TDT4195: Visual Computing Fundamentals

Image Processing – Assignment 2

Deadline: 08.11.2019

Name: Niklas Sølvberg

Task 1

a) $W_2 = \frac{(W_1 - F + 2P)}{S} + 1 = \frac{(W_1 - 5 + 2P)}{1} + 1 = W_1 + 2P - 4$. To make $W_2 = W_1$, we have to set $2P = 4$, which gives us a padding of 2 (on both sides, as $F_W = F_H$ and $S_W = S_H$).

b) $504 = \frac{512 - F + 2P}{S} + 1$ which gives us $504 = 512 - F + 1$, and from this we learn that the kernel must be of size 9×9.

c) Pooling each 2×2 square together with a stride of 2, gives us half the width and height, meaning the pooled feature maps will have dimensions 252×252.

d) Because we have 12 feature maps in the first layer, we must also have 12 pooled feature maps in the first layer, as the pooling is done for each individual feature layer.

e) $W_3 = \frac{W_2 - F + 2P}{S} + 1 = 252 - 3 + 1 = 250$. Because $H_3 = W_3$, the dimensions are 250×250.

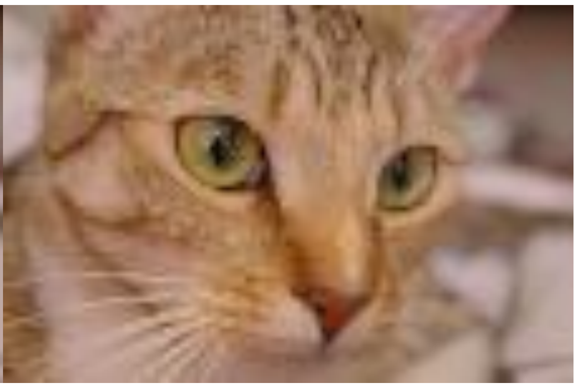
f) To find the number of parameters, we first calculate the number of parameters at each layer, and at the end add them up to get the total number of parameters in the network.

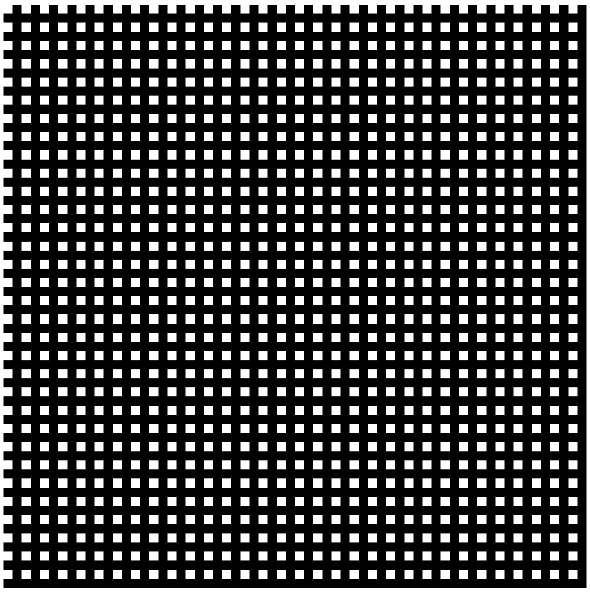
$$L_1 = \big((5 \times 5) + 1\big) \times 32 = 832$$
$$L_2 = \big((5 \times 5) + 1\big) \times 64 = 1664$$
$$L_3 = \big((5 \times 5) + 1\big) \times 128 = 3328$$
$$Parameters_{total} = L_1 + L_2 + L_3$$
$$Parameters_{total} = 5824$$

Task 2

a)

chelsea.png, 451×300

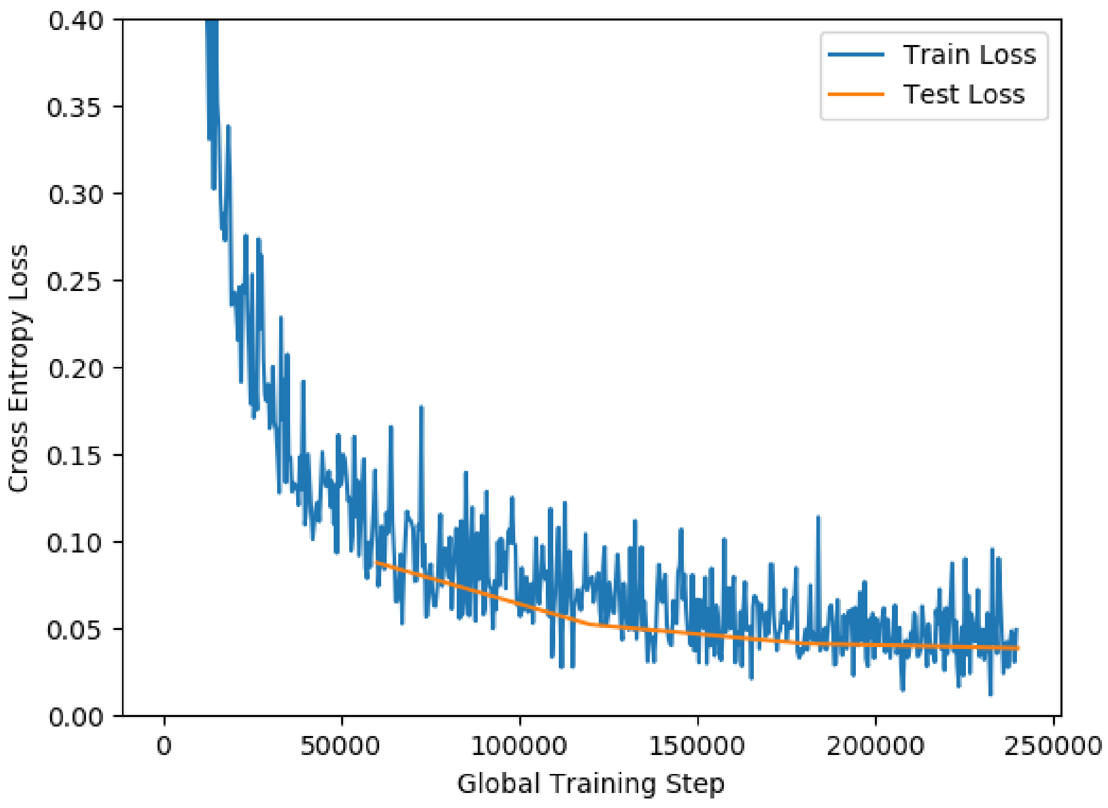

chelsea_maxpooled.png, 113×75


checkerboard.png, 64×64

checkerboard_maxpooled.png, 32×32

What has happened with the max-pooled checkerboard image is that the kernel used has a size of 2×2, which means that each pixel in the max-pooled image has the highest R, G, and B value from each 2×2 square in the input image. White being the highest possible RGB value (for each of the three colors), combined with the structure of the image (always exactly two white pixels in each 2×2 square) and the predetermined kernel size, means that each pixel in the max-pooled image will be white, resulting in an all-white image.

b)



Final Validation loss: 0.03896487135139619. Final Validation accuracy: 0.9875.

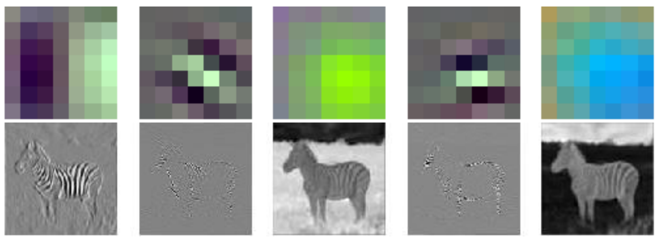
The cross-entropy loss is getting smaller by each epoch, but by smaller and smaller margins. I do not see any evidence of overfitting.

c)

Final Validation loss: 0.0245074814760022273. Final Validation accuracy: 0.9918.

The cross-entropy loss is getting smaller during the first few epochs (just like when having just four epochs), but it begins to go up and down repeatedly after the first 5 epochs. This looks like it could be the noise in the images being learned as features to look out for, and the results may therefore be less consistent because of it. This may therefore be evidence of overfitting taking place.

d)



e)

Filter + activation 1: It looks like this filter extracts wide, vertical stripes, and compared to filter-activation-pair 2 and 4, this makes sense comparing the filters and activations.

Filter + activation 2: It looks like this filter extracts diagonal stripes (upper-right to lower-right, easily seen in the filter), as the parts of the image that is the most visible in the activation is the parts where the zebra's stripes are diagonal.

Filter + activation 3: It looks like this filter extracts the inverse (negative) of the contrast (filter + activation 5) in the image.

Filter + activation 4: It looks like this filter extracts almost horizontal stripes (almost horizontal, but looking at the filter, it looks like it is a few degrees off, going slightly from lower-left to upper-right), as the part of the zebra showing (almost) horizontal stripes are the most visible in the activation.

Filter + activation 5: It looks like this filter extracts the contrast in the image. The darkest parts of the original image is pitch black, while the brighter parts of the original is brighter than before.

Task 3

a) Having a grayscale image and a kernel, both in the spatial domain, I would perform a discrete Fourier Transform on both (given that the image and the kernel is the same size, otherwise, padding is needed). Having an image and a kernel in the frequency domain, I would simply perform a pointwise multiplication of the image and the kernel. Next, I would perform the inverse discrete Fourier Transform, sending it back to the spatial domain. Finally, we would need to remove all imaginary parts from all entries of our image.

b) High-pass filters are filters that let high frequencies through while blocking low frequencies, and low-pass filters behave the opposite way, blocking high frequencies while letting through the low frequencies.

c) Because low frequencies are in the center of the kernel, and high frequencies are further out, I conclude that (a) and (b) is high-pass filters, as they are black in the center and has brighter spots further out, and that (c) is a low-pass filter, as it is bright in the center and darker further away from the center.

Task 4

a)

Figure 1 - Low-pass

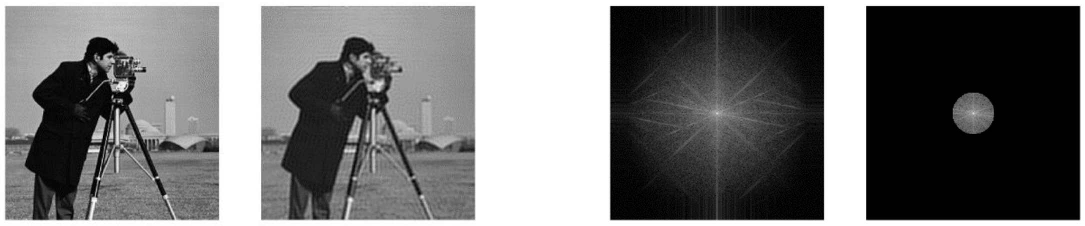
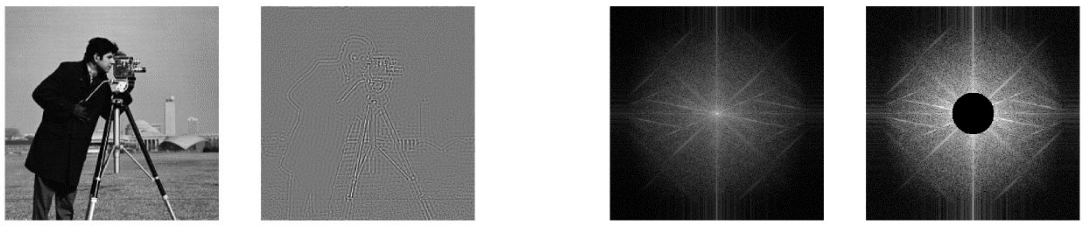


Figure 2 - High-pass



b)

Figure 3 - Horizontal sobel

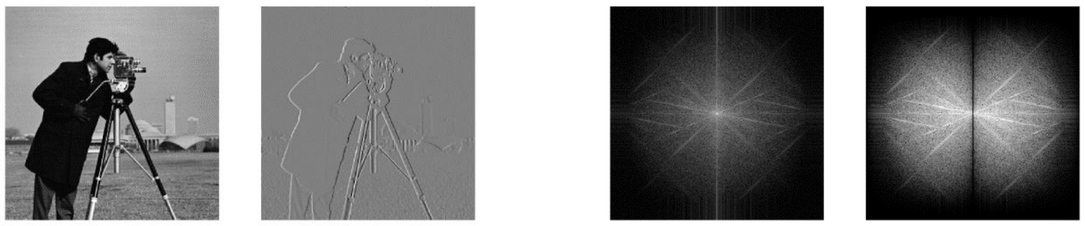
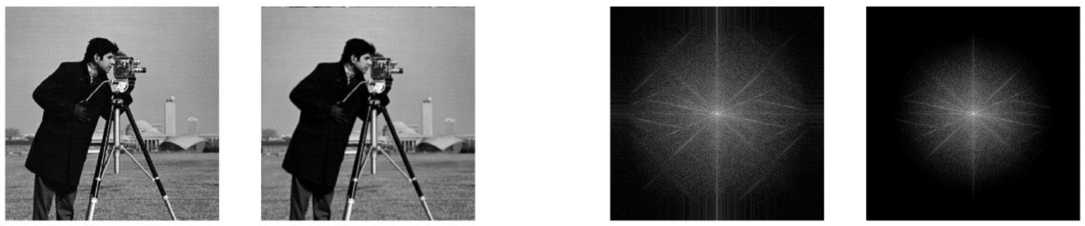
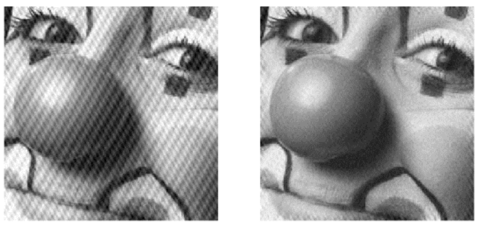


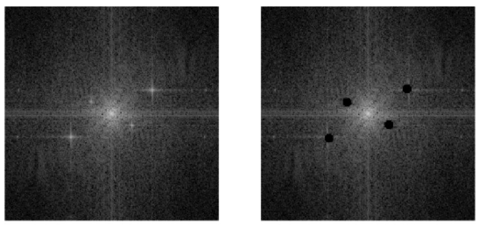
Figure 4 - Gaussian



c)

It seems this filter allows most frequencies to pass through just fine, but lowers certain ranges of frequencies, and thus quite a bit of the noise in the image disappears. This is very easy to spot in the images to the right, showing the amplitudes; apart from in the very center, the brightest spots are, after convolution, the darkest spots, as their frequency has been lowered to a high degree.





d)