VIP: Assignment 2

Mostafa Elshamy, Tom Vig, Andreas Hammer

November 2024

1 Gaussian Filtering

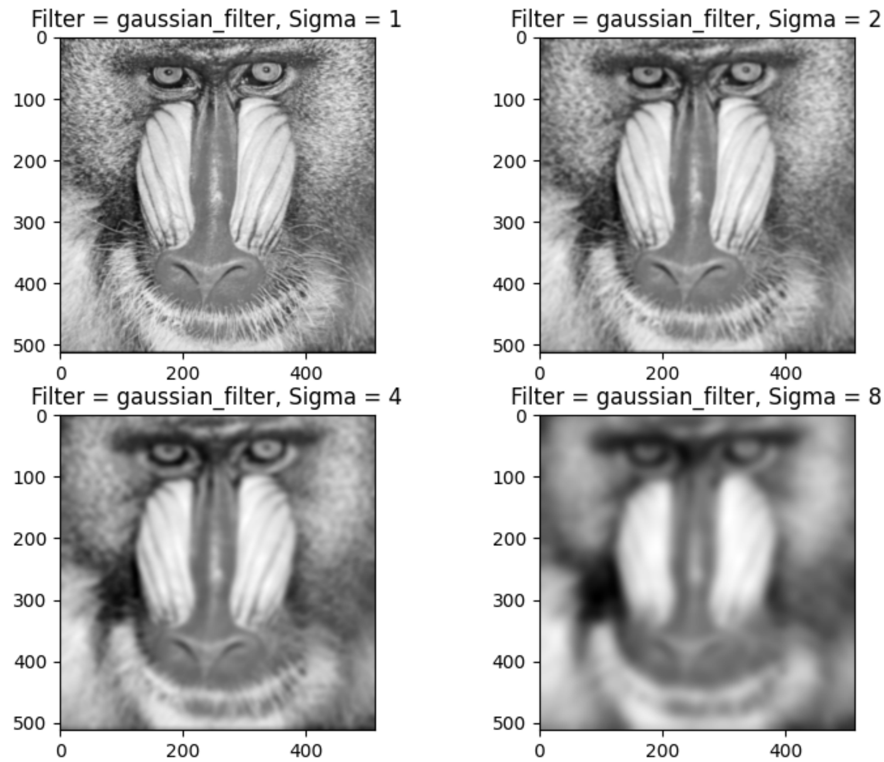

Figure 1: Gaussian filtering results for varying $\sigma = 1, 2, 4, 8$

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Figure 1 demonstrates the effect of different σ values on the Gaussian filtering of the 'Mandrill.jpg' image provided. Gaussian filtering is a method of image manipulation that smooths and blurs an image by averaging each pixel with it's surrounding pixels in a weighted manner. By increasing the σ values, the image becomes smoother and more blurred, with fine details being lost. This is because the σ value is responsible for the standard deviation of the Gaussian distribution, and by increasing it's value, we increase the size of the filter kernel, hence averaging each pixel with a larger group of surrounding pixels. The result is clear as we observe the transition in smoothness and blur between the top left image and the bottom right.

2 Gaussian Gradient Magnitude

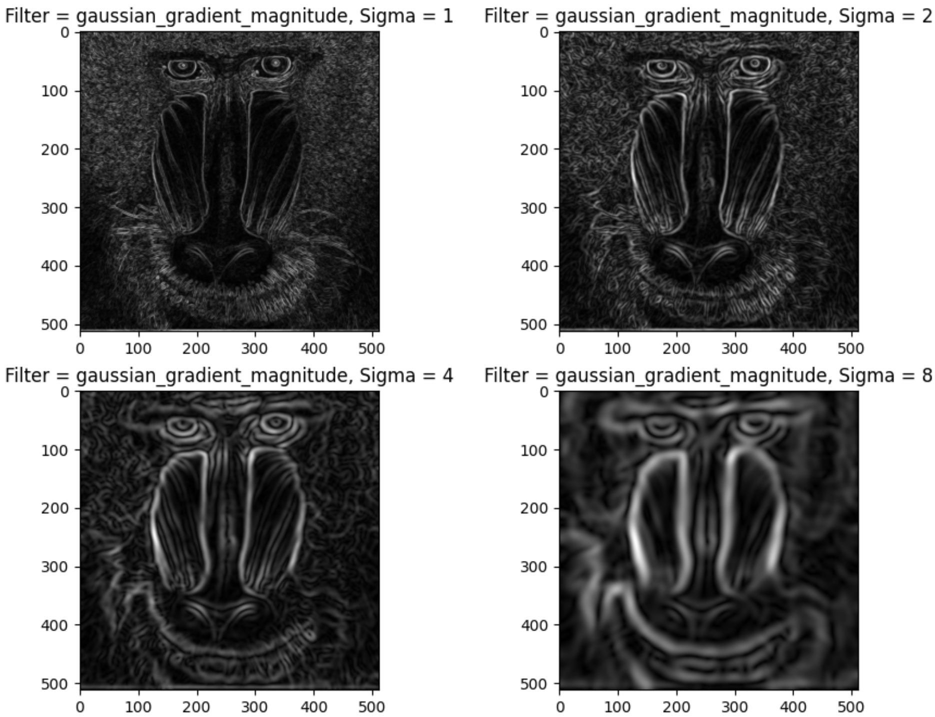

Figure 2: Gaussian filtering results for varying $\sigma = 1, 2, 4, 8$

The Gaussian Gradient Magnitude filter is based upon the previously discussed Gaussian Filtering method. However, this filter is interested in observing the rate of change of the intensity of pixels. This is achieved by computing the gradient of the Gaussian function. The purpose of displaying this rate of change in intensity is to draw attention to parts of the image at which sudden changes in intensity occur. The visual significance of this is edge detection. As figure 2 illustrates, all four images (regardless of σ values) perform edge detection.
The significance of varying σ is visually clear in the figure. As the σ value increases, the edge detection becomes more blurry, with less detail being picked up on, as well as a smoother looking image. The mathematical reasoning of this result proves to be quite similar to that of the Gaussian Filter. As σ, being the standard deviation of the Gaussian distribution, is increased, the filter kernel also increases in size. This larger kernel compares more pixels in the surrounding area when computing the weighted average for each pixel. This results in greater smoothing.

This is further built upon when computing the gradient of the Gaussian. Low standard deviation levels result in more fine-grained edge detection, with the opposite true for larger values. The bottom left image in the figure has the highest σ value, and it can be clearly observed that the Gaussian smoothing has caused a loss in accuracy but also a reduction of noise in the edge detection.

3 Laplacian-Gaussian Filtering

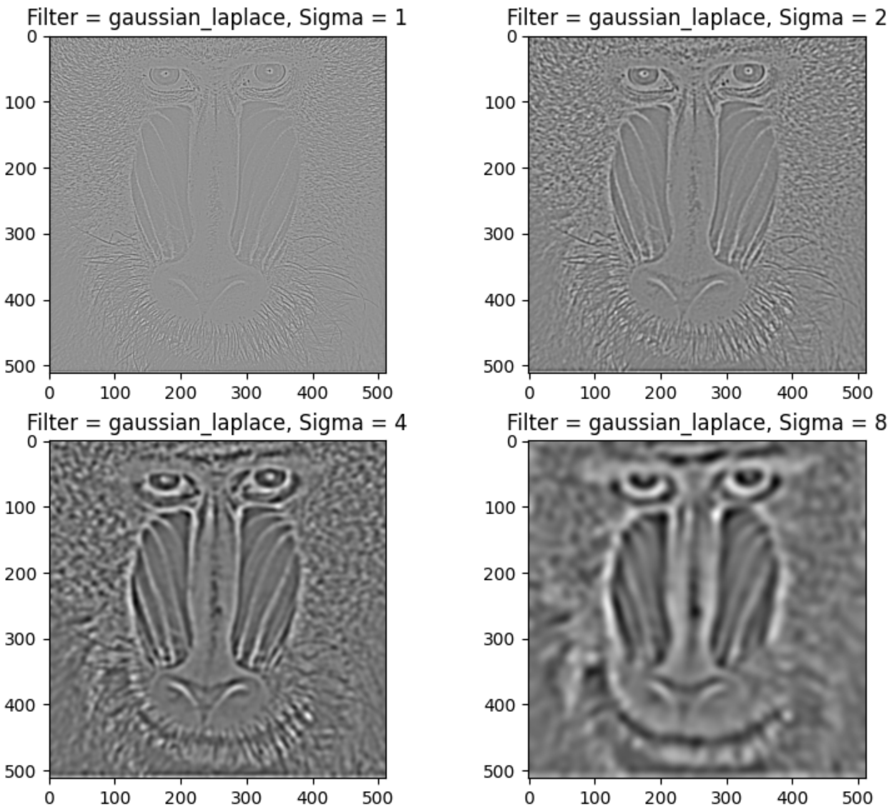

Figure 3: Laplacian of Gaussian filtering results for $\sigma = 1, 2, 4, 8$

The results shown in Figure 3 were produced using the `gaussian_laplace` function from the `scipy.ndimage` library. The kernel size for each axis is calculated by the following formula:

$$\text{Kernel Size} = 2 \times \lceil 4 \times \sigma \rceil + 1$$

(see: `https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.gaussian_filter.html`).

From the figure, we can see that increasing σ leads to a loss of fine details in the image. For $\sigma = 1$, intricate textures like the fur and subtle facial features of the mandrill are sharply visible (zooming in is recommended). At $\sigma = 2$, the details begin to blur, and edges become softer. By $\sigma = 4$, only pronounced features like the eyes and mouth stand out. Finally, at $\sigma = 8$, the image is significantly smoothed, with only the largest structures or "blobs" remaining. This progression highlights the LoG filter's effectiveness at detecting blobs, with higher σ values achieving "smoother" results.

The behavior of the LoG function as σ changes is similar to the Gaussian distribution. As illustrated in Figure 4, increasing σ reduces the concentration at the center. Kernel-wise, lower σ values mean that each pixel is less influenced by its neighbors during convolution, resulting in less smoothing. On the other hand, higher σ values increase the influence of neighboring pixels, producing a stronger smoothing effect. This can help explain some of the filtering results observed in Figure 3.

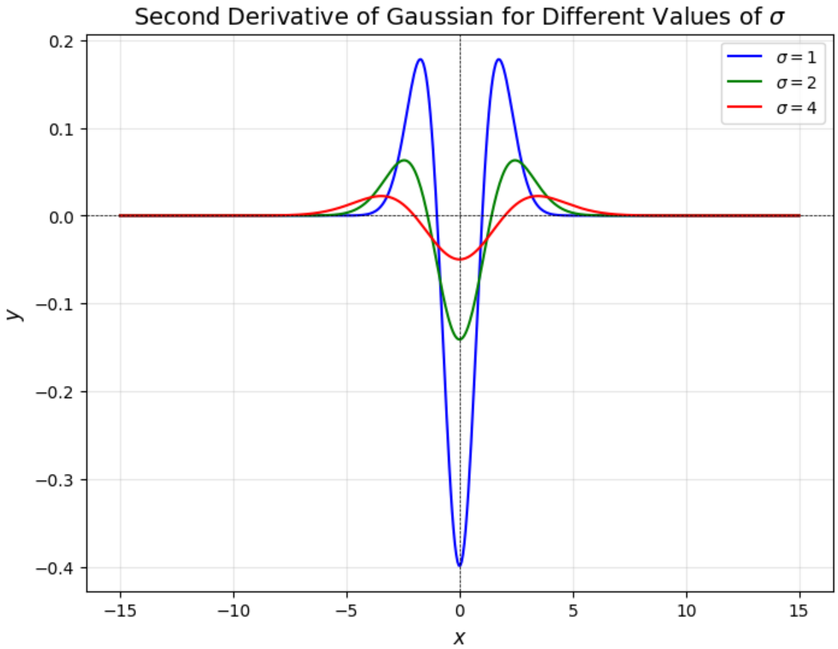

Figure 4: Visualization of the Laplacian concentration behavior as σ changes (taken from assignment 1).

4 Canny Edge Detection

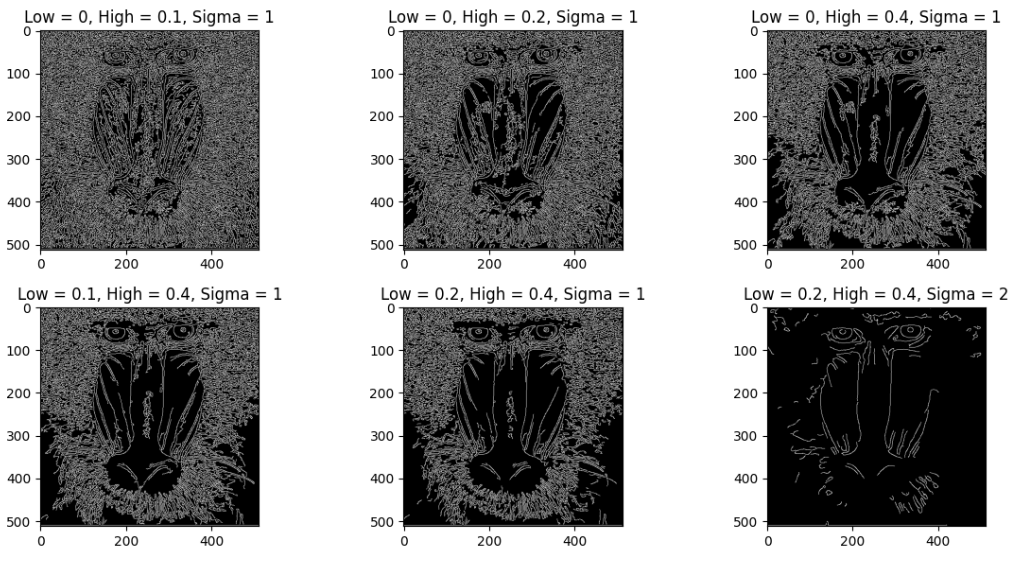

Figure 5: Canny Edge Detection with varying High/Low thresholds and Sigma values.

The Canny Edge Detection algorithm operates in the following way: It begins with applying a Gaussian filter to smooth the image and reduce noise, where the σ value determines the level of smoothing. Next, the algorithm computes the intensity gradients of the image. The provided "low" and "high" thresholds are used as double boundaries for the gradient magnitudes. Pixels with gradient magnitudes higher than the "high" threshold are identified as strong edges, while those below the "low" threshold are ignored. Pixels falling between these thresholds are classified as edges only if they are connected to strong edges.

The results shown in Figure 5 show the results of the Canny edge detection with different parameters:

- **Low = 0, High = 0.1,** σ **= 1**: As seen in the figure, the combination of a low threshold and minimal smoothing results in a highly detailed and jumbled edge map. While many fine details are preserved, the image is difficult to interpret.
- **Low = 0, High = 0.2,** σ **= 1**: Increasing the "High" threshold by 0.1 reduces the number of strong edges. The fine details in the fur remain visible, but the facial features start to become more distinct.
- **Low = 0, High = 0.4,** σ **= 1**: Raising the "High" threshold further significantly reduces noise. Prominent facial features are clearly visible, while finer details, such as those in the fur, are more repressed.
- **Low = 0.1, High = 0.4,** σ **= 1**: The nonzero "Low" threshold helps filter out weak edges, reducing background noise. This adjustment highlights the mandrill's key features, such as the nose, eyes, and major fur lines.
- **Low = 0.2, High = 0.4,** σ **= 1**: A higher "Low" threshold eliminates even more weak edges, leaving an edge map with minimal clutter. The remaining edges capture the mandrill's facial features with clarity, and most minor details are removed.
- **Low = 0.2, High = 0.4,** σ **= 2**: Keeping the same "Low" and "High" thresholds but increasing the σ value leads to more smoothing during the initial Gaussian filtering step. This results in a simplified edge map, focusing primarily on bold and essential facial features, with finer details effectively smoothed out.