[Pick the date]

Report for Week 1

Introduction to the basics

Patrick de Kok 5640318
Georgios Methenitis 10407537

Report for Week 1

Introduction to the basics

Step A

We have been able to run the `main_script_week1.py` file without any syntax errors. This required removing the line

```
cd “CHANGE THIS PATH TO YOUR WORKING DIRECTORY $main_dir/python/week1”
```

as this is not proper Python syntax. When the Python session is started in the same directory as where the script resides, the current working directory is already set correctly.

The data we are going to use is located in the `data/objects/flowers` folder, two folders up from the current working directory.

Step B

Now we know how to locate the images, we can load them and display them to the user. As an example, `../../data/objects/flowers/1.jpg` is displayed in Figure 1. This requires the usage of `matplotlib.pyplot.imshow(image)` function. The function reads a 2D or 3D numpy.array object and displays it as an image.

When we count the number of elements in the first layer of the array, we obtain the number of rows the image has. Each of the first layers' elements should have another array, representing a column position at that row. All these nested arrays should have the same length, as it would be strange to have an image with fringed edges. These nested arrays most often contain either a scalar value for single-channel images, such as gray-scale or depth map images, or a triple of scalars which is appropriate for RGB, HSV and other three-channel representations.

Inspection shows that the image of Figure 1 contains 1024 rows and 684 columns. As imshow automatically interprets three-channel images as RGB and the image does seem to represent something from nature, it most probably is in the RGB color space.

Figure 1: Shown is the first object of the `flowers` data set.

Step C

One common used image representation which is easy to compare, is the combined histogram of color channels. For each possible value per color channel, the frequency of that value is counted in the image. This value-frequency pair is stored. On the end, the frequencies are normalized, such that all frequencies sum up to 1. In this way, the histogram of an image with many pixels is very similar to that of the same image with a lower resolution. With these single channel histograms we can see easily the amount of light and dark pixels per color. Each such histogram contains 256 different values (bins). Because all color channels are important, we combine these to a single histogram of 768 bins, where the single channel histograms are concatenated.

The relative pixel value frequencies (so seeing each triple as a unique value in the image) does not generalize enough over the image. This would result in very low frequencies. If someone takes a picture from the same object from the same position, but only with a slight difference in colors (because of the moving Sun, clouds or something else), these values will be different and the histogram will be quite different.

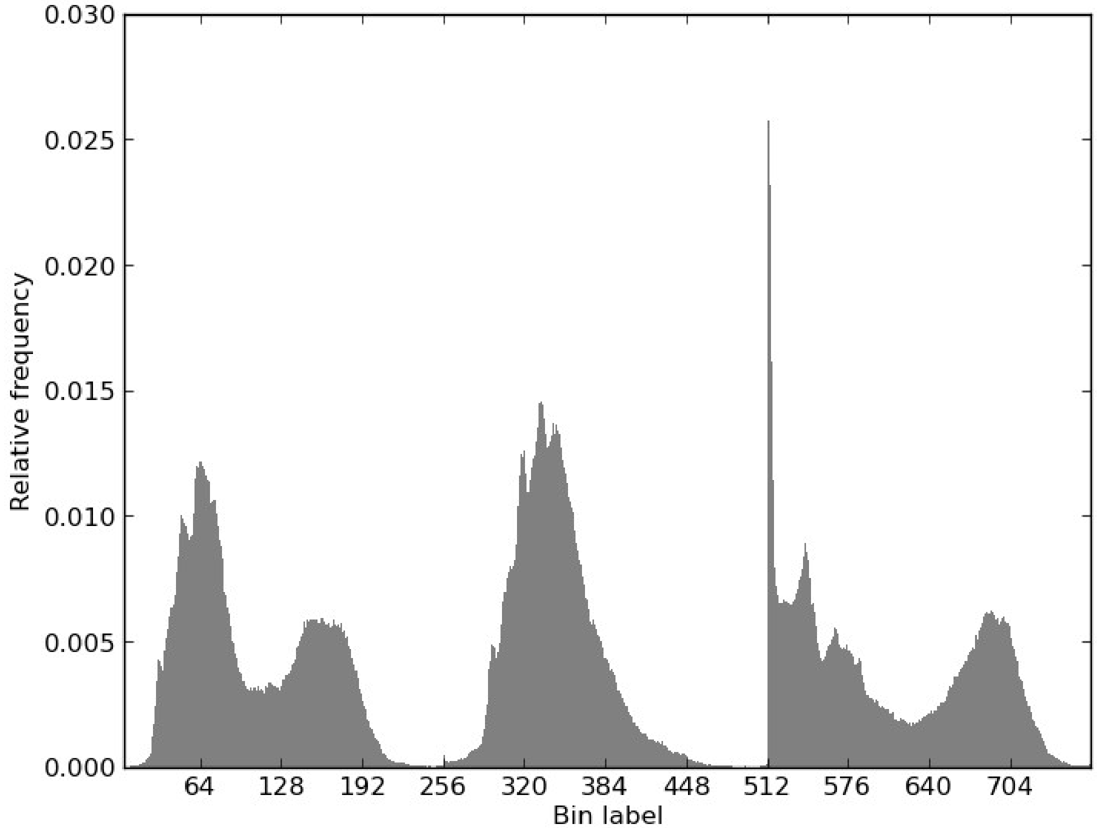

Figure 2: Relative color channel frequencies of Figure 1. Bin labels 0 to 256 correspond to the relative frequency of the red values 0 to 256 in the image, labels 256 to 512 correspond to the relative frequency of the green values 0 to 256, and labels 512 to 768 correspond to the relative frequency of blue values 0 to 256.

Step D

To compare the image representations, one can apply different distance measures. We have implemented five different distance measures. For two histograms x, y with $0 \leq i < n$ bins, we have the following distance measures:

- Euclidean distance: $d_E(x, y) = \sum \left(\frac{x_i}{x_\surd} - \frac{y_i}{y_\surd} \right)^2$ where $h_\surd = \sqrt{\sum h_i^2}$
- ℓ_2 distance: $d_{\ell_2}(x, y) = \sum \frac{x_i}{x_\surd} \frac{y_i}{y_\surd}$ where $h_\surd = \sqrt{\sum h_i^2}$

- χ^2 distance: $d_{\chi^2}(x,y)=\frac{\sum(x_i-y_i)^2}{x_i+y_i}$
- Histogram intersection distance: $d_h(x,y)=\sum min\{x_i,y_i\}$
- Hellinger distance: $d_H(x,y)=\sum\sqrt{x_i}\sqrt{y_i}$

Note that for d_E, d_{χ^2} lower outcomes indicate that x, y are more similar ("more nearby" in the space of histograms with the given distance measure as inner product), but for d_{ℓ_2}, d_h, d_H more similar objects have higher outcomes. Actually, d_{ℓ_2}, d_h, d_H are not distance measures (one rule to be a true distance measure d is that the distance of a vector to itself is $d(x,x)=0$). d_{ℓ_2}, d_h, d_H are *similarity measures*. Any distance measures can easily be transformed to a similarity measure:

$$s(x)=-d(x)$$

In our code, we have transformed each distance measure in a similarity measure.

Step E

Ranking the images for similarity to a chosen image is now very easy. First, we have to compute the combined histogram of each image in the data set. Then, compute the similarity of each image with the chosen image. Sort all images in descending order with respect to their similarity with the chosen image. The first image on your list is most similar to the provided query image.

This has been done for each similarity measure with query images `5.jpg`, `10.jpg` and `15.jpg`. The results summarized in Table 1. Corresponding images are included in Appendix A.

Query id	**Sim. meas.**	**Rank 1**	**Rank 2**	**Rank 3**	**Rank 4**	**Rank 5**
5	d_E	5	53	35	37	54
5	d_{ℓ_2}	5	53	35	37	54
5	d_{χ^2}	5	37	54	57	56
5	d_h	5	56	37	54	51
5	d_H	5	37	56	54	27

Query id	**Sim. meas.**	**Rank 1**	**Rank 2**	**Rank 3**	**Rank 4**	**Rank 5**
10	d_E	10	60	17	2	54
10	d_{ℓ_2}	10	60	17	2	54
10	d_{χ^2}	10	60	17	2	35
10	d_h	10	60	17	2	43
10	d_H	10	60	17	43	30
15	d_E	15	27	12	18	31
15	d_{ℓ_2}	15	27	12	18	31
15	d_{χ^2}	15	27	18	31	19
15	d_h	15	27	18	20	31
15	d_H	15	27	12	54	18

Table 1: Overview of image ids similar to the query under the provided similarity measure.

Step F

In this step, an introduction to blurring and the discrete derivative of images is given. We are going to blur a gray-scale version of Figure B.1, which can be seen in Figure 3. To achieve this, a convolution of a Gaussian kernel G_σ is applied to the image with different spreads σ. To demonstrate the effect of the spread of the kernel on the "blurriness" of the image, and to demonstrate a comparison of our implementation of the convolution with the already provided version, consider Figures B.2, B.3 and B.4.

The magnitude of the gradient corresponds with the concept of how "present" an edge is visually. Higher values indicate a more prominent edge. This can be used to create a binary edge detector, which sets each pixel to `True` (or 1) if its magnitude is larger than a certain threshold θ and to `False` (or 0) otherwise.

The Canny edge detector builds forth on this basic idea, also retrieving the direction of the edge. Explaining this algorithm is outside the scope of the assignment.

Bonus

The first-order derivative of the Gaussian is easy to compute:

$$G_\sigma(x)' = \left(\frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\frac{-x^2}{2\sigma^2}\right)' \qquad \text{By definition.}$$

$$= \frac{1}{\sigma\sqrt{2\pi}} \cdot \left(\exp\frac{-x^2}{2\sigma^2}\right)'$$

Constants are unaffected by derivation.

$$= \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp'\frac{-x^2}{2\sigma^2} \cdot \left(\frac{-x^2}{2\sigma^2}\right)'$$

Chain rule.

$$= \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\frac{-x^2}{2\sigma^2} \cdot -\frac{2x}{2\sigma^2}$$

$$= \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\frac{-x^2}{2\sigma^2} \cdot -\frac{x}{\sigma^2}$$

$$= G_\sigma(x) \cdot -\frac{x}{\sigma^2}$$

$$= -\frac{x}{\sigma^2} G_\sigma(x)$$

The second order derivative is also easy to compute:

$$G_\sigma(x)'' = \left(\frac{-x}{\sigma^2}\right)' G_\sigma(x) + \frac{-x}{\sigma^2} G_\sigma(x)'$$

Product rule.

$$= \left(\frac{-1}{\sigma^2}\right)' G_\sigma(x) + \left(\frac{-x}{\sigma^2}\right)^2 G_\sigma(x)$$

$$= \frac{G_\sigma(x)}{-\sigma^2} + \frac{x^2}{\sigma^4} G_\sigma(x)$$

The second order derivatives of the gray-scale version of Figure 1 with varying values for σ are visualized in Appendix C.

Appendix A: Similar images

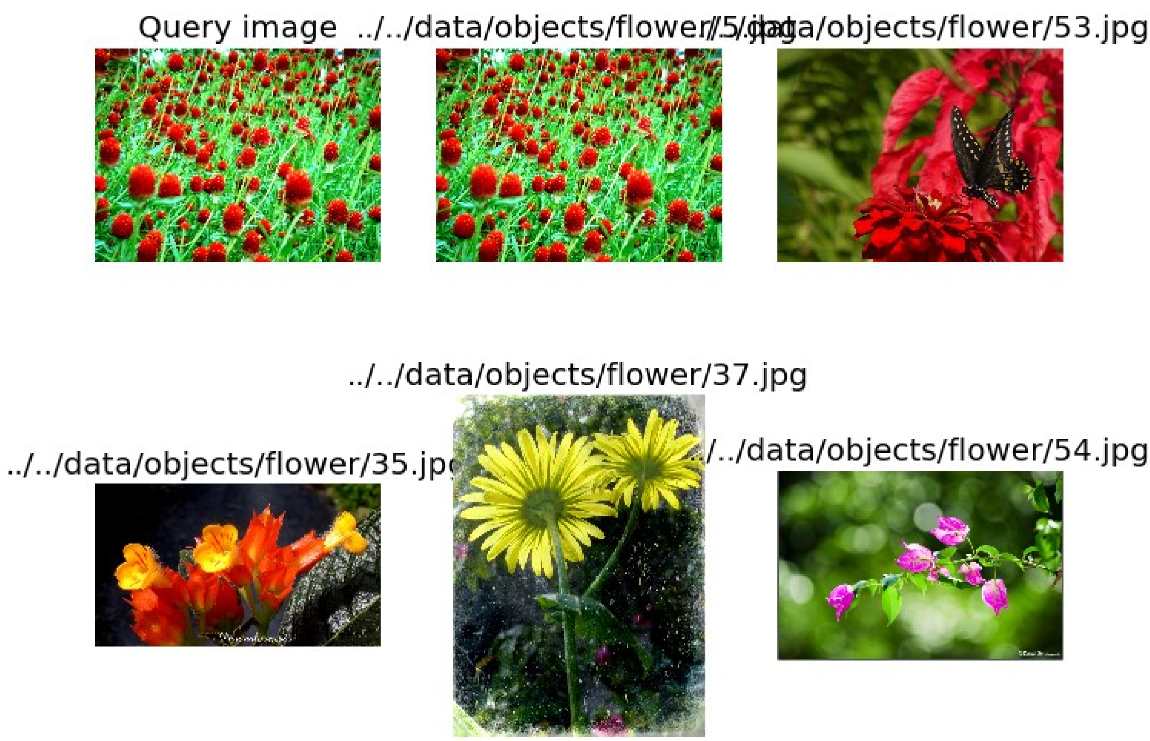

Figure A.1: Similar images to 5 under the Euclidean and l2 measure.

Figure A.2: Similar images to 5 under chi2 measure.

Figure A.3: Similar images to 5 under histogram measure.

Figure A.4: Similar images to 5 under Hellinger measure.8899i

Figure A.5: Similar images to 10 under Euclidean and l2 measure.

Figure A.6: Similar images to 10 under chi2 measure.

Figure A.7: Similar images to 10 under histogram measure.

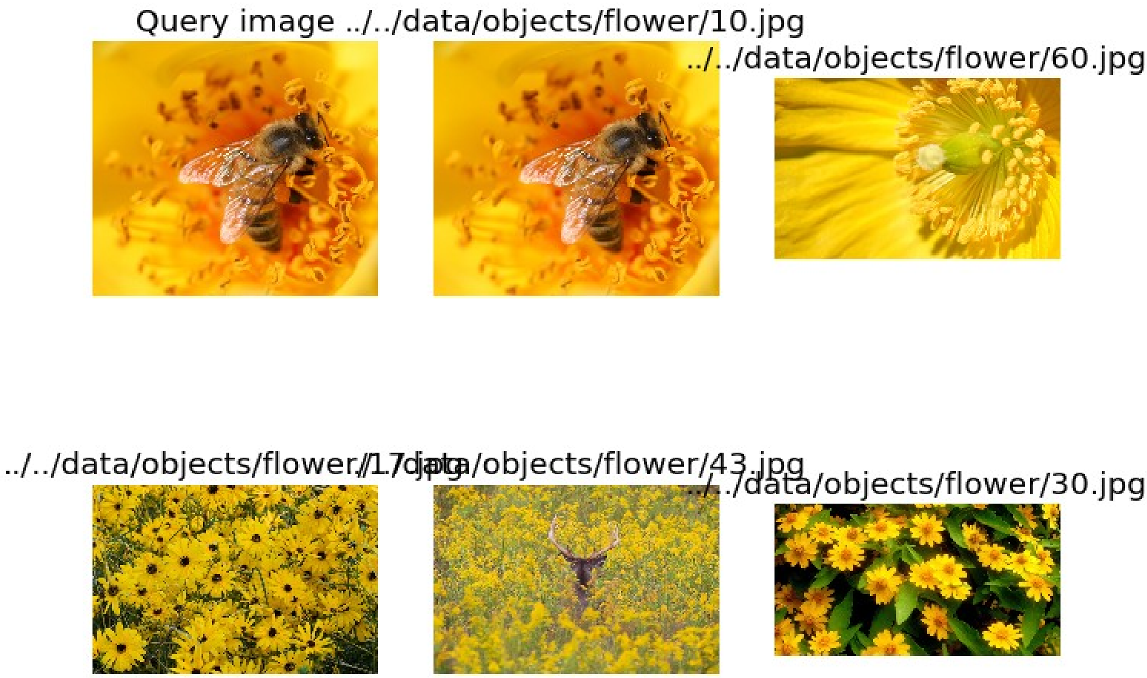

Figure A.8: Similar images to 10 under Hellinger measure.

Figure A.9: Similar images to 15 under Euclidean and l2 measure.

Figure A.10: Similar images to 15 under chi2 measure.

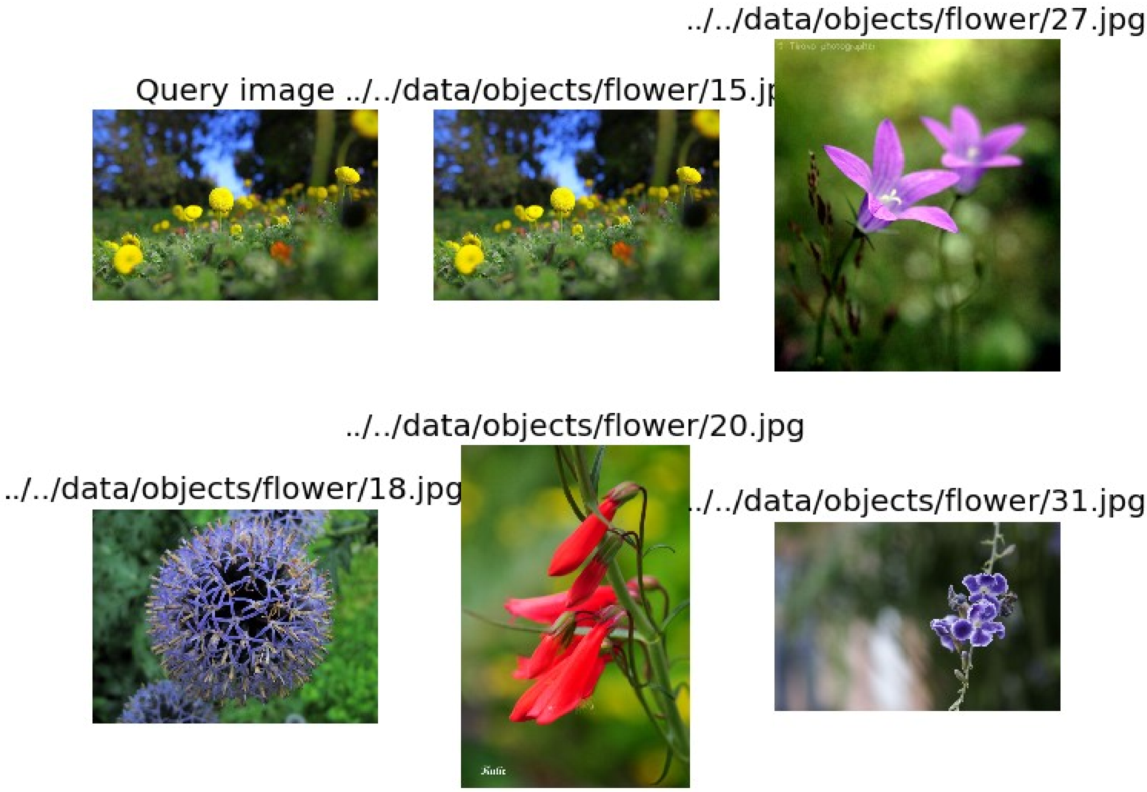

Figure A.11: Similar images to 15 under histogram measure.

Figure A.12: Similar images to 15 under Hellinger measure.

Appendix B: Blurry images and derivatives

Figure B.1: a gray-scale version of Figure 1.

Figure B.2: The gray-scale image convoluted with a Gaussian kernel with scaling parameter $\sigma=1$.
The blurring effect is barely noticeable. *Left:* provided implementation. *Right:* own implementation.

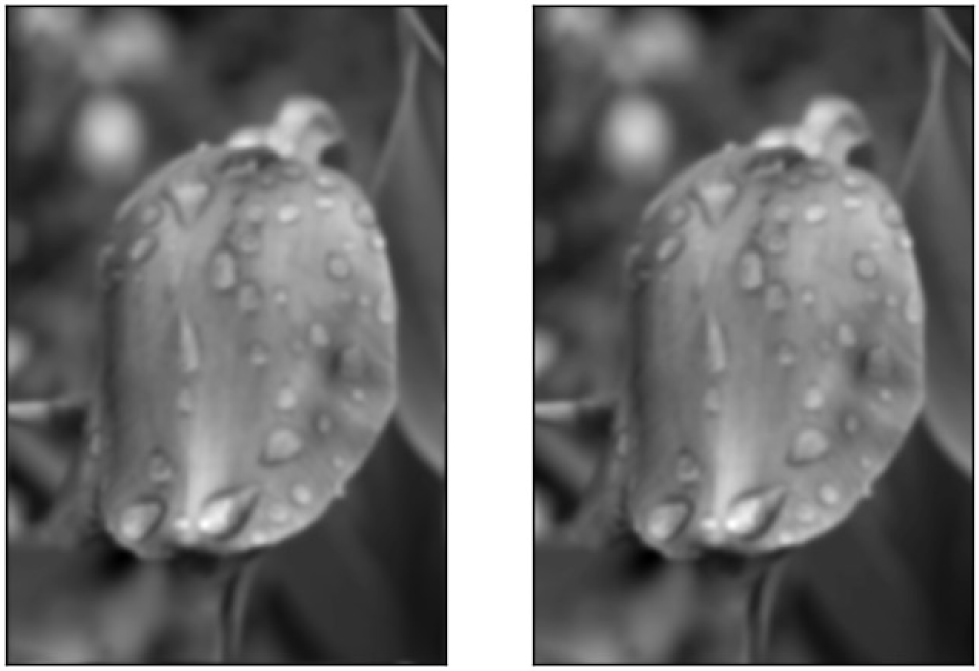

Figure B.3: The gray-scale image convoluted with a Gaussian kernel with scaling parameter $\sigma = 5$.

The blurring effect is now clearly present. Sharp edges and thin lines are not visible anymore. *Left:* provided implementation. *Right:* own implementation.

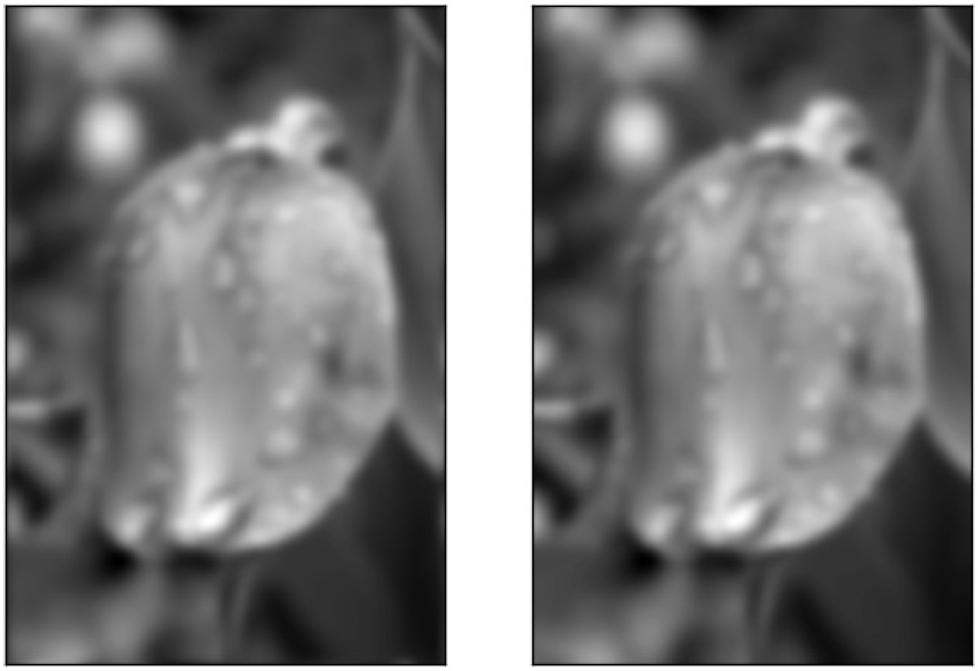

Figure B.4: The gray-scale image convoluted with a Gaussian kernel with scaling parameter $\sigma = 10$.

The blurring effect is now so string that it becomes difficult to discern the small water drops on the flower. *Left:* provided implementation. *Right:* own implementation.

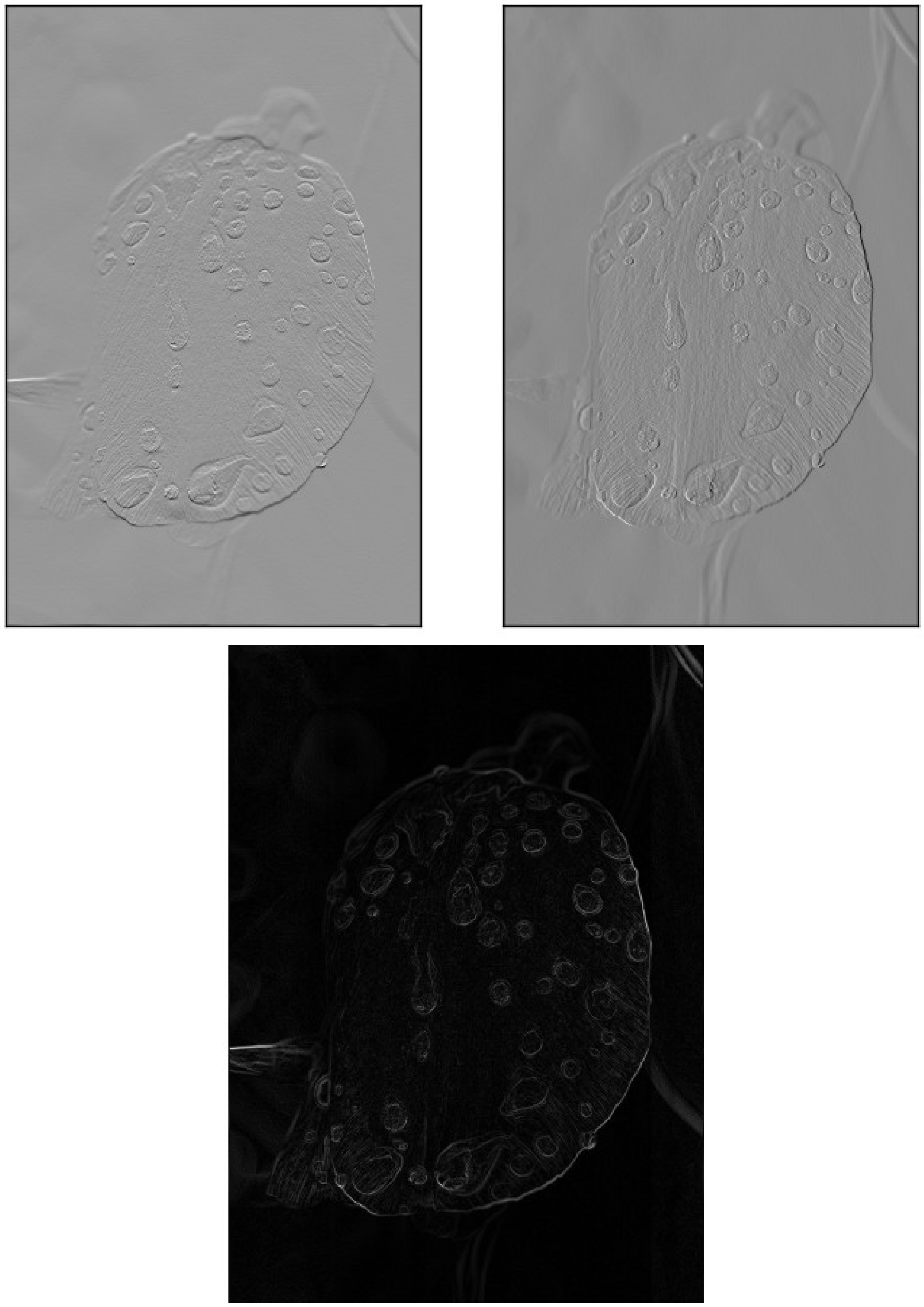

Figure B.5: The first order derivative of the gray-scale image for $\sigma = 1$ and its magnitude. Only really sharp edges of the original image show up. *Left:* the derivative with respect to *y*. *Right:* the derivative with respect to *x*. *Bottom:* magnitude of the gradient.

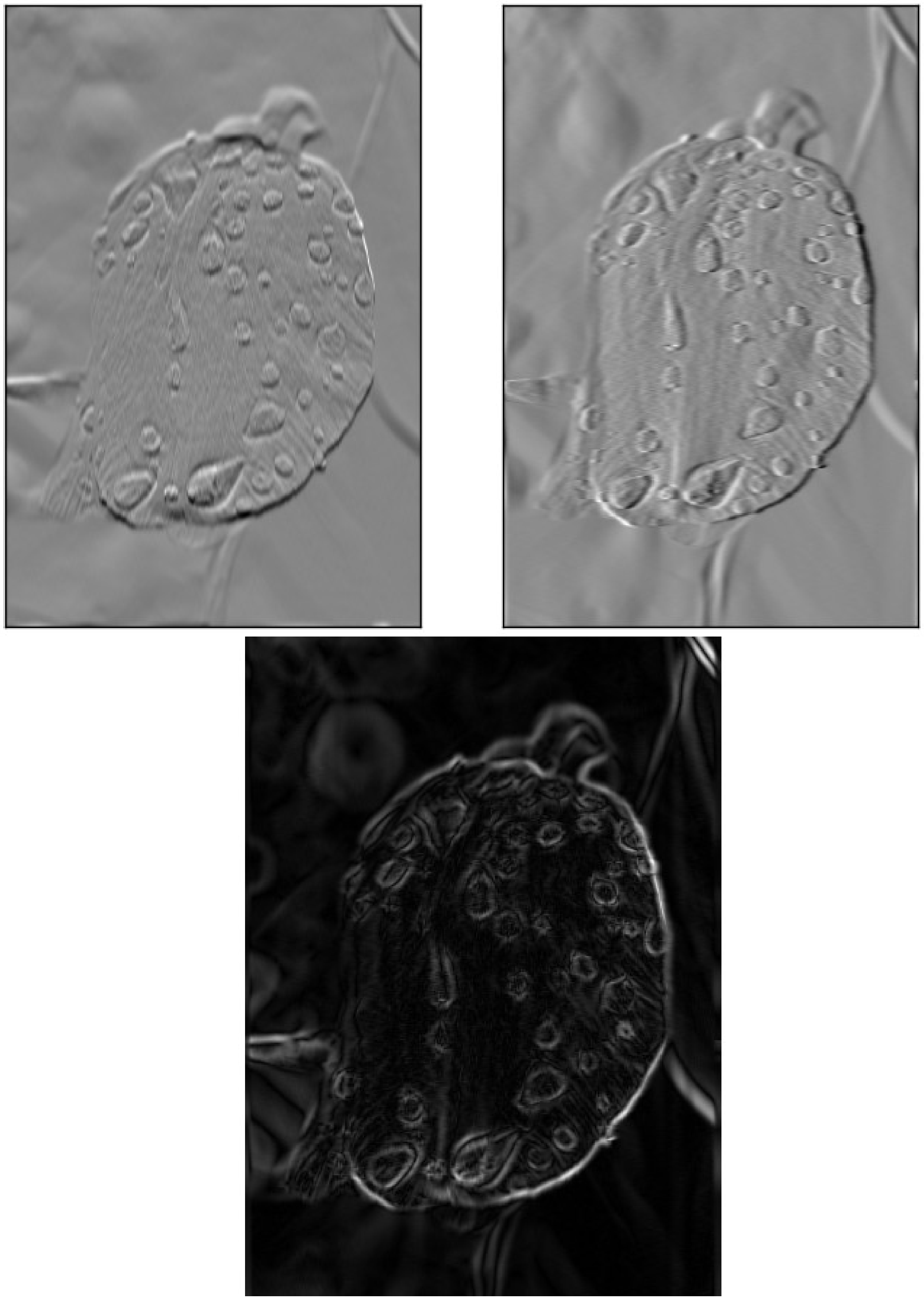

Figure B.6: The first order derivative of the gray-scale image for $\sigma = 5$ and its magnitude. Edges of bigger structures have become visible. *Left:* the derivative with respect to y. *Right:* the derivative with respect to x. *Bottom:* magnitude of the gradient.

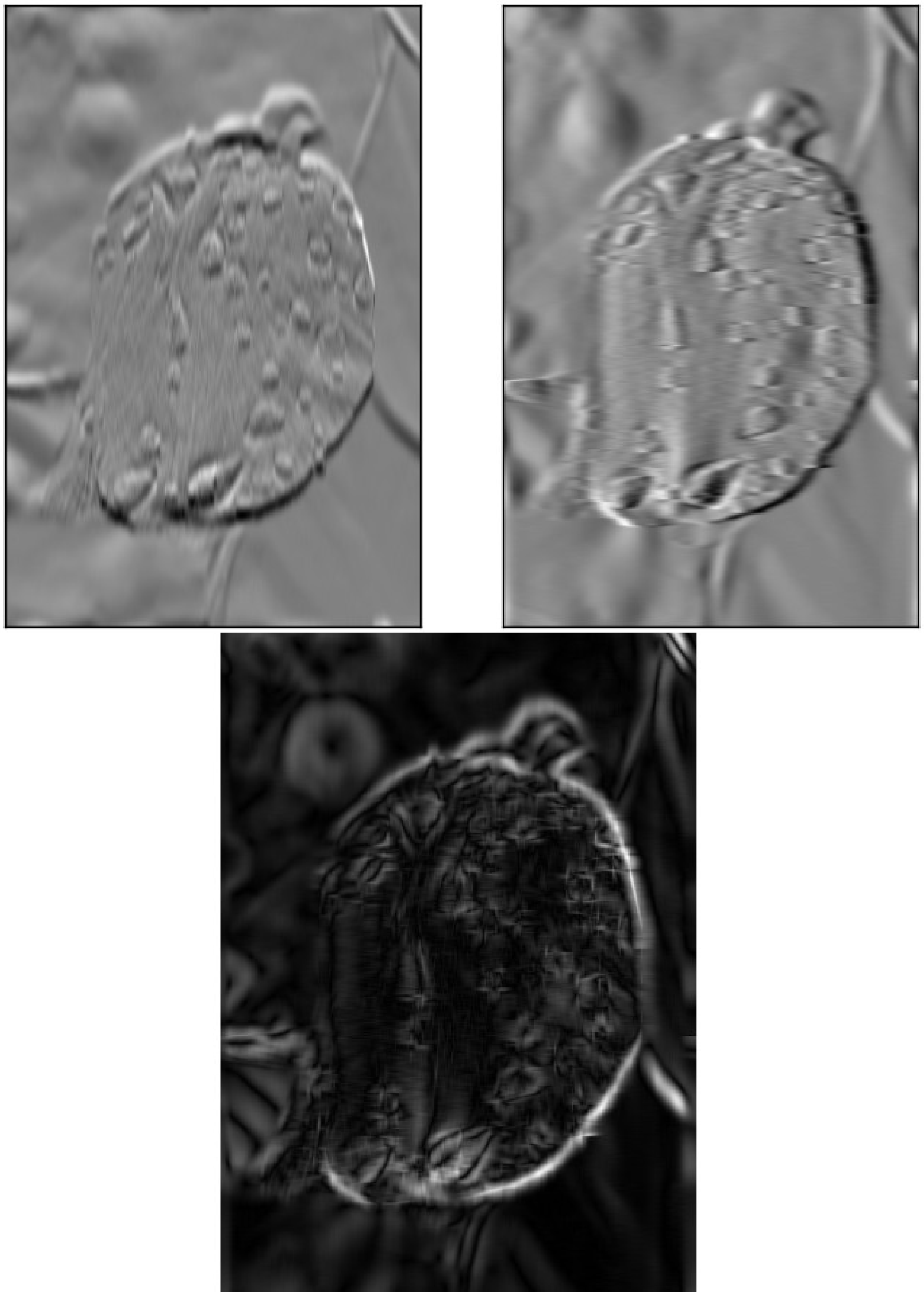

Figure B.7: The first order derivative of the gray-scale image for $\sigma = 10$ and its magnitude. Now, edges show up where the original image transcended slowly, such as on the left top of the flower.. *Left:* the derivative with respect to *y*. *Right:* the derivative with respect to *x*. *Bottom:* magnitude of the gradient.

Appendix C: Second order derivatives

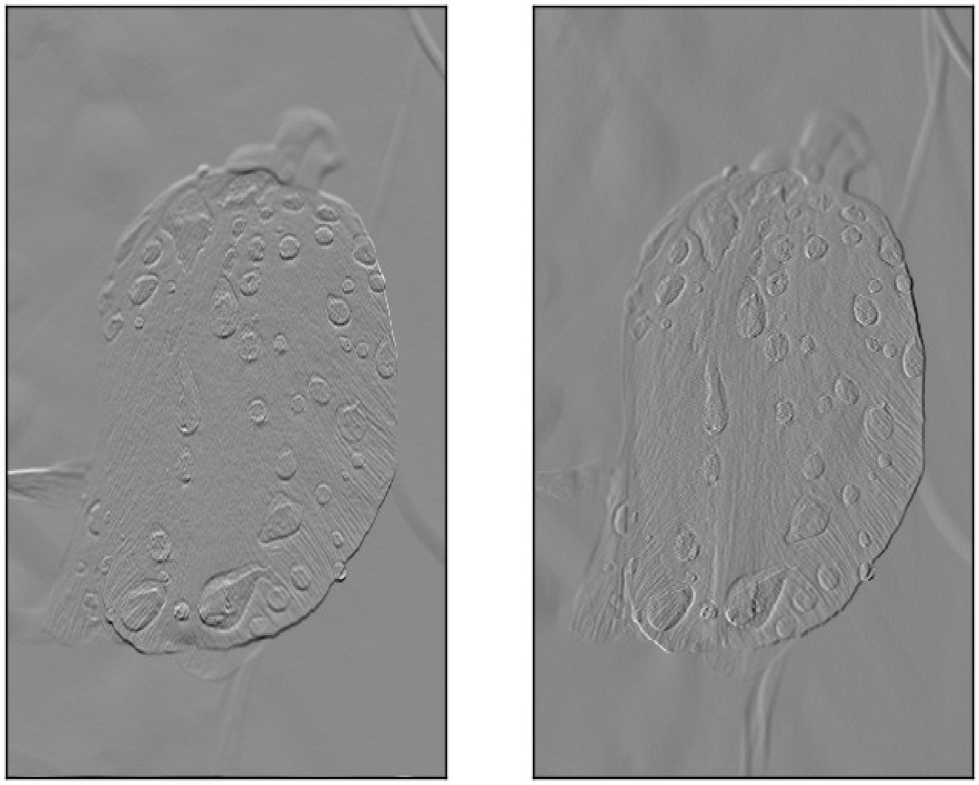

Figure C.1: The second order derivative for $\sigma = 1$. *Left:* the derivative with respect to *y*. *Right:* the derivative with respect to *x*.

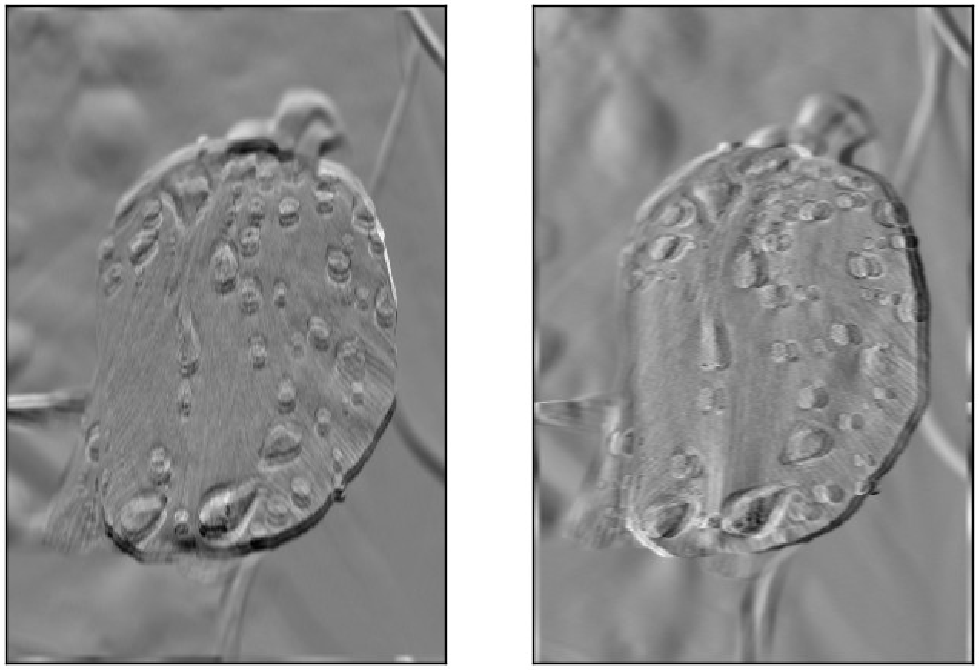

Figure C.2: The second order derivative for $\sigma = 5$. *Left:* the derivative with respect to *y*. *Right:* the derivative with respect to *x*.

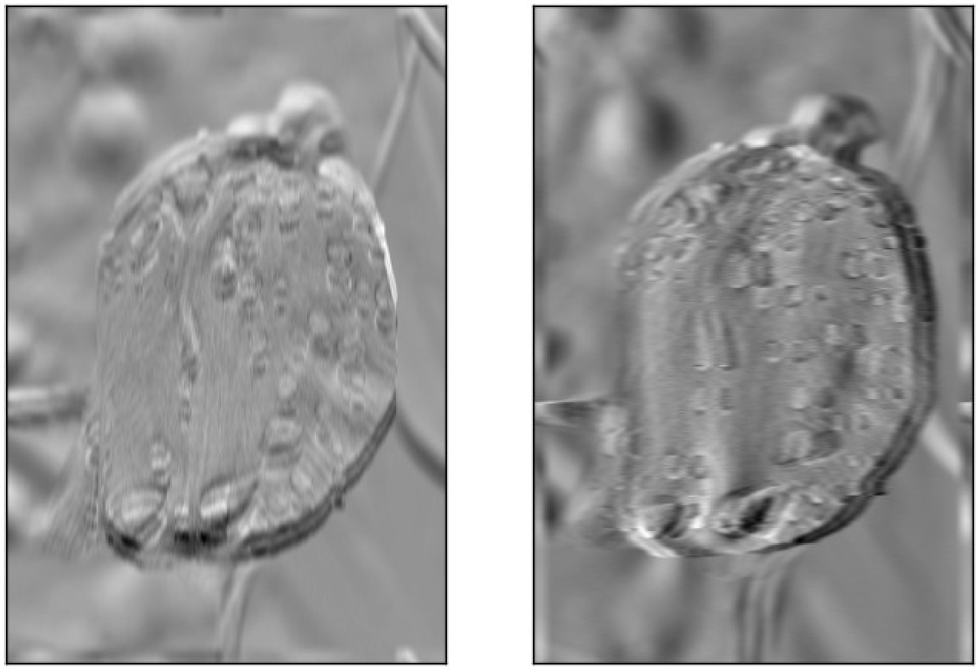

Figure C.3: The second order derivative for $\sigma = 10$. *Left:* the derivative with respect to *y*. *Right:* the derivative with respect to *x*.