CSC420 A1

Jenney Ren

May 2020

1 Part I

1. Replacing the input signal $I(x)$ by its representation in terms of impulses, given $I(x)$ is discrete:

$$I(x) = \sum_{u=-\infty}^{\infty} I(u)\delta(x-u)$$

Let T be a linear time invariant system representing operation $h(x) * I(x)$.

$$\begin{aligned} T[I(x)] &= T\left[\sum_{u=-\infty}^{\infty} I(u)\delta(x-u)\right] \\ &= \sum_{u=-\infty}^{\infty} I(u)T[\delta(x-u)] && \text{By linearity since } I(u) \text{ is a constant term} \\ &= \sum_{u=-\infty}^{\infty} I(u)h(x-u) && \text{By definition of } h \end{aligned}$$

2. For some polar coordinates (x, y) in image I, multiplying by the rotation matrix gives:

$$\begin{pmatrix} x\cos\theta - y\sin\theta \\ y\cos\theta + x\sin\theta \end{pmatrix}$$

Let $r = x\cos\theta - y\sin\theta, r' = y\cos\theta + x\sin\theta$. Let $f = I(x, y)$.

$$
\begin{aligned}
\frac{\partial^2 f}{\partial x^2} &= \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) \\
&= \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial r}\frac{\partial r}{\partial x} + \frac{\partial f}{\partial r'}\frac{\partial r'}{\partial x}\right) \\
&= \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial r}\cos\theta + \frac{\partial f}{\partial r'}\sin\theta\right) \\
&= \frac{\partial}{\partial r}\frac{\partial f}{\partial x}\cos\theta + \frac{\partial}{\partial r'}\frac{\partial f}{\partial x}\sin\theta \\
&= \frac{\partial}{\partial r}\left(\frac{\partial f}{\partial r}\cos\theta + \frac{\partial f}{\partial r'}\sin\theta\right)\cos\theta + \frac{\partial}{\partial r'}\left(\frac{\partial f}{\partial r}\cos\theta + \frac{\partial f}{\partial r'}\sin\theta\right)\sin\theta \\
&= \frac{\partial^2 f}{\partial r^2}\cos^2\theta + 2\frac{\partial^2 f}{\partial r \partial r'}\sin\theta\cos\theta + \frac{\partial^2 f}{\partial r'^2}\sin^2\theta
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 f}{\partial y^2} &= \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial y}\right) \\
&= \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial r}\frac{\partial r}{\partial y} + \frac{\partial f}{\partial r'}\frac{\partial r'}{\partial y}\right) \\
&= \frac{\partial}{\partial y}\left(-\frac{\partial f}{\partial r}\sin\theta + \frac{\partial f}{\partial r'}\cos\theta\right) \\
&= -\frac{\partial}{\partial r}\frac{\partial f}{\partial y}\sin\theta + \frac{\partial}{\partial r'}\frac{\partial f}{\partial y}\cos\theta \\
&= -\frac{\partial}{\partial r}\left(-\frac{\partial f}{\partial r}\sin\theta + \frac{\partial f}{\partial r'}\cos\theta\right)\sin\theta + \frac{\partial}{\partial r'}\left(-\frac{\partial f}{\partial r}\sin\theta + \frac{\partial f}{\partial r'}\cos\theta\right)\cos\theta \\
&= \frac{\partial^2 f}{\partial r^2}\sin^2\theta - 2\frac{\partial^2 f}{\partial r \partial r'}\sin\theta\cos\theta + \frac{\partial^2 f}{\partial r'^2}\cos^2\theta
\end{aligned}
$$

Now, cancelling the terms with both sin and cos:

$$
\begin{aligned}
\Delta I &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\
&= \frac{\partial^2 f}{\partial r^2}\cos^2\theta + \frac{\partial^2 f}{\partial r'^2}\sin^2\theta + \frac{\partial^2 f}{\partial r^2}\sin^2\theta + \frac{\partial^2 f}{\partial r'^2}\cos^2\theta \\
&= \frac{\partial^2 f}{\partial r^2}\left(\sin^2\theta + \cos^2\theta\right) + \frac{\partial^2 f}{\partial r'^2}\left(\sin^2\theta + \cos^2\theta\right) \\
&= \frac{\partial^2 f}{\partial r^2} + \frac{\partial^2 f}{\partial r'^2} \\
&= I_{rr} + I_{r'r'}
\end{aligned}
$$

3. The condition is linear variation. The ideal case would be a sharp, straight edge in the part of the picture we are analyzing. If the picture differs from the ideal case and has curves or corners, the algorithm may detect false edges.

   Mathematically, I believe that the Laplace being rotation invariant means that we can show that the algorithm works for one direction and generalize this to all directions. A sharp, straight edge along the x-axis would be represented by the equation:

$$I(x,y) = \begin{cases} ay+b, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

   To make the math easier, I assumed the image was continuous and x, y were scaled such that $\sigma = 1$, meaning $\int G(x) = 1$. Using an integral calculator, I found that $\int xG(x) = 0$.

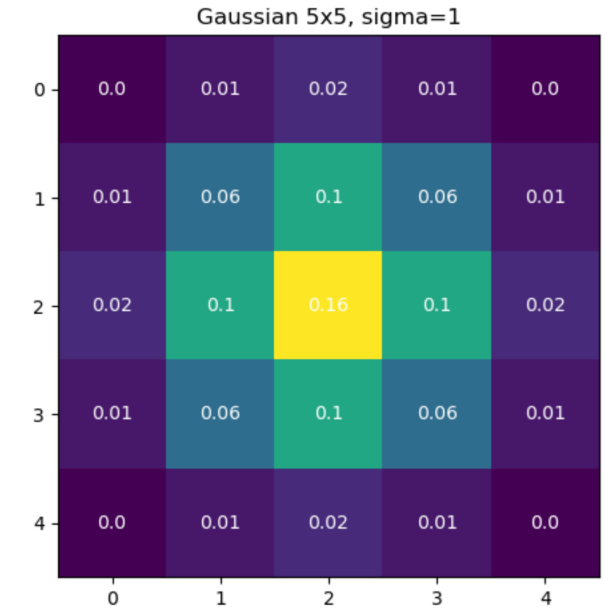
   If we apply the algorithm, we would take the Laplacian of the Gaussian convolved with the image:

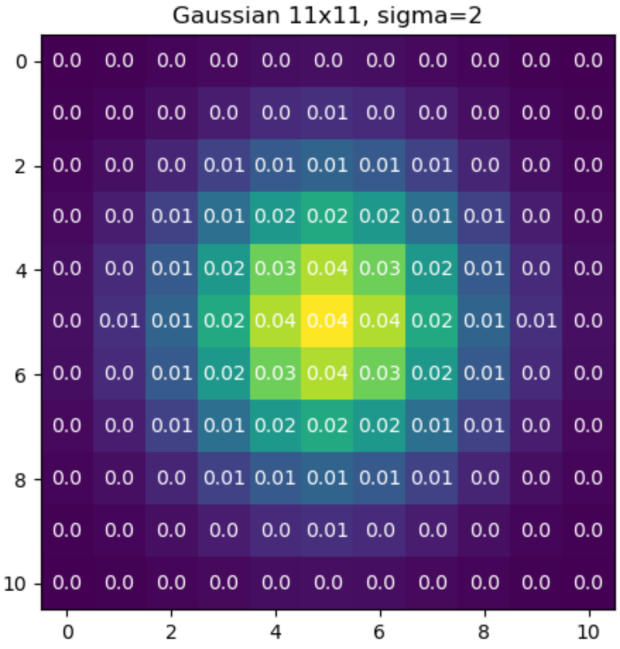
$$\begin{aligned}
\Delta G * I &= \nabla^2 \int_{u=-\infty}^{\infty} \int_{v=-\infty}^{\infty} G(u,v) \cdot I(x-u, y-v) du dv \\
&= \nabla^2 \int_{u=-\infty}^{\infty} \int_{v=-\infty}^{\infty} G(u)G(v) \cdot I(x-u, y-v) du dv \qquad \text{Since } G \text{ is separable} \\
&= \nabla^2 \int_{u=-\infty}^{x} G(u) du \int_{v=-\infty}^{\infty} G(v) \cdot (a(y-v)+b) dv \qquad \text{Since for } I \neq 0, x-u > 0 \\
&= \nabla^2 \int_{u=-\infty}^{x} G(u) du \left(ay \int_{v=-\infty}^{\infty} G(v) dv - a \int_{v=-\infty}^{\infty} vG(v) dv + b \int_{v=-\infty}^{\infty} G(v) dv \right) \\
&= \nabla^2 \int_{u=-\infty}^{x} G(u) du (ay(1) - a(0) + b(1)) \\
&= \nabla G(x)(ay+b) \\
&= \nabla \left(\frac{1}{\sqrt{2\pi}} e^{-x^2} \right) (ay+b) \qquad \text{From slides} \\
&= -2x \left(\frac{1}{\sqrt{2\pi}} e^{-x^2} \right) (ay+b)
\end{aligned}$$

   The algorithm then looks for where the Laplacian of Gaussian of the image is 0, which occurs when $x = 0$. The original edge was horizontal along the x-axis, so $x = 0$ is perpendicular to the edge.

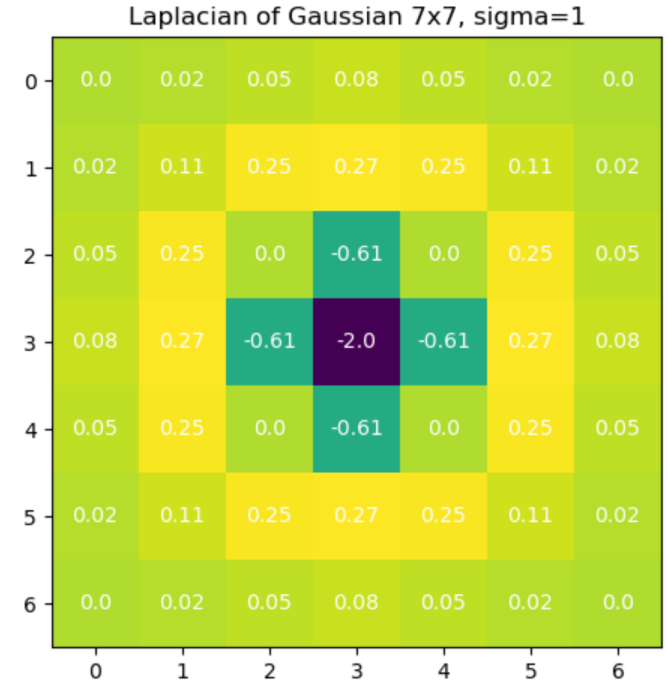
2 Part II

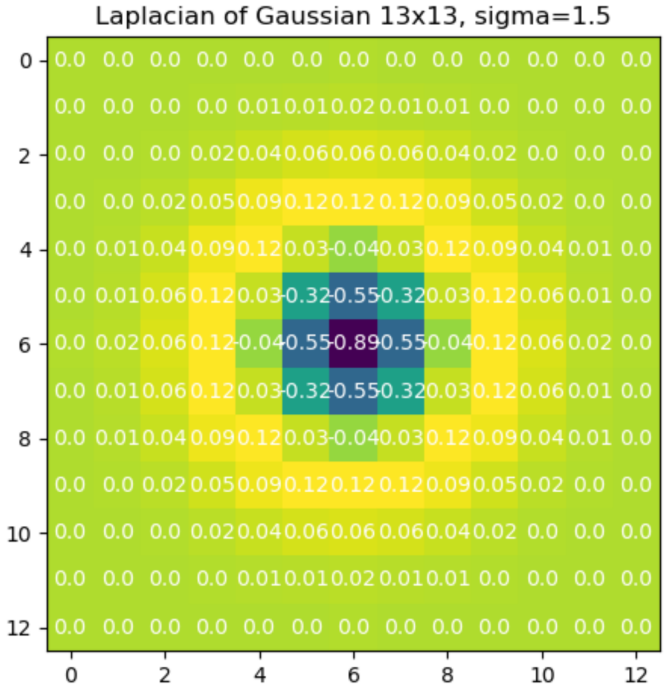
4. Gaussian filters:





5. Laplacian of Gaussian filters:


Laplacian of Gaussian 7x7, sigma=1
0 0.0 0.02 0.05 0.08 0.05 0.02 0.0
1 0.02 0.11 0.25 0.27 0.25 0.11 0.02
2 0.05 0.25 0.0 -0.61 0.0 0.25 0.05
3 0.08 0.27 -0.61 -2.0 -0.61 0.27 0.08
4 0.05 0.25 0.0 -0.61 0.0 0.25 0.05
5 0.02 0.11 0.25 0.27 0.25 0.11 0.02
6 0.0 0.02 0.05 0.08 0.05 0.02 0.0
0 1 2 3 4 5 6


Laplacian of Gaussian 13x13, sigma=1.5
0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.01 0.01 0.02 0.01 0.01 0.0 0.0 0.0 0.0
2 0.0 0.0 0.0 0.02 0.04 0.06 0.06 0.06 0.04 0.02 0.0 0.0 0.0
0.0 0.0 0.02 0.05 0.09 0.12 0.12 0.12 0.09 0.05 0.02 0.0 0.0
4 0.0 0.01 0.04 0.09 0.12 0.03 -0.04 0.03 0.12 0.09 0.04 0.01 0.0
0.0 0.01 0.06 0.12 0.03 -0.32 -0.55 -0.32 0.03 0.12 0.06 0.01 0.0
6 0.0 0.02 0.06 0.12 -0.04 -0.55 -0.89 -0.55 -0.04 0.12 0.06 0.02 0.0
0.0 0.01 0.06 0.12 0.03 -0.32 -0.55 -0.32 0.03 0.12 0.06 0.01 0.0
8 0.0 0.01 0.04 0.09 0.12 0.03 -0.04 0.03 0.12 0.09 0.04 0.01 0.0
0.0 0.0 0.02 0.05 0.09 0.12 0.12 0.12 0.09 0.05 0.02 0.0 0.0
10 0.0 0.0 0.0 0.02 0.04 0.06 0.06 0.06 0.04 0.02 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.01 0.01 0.02 0.01 0.01 0.0 0.0 0.0 0.0
12 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0 2 4 6 8 10 12

3 Part III

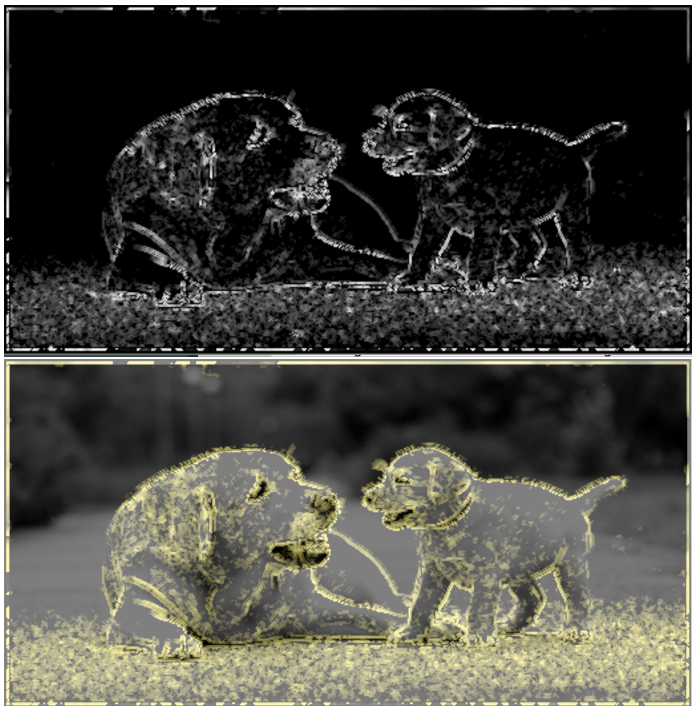
6. See Q6 in Python file

7. Paolina with $\sigma = 1$:

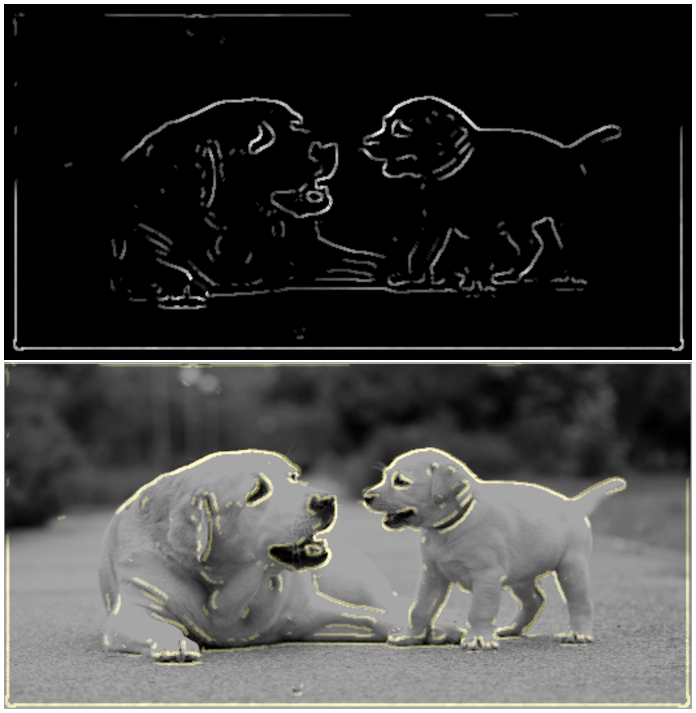
Paolina with $\sigma = 3$:

The other picture I chose was a picture of two dogs I found on Google images. This is the original picture in black and white:

Dog with $\sigma = 1$:

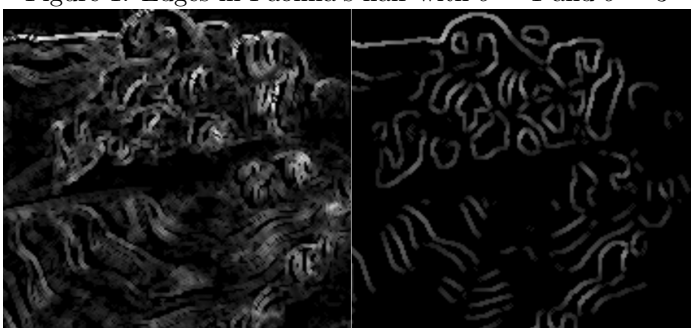


Dog with $\sigma = 3$:



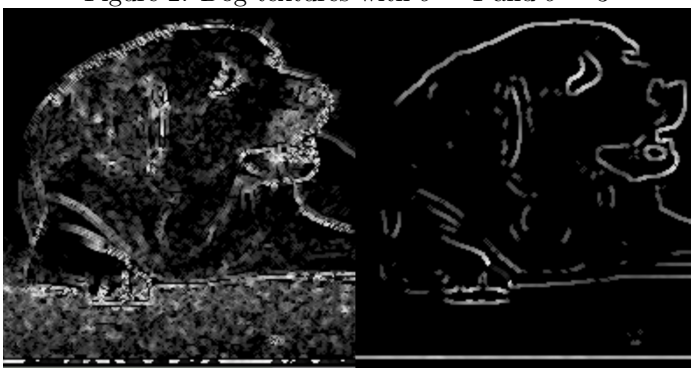
8. Comparing the $\sigma = 1$ to the $\sigma = 3$ filter, the $\sigma = 1$ has fewer edges overall. This is because more Gaussian blur is applied, causing the changes in intensity to be smaller since the Gaussian blur is taking a weighted average of the intensity of the pixel with the intensities of the pixels around it. This effect can easily be seen in Paolina's hair, where the $\sigma = 1$ filter results in many fine edges from the texture of her hair while the $\sigma = 3$ only shows the larger curls (as shown in Figure 1).

Figure 1: Edges in Paolina's hair with $\sigma = 1$ and $\sigma = 3$



With $\sigma = 1$, the dog picture has false edges since it has more texture (as shown in Figure 2). On the dog's fur or on the pavement, there might be large intensity changes that are due to the texture rather than an edge. A higher σ gets rid of the false edges but also greatly reduces the detail on the dog and the ears and paws are difficult to identify. This problem could be solved with hysteresis thresholding.

Figure 2: Dog textures with $\sigma = 1$ and $\sigma = 3$



The algorithm appears to fail when there is an edge but the background is about the same colour as the edge due to shadows. For example, Paolina's jawbone is not detected with $\sigma = 1$ or $\sigma = 3$, and neither is her forehead (as shown in Figure 3). Since the algorithm finds the areas with the

highest rate of change in intensity, if the background is close to the same colour then the rate of change in intensity is not very large. Without thresholding, this edge is faintly visible with $\sigma = 1$, but this also results in more false edges, especially for the dog picture (as shown in Figure 4).

Figure 3: Missing edges in Paolina's jawbone and forehead

False edges are detected at the edges of the picture, for example in the lower lefthand corner of the Paolina image (as shown in Figure 5). This is because with zero-padding, there is a black border around the image, which represents a rapid change in intensity if the image itself was bright near the edge. These edges can be removed by cropping the image.

As shown in Figure 6, the edges in the background of the dog picture are not detected, even for $\sigma = 1$. For example, there is no edge detected where the grass/trees meet the pavement, even though on the left side of the image the intensity change between the pavement and the greenery is fairly large. The reason this does not register as an edge is similar to why the shadows on Paolina don't register as edges. Even though there is a significant change in intensity, the intensity is too slow. As stated in question 3, the algorithm works best on a sharp, straight edge. Even though the edge is fairly straight, it is not sharp enough, so it violates the assumption of linear variation.

Theoretically, the algorithm does not work well if there are curves or

Figure 4: Dog picture with $\sigma = 3$ and no thresholding shows many false edges

corners in the local area where the filter is being applied. This may be what causes the dog's feet to look strange for $\sigma = 1$ (as shown in Figure 7), since the dogs' toes look like corners. The dogs' toes seem to be a rapid change in intensity in the original image, but the edge detection does not seem to indicate clear lines where the toes are. However, this may also be caused by noise in the image, since the other edges like the outline of the dogs' bodies also seem fuzzy.

Figure 5: False edge in corner of Paolina image due to zero padding

Figure 6: Edge not detected in dog picture despite significant intensity change

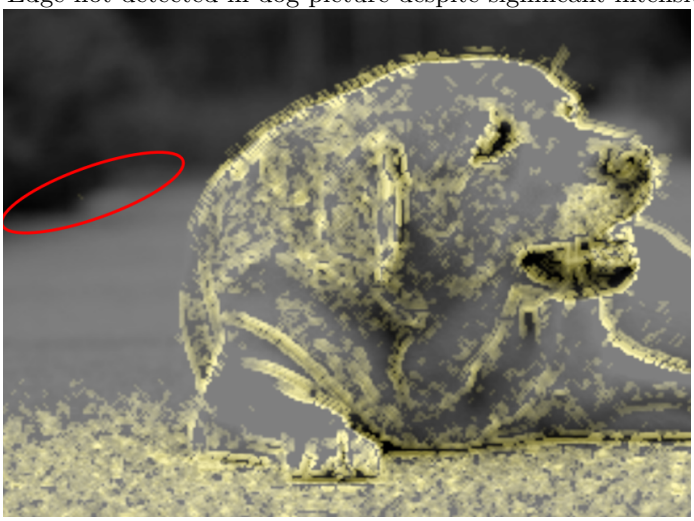


Figure 7: Dog toes have poorly-defined edges