $[5, 20, 50, 75, 100]$

Each row represents an image, and each column represents r_{max}.





We see that by increasing the r_{max} the total amount of spikes will increase linearly. The slope of the line depends on the image's histogram. The image with more pixels' of high intensity has the line with higher slope (Image-2). The overall shape of the raster plots however, is not changing with changing r_{max}.

Chapter 4

PositionEncoder (NumericalEncoder)

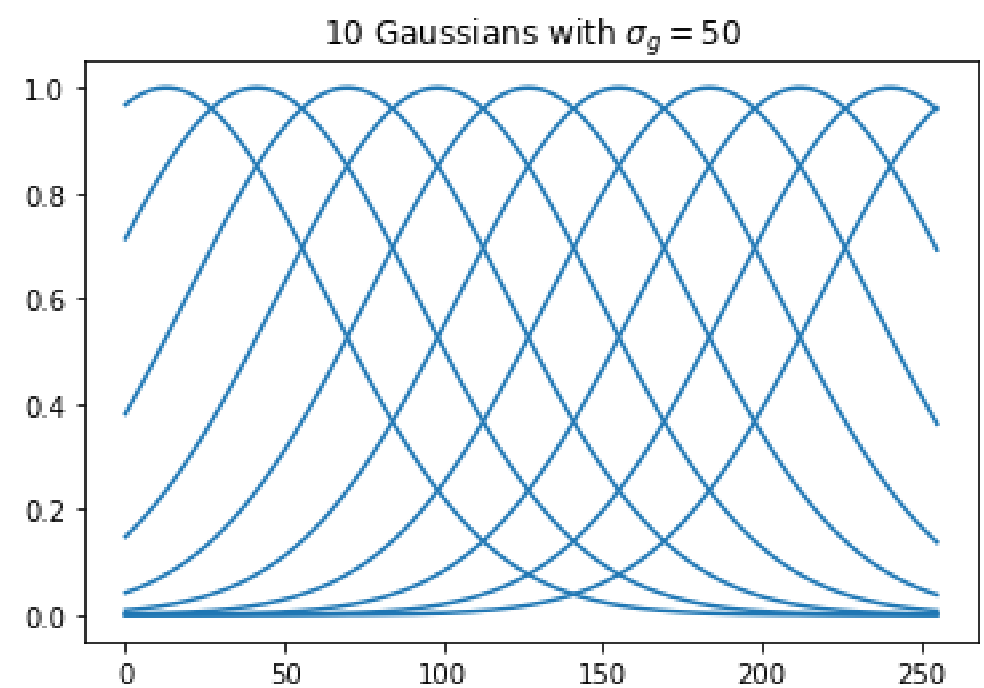
This encoder uses multiple gaussian functions in order to encode a certain value. If the value intersect with a gaussian with high height, it means that the neuron corresponding to that gaussian should spike faster than the others. Also, a threshold is set to determine which intersections should contribute to encoding. If the intersection's height is lower than this threshold, we ignore the spike for that neuron. This way, we ignore the gaussians whose contribution to the encoding is minimal. Otherwise, there would be a peak at the end of raster plots, as all neurons with low contribution will spike then. The threshold is fixed to 0.95 for all experiments. It means that the intersection point's height should be greater than 0.95 to be considered valid.

4.1 Experiment #1 (Overall Behaviour)

$n_{neurons} = 10$

$\sigma_{gaussian} = 50$

Since we have $n_{neurons} = 10$, the total amount of neurons in the raster plot for 8x8 images is: $8 * 8 * 10 = 640$. All the neurons corresponding to the same pixel, are put next to each other in the raster plots.



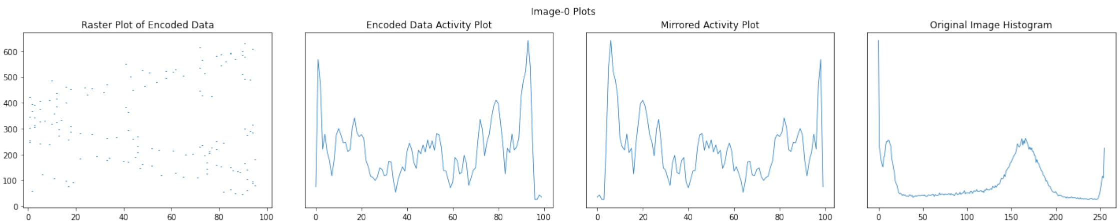


Image-0 Plots
Raster Plot of Encoded Data
Encoded Data Activity Plot
Mirrored Activity Plot
Original Image Histogram

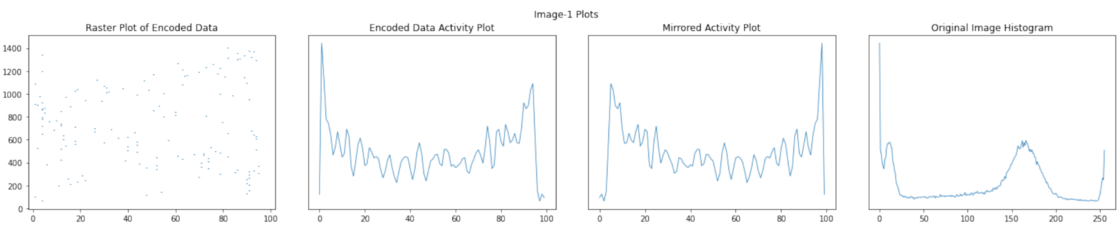


Image-1 Plots
Raster Plot of Encoded Data
Encoded Data Activity Plot
Mirrored Activity Plot
Original Image Histogram

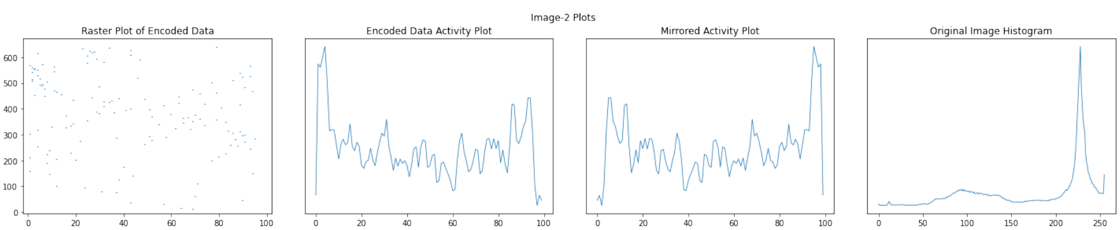


Image-2 Plots
Raster Plot of Encoded Data
Encoded Data Activity Plot
Mirrored Activity Plot
Original Image Histogram

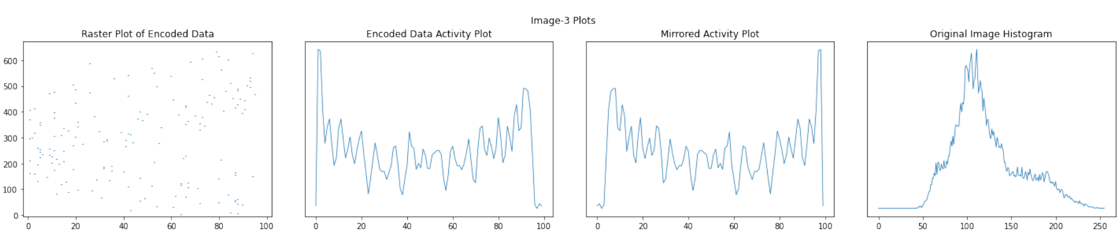


Image-3 Plots
Raster Plot of Encoded Data
Encoded Data Activity Plot
Mirrored Activity Plot
Original Image Histogram

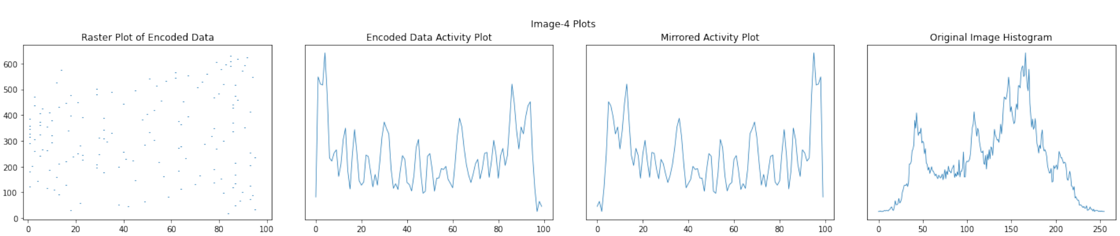


Image-4 Plots
Raster Plot of Encoded Data
Encoded Data Activity Plot
Mirrored Activity Plot
Original Image Histogram

The above plots indicate that unlike the two previous encoding methods, the activity plots are not similar to the histogram of image. This is due to the fact that this encoding method is inherently using different approach to encode the pixel intensities. Unlike Time2FirstSpike, the spikes are almost uniformly distributed in time.

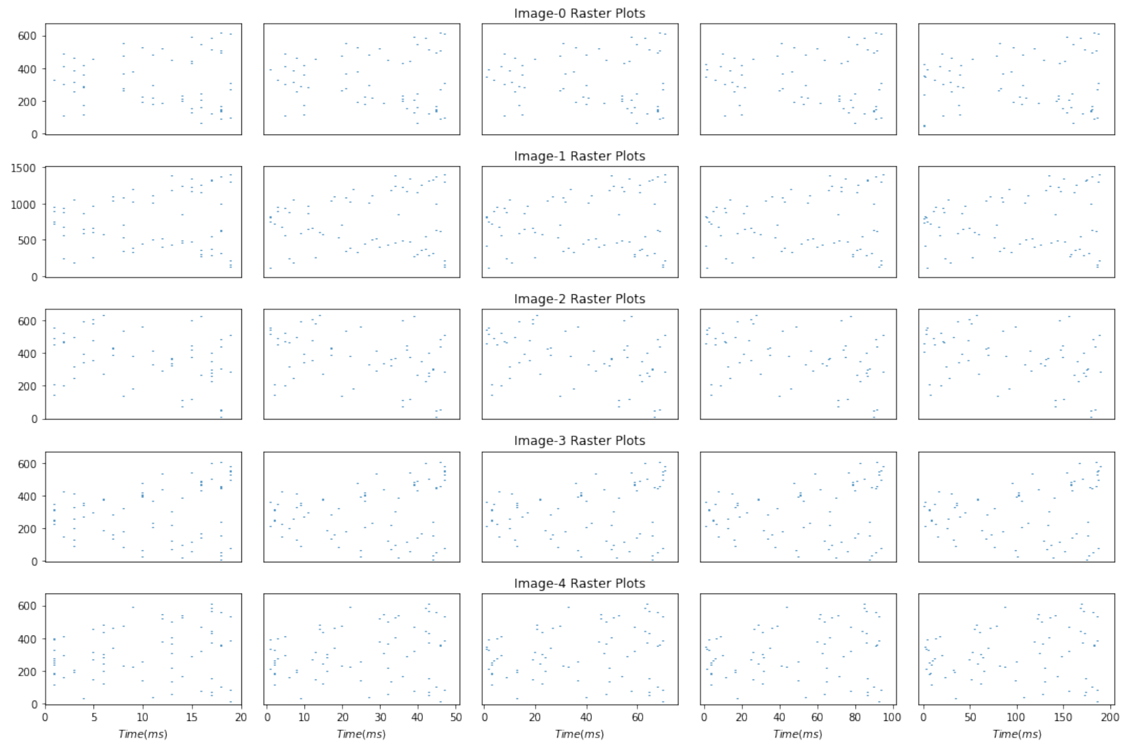
4.2 Experiment #2 (Encoding Time)

Now, we test different values for $Time_{encode}$: $[20, 50, 100, 200]$

Each row represent one image, and each column represent the encoding time parameter.

$n_{neurons} = 10$

$\sigma_{gaussian} = 50$



The result is the same with the other encoding methods when we tested changing the encoding time. Only the sparsity of the output is affected not the shape.

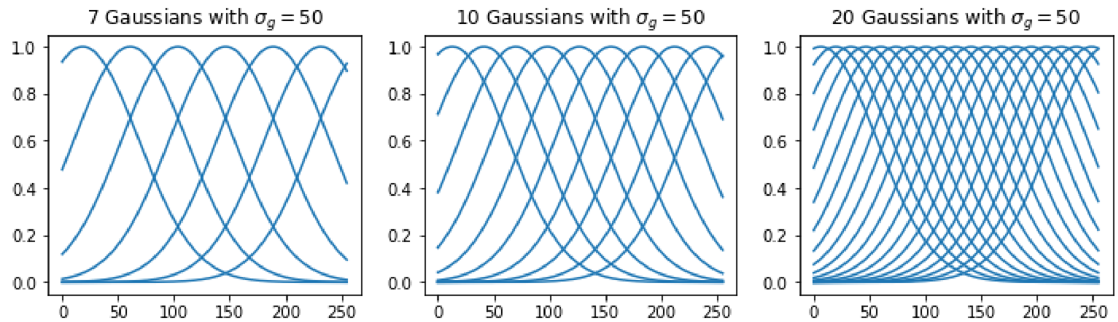
4.3 Experiment #3 (Number of Neurons)

Now, we test different values for $n_{neurons}$: $[7, 10, 20]$

Each column represents one of the tested parameters.

$Time_{encoding} = 50ms$

$\sigma_{gaussian} = 50$


7 Gaussians with $\sigma_g = 50$
10 Gaussians with $\sigma_g = 50$
20 Gaussians with $\sigma_g = 50$

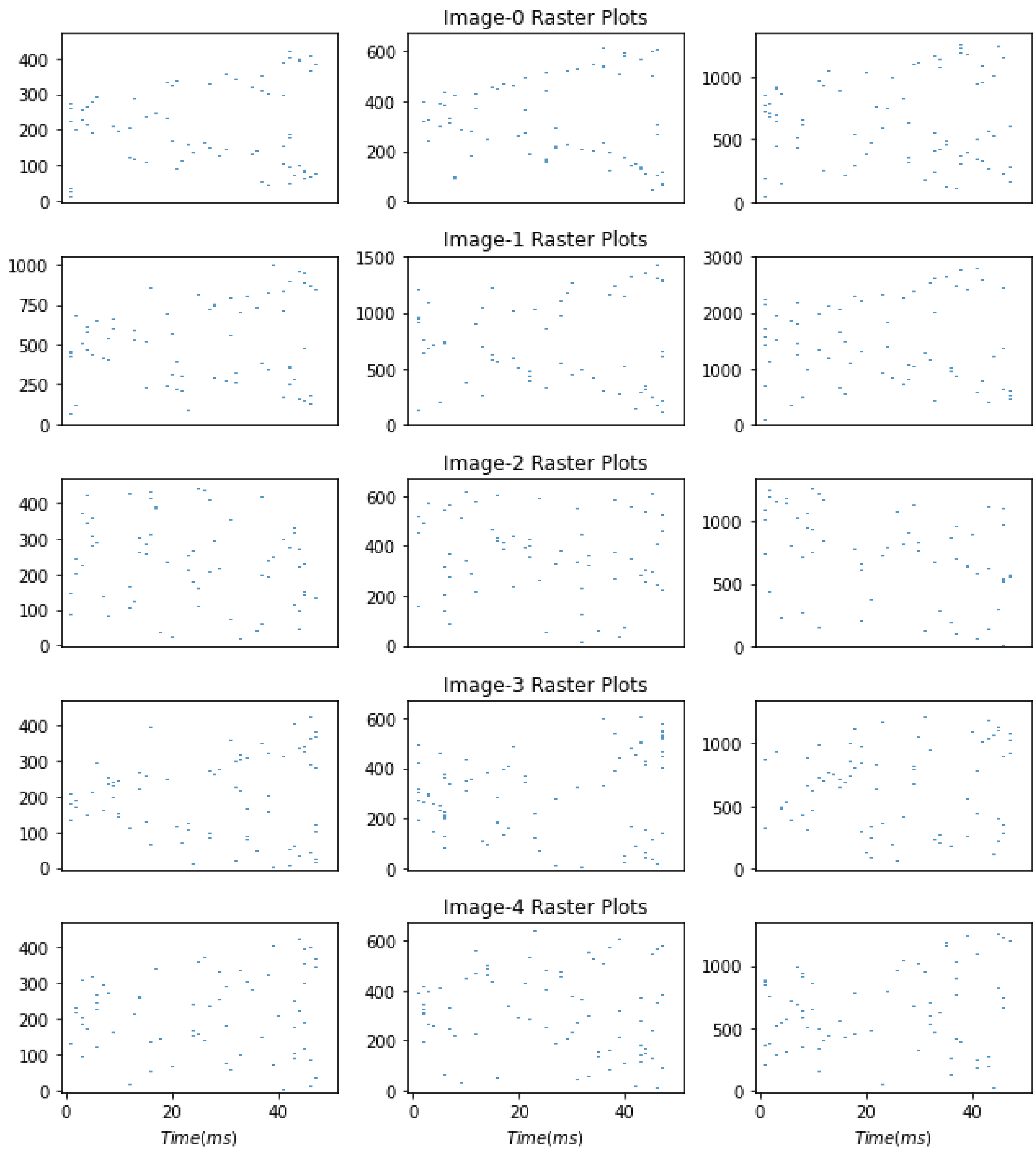


Image-0 Raster Plots
Image-1 Raster Plots
Image-2 Raster Plots
Image-3 Raster Plots
Image-4 Raster Plots
$Time(ms)$
$Time(ms)$
$Time(ms)$

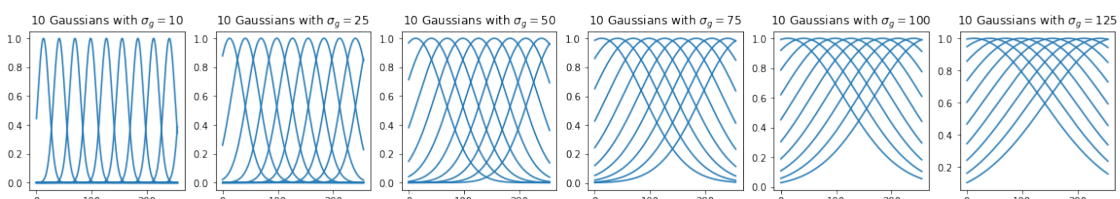
We see that the number of neurons increases the total spike count linearly in each case. The slop of the line depends on the input image. The shape of the raster plot changes slightly by changing the number of neurons, but the overall shape is preserved.

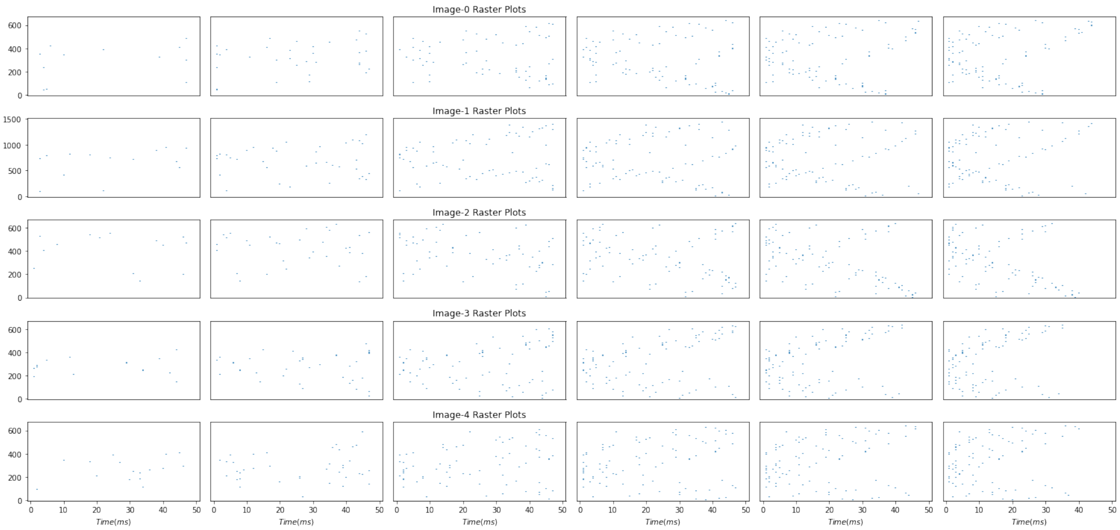
4.4 Experiment #4 ($\sigma_{gaussian}$)

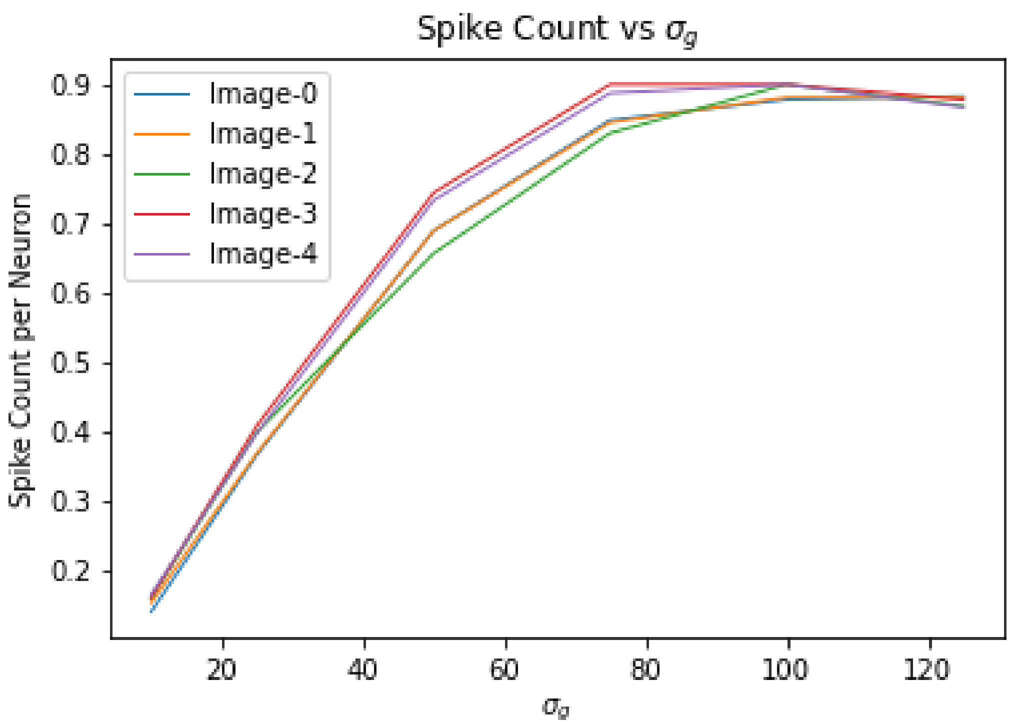
Each column represents a σ_g, and the rows represent the images.

$n_{neurons} = 10$

$time_{encoding} = 50ms$







From the above plots, we conclude that increasing σ_g, increases the number of spikes increases but not linearly. The slope of the plot vanishes after a certain point, and the plot will plateau. The overall shape of the plots however, are remained unchanged.

Chapter 5

Summary

We conclude the following results from all experiments:

1. Changing the $Time_{encoding}$ only affects the sparsity of the encoded data through time. Increasing this parameter will result in more sparse raster plot. The overall shape of the plots are preserved. This observation is the same for all three encoding methods.
2. The scale of the image does not affect the encoding besides that the number of neurons for encoding would be different. We calculated the spike counts per neuron for different images and saw that the first and second images spike count per neuron are exactly the same. These two images are only different in their size. (8x8 vs 12x12)
3. In PoissonEncoder, increasing r_{max} increases the spike counts linearly. The shape of the raster plot is preserved.
4. In PositionEncoder, increasing $n_{neurons}$ increases the spike count linearly. The shape of the raster plot is preserved.
5. In PositionEncoder, increasing $\sigma_{gaussian}$ increases the spike count in a non-linear manner. The slope of the spike-count vs $\sigma_{gaussian}$ plot will converge to zero slowly (the plot will plateau).
6. In Time2FirstSpikeEncoder and PoissonEncoder, the mirrored plot of output activity is mimicking the original image's histogram. The Time2FirstSpikeEncoder mirrored activity is more similar to the histogram compared with PoissonEncoder.
7. Time2FirstSpikeEncoder and PoissonEncoder tend to encode the statistics of the image (similar to histogram) throughout the time. The PositionEncoder encodes the intensity of each pixel in time.