TDT4195 Assignment 1

Olof Ljunggren

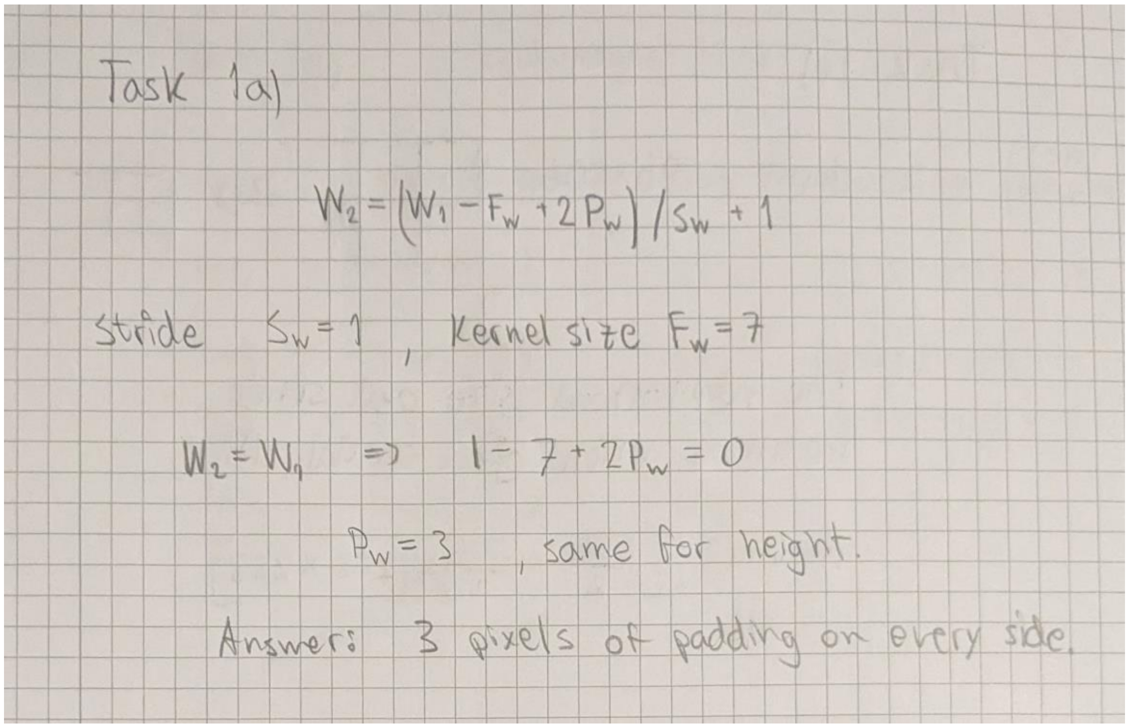
TDT4195 Image Processing Assignment 2

Task 1: Theory, Convolutional Neural Networks [1 point]

Task 1.a

[0.1pt] Given a single convolutional layer with a stride of 1, kernel size of 7 × 7, and 6 filters. If want the output shape (Height ×Width) of the convolutional layer to be equal to the input image, how much padding should I use on each side?

- The number of filters does not matter.



Task 1a)

$$W_2 = (W_1 - F_w + 2P_w)/S_w + 1$$

stride $S_w = 1$, kernel size $F_w = 7$

$W_2 = W_1 \Rightarrow 1 - 7 + 2P_w = 0$

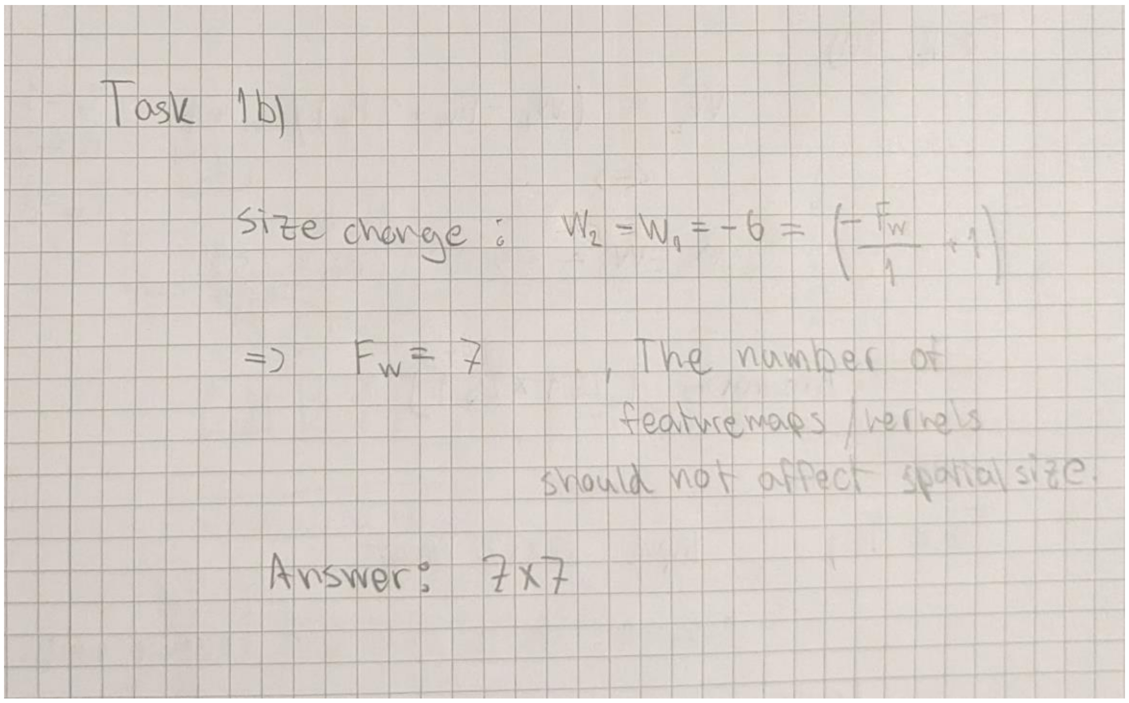
$P_w = 3$, same for height.

Answer: 3 pixels of padding on every side.

Task 1a.

Task 1.b

[0.2pt] You are told that the spatial dimensions of the feature maps in the first layer are 506 × 506, and that there are 12 feature maps in the first layer. Assuming that no padding is used, the stride is 1, and the kernel used are square, and of an odd size, what are the spatial dimensions of these kernels? Give the answer as (Height) × (Width).


Task 1b)

size change : $W_2 - W_1 = -6 = \left(-\frac{F_W}{1} + 1\right)$

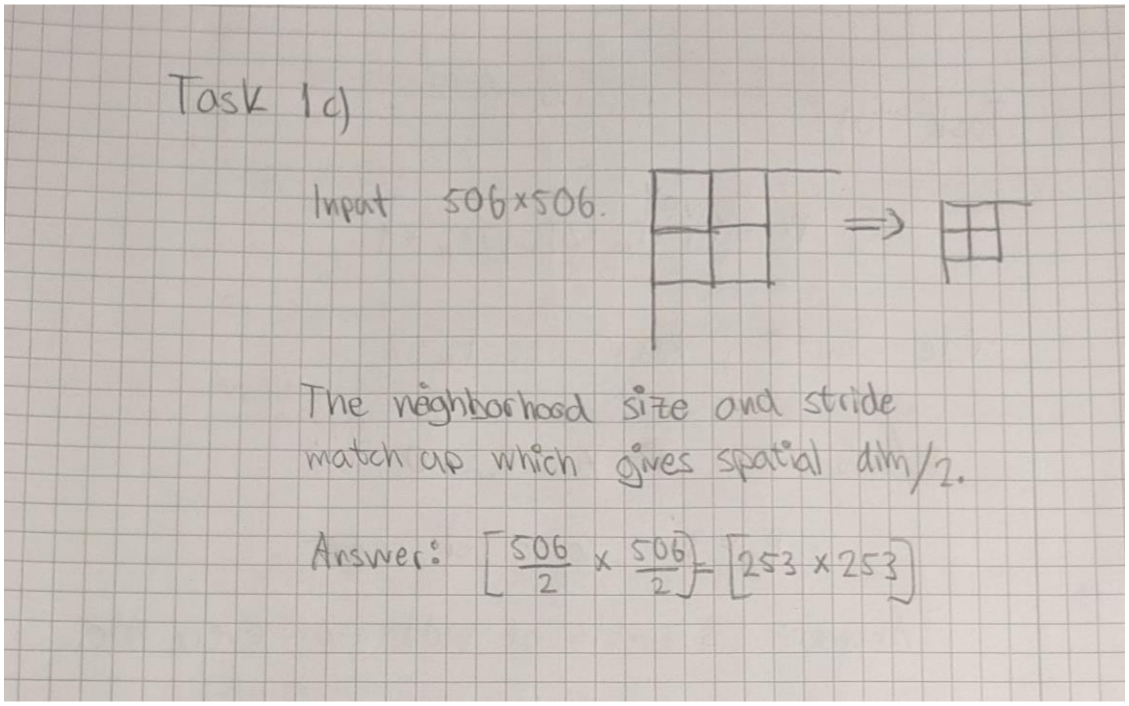
$\Rightarrow$ $F_W = 7$, The number of featuremaps / kernels should not affect spatial size.

Answer: 7x7

Task 1b.

Task 1.c

[0.2pt] If subsampling is done using neighborhoods of size 2 × 2, with a stride of 2, what are the spatial dimensions of the pooled feature maps in the first layer? (assume the input has a shape of 506 × 506). Give the answer as (Height) × (Width).


Task 1c)

Input 506×506.

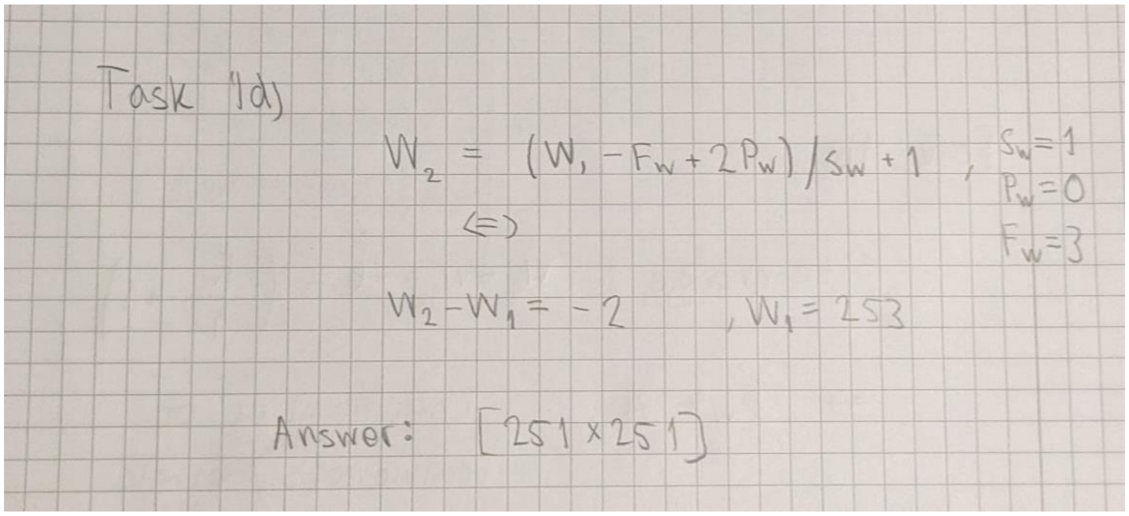
The neighborhood size and stride match up which gives spatial dim/2.

Answer: $\left[\frac{506}{2} \times \frac{506}{2}\right] = [253 \times 253]$

Task 1c.

Task 1.d

[0.2pt] The spatial dimensions of the convolution kernels in the second layer are 3 × 3. Assuming no padding and a stride of 1, what are the sizes of the feature maps in the second layer? (assume the input shape is the answer from the last task). Give the answer as (Height) × (Width).


Task 1d)

$$W_2 = (W_1 - F_W + 2P_W)/S_W + 1, \quad S_W = 1, \; P_W = 0, \; F_W = 3$$

$$\Leftrightarrow$$

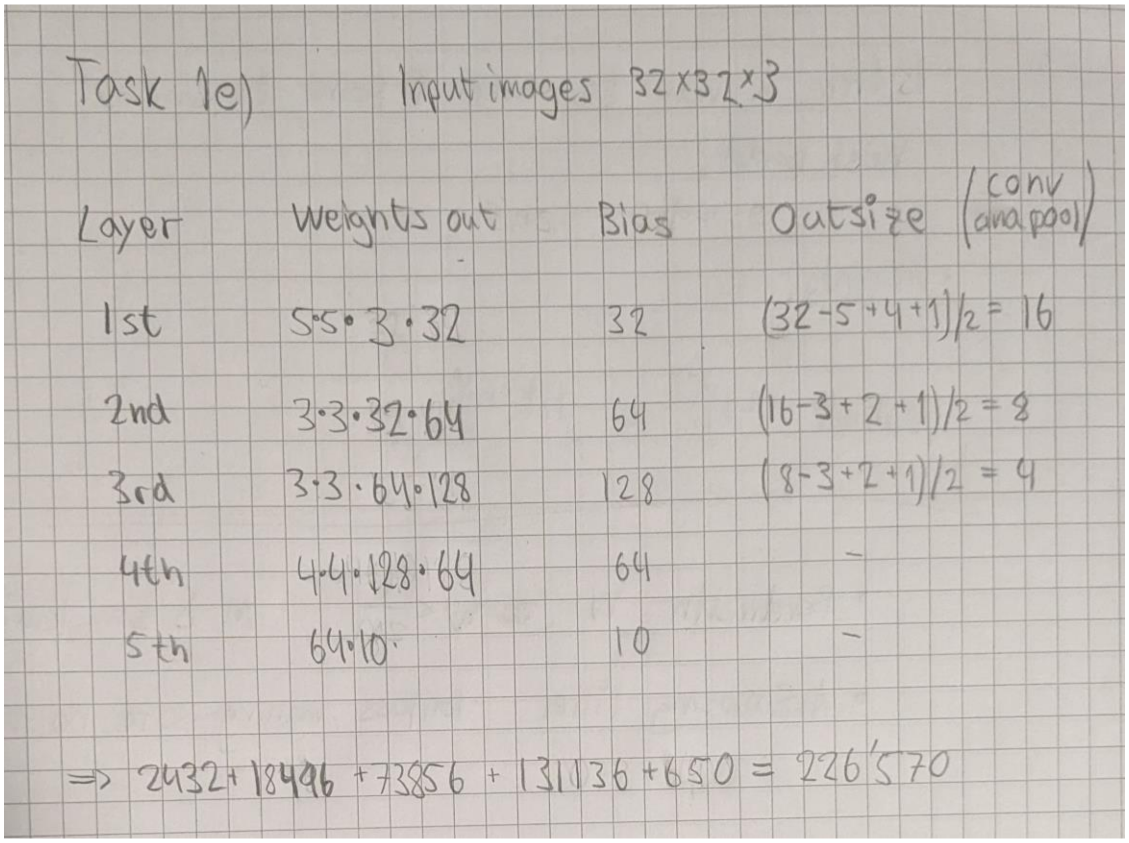
$$W_2 - W_1 = -2, \quad W_1 = 253$$

Answer: $[251 \times 251]$

Task 1d.

Task 1.e

[0.3pt] Table 1 shows a simple CNN. How many parameters are there in the network? In this network, the number of parameters is the number of weights + the number of biases. Assume the network takes in an 32 × 32 image.



Task 1e) Input images 32x32x3

Layer	Weights out	Bias	Outsize (conv and pool)
1st	5·5·3·32	32	(32-5+4+1)/2 = 16
2nd	3·3·32·64	64	(16-3+2+1)/2 = 8
3rd	3·3·64·128	128	(8-3+2+1)/2 = 4
4th	4·4·128·64	64	-
5th	64·10	10	-

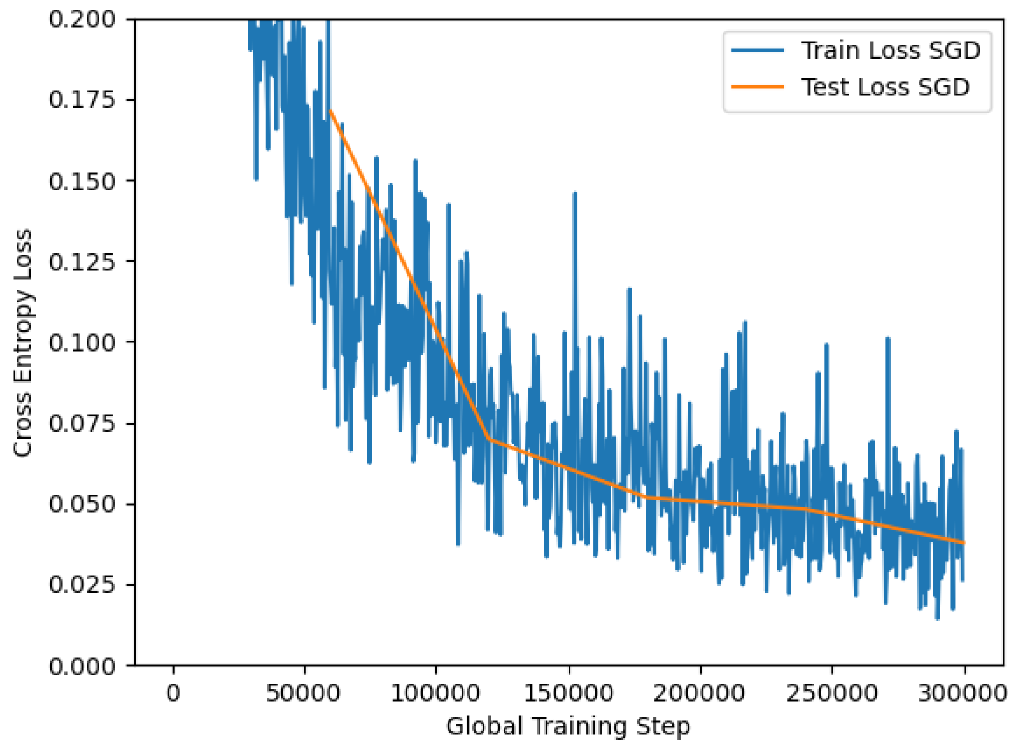
$\Rightarrow$ 2432 + 18496 + 73856 + 131136 + 650 = 226'570

Task 1e.

Task 2: Programming, Convolutional Neural Networks [2 points]

Task 2.a

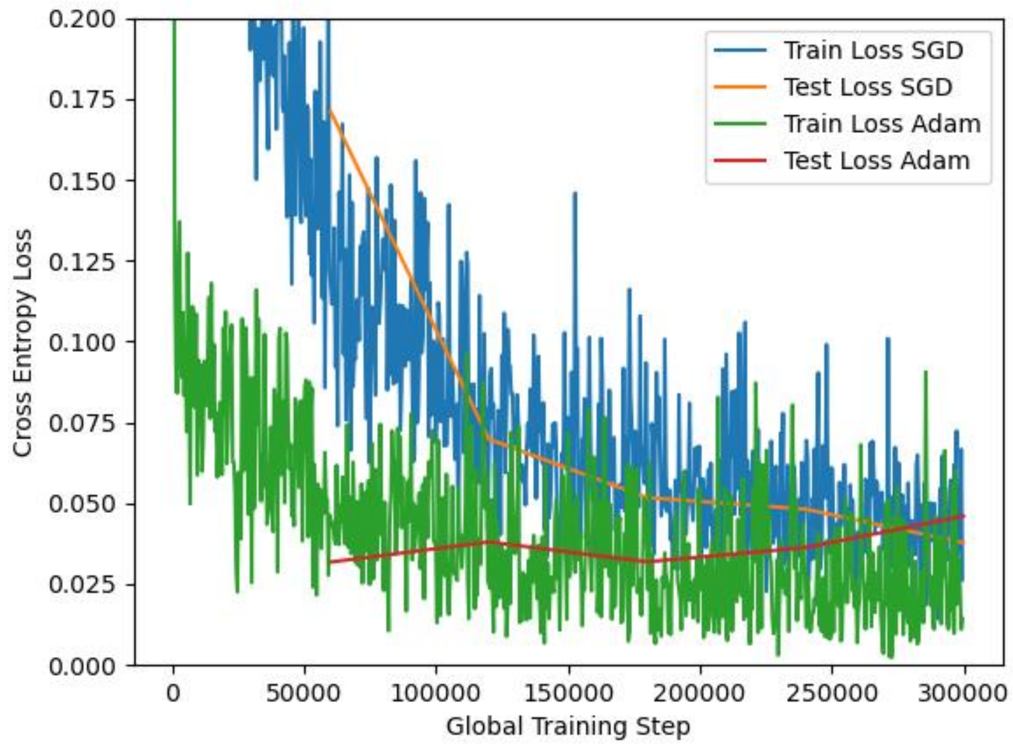
[0.7pt] Implement the network in Table 1. Implement this in the jupyter notebook (or python file) task2.py/ipynb. Report the final accuracy on the validation set for the trained network. Include a plot of the training and validation loss during training.



Loss plot CNN.

Task 2.b

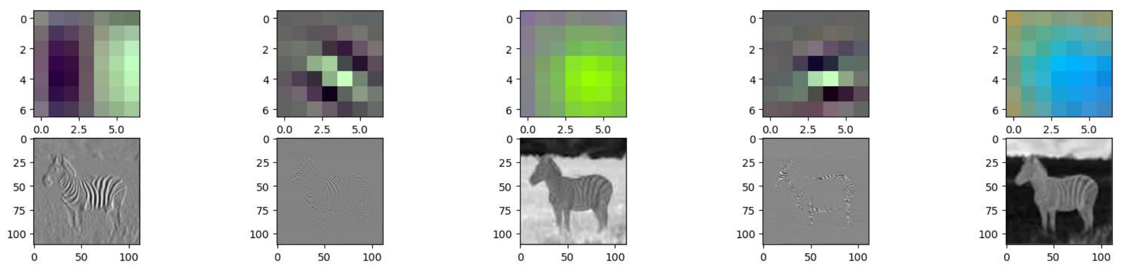
[0.3pt] Plot the training/validation loss from both models (the model with Adam and the one with SGD) in the same graph and include this in your report. (Note, you should probably change the plt.ylim argument to [0, 0.1]).



Loss plot SGD vs Adam optimizer.

Task 2.c

[0.5pt] Run the image zebra.jpg through the first layer of the ResNet50 network. Visualize the filter, and the grayscale activation of a the filter, by plotting them side by side. Use the pre-trained network ResNet50 and visualize the convolution filters with indices [5, 8, 19, 22, 34]. Include the visualized filters and activations in your report.



Zebra weights and corresponding activations.

Task 2.d

[0.5pt] Looking at the visualized filter, and its corresponding activation on the zebra image, describe what kind of feature each filter extracts. Explain your reasoning.

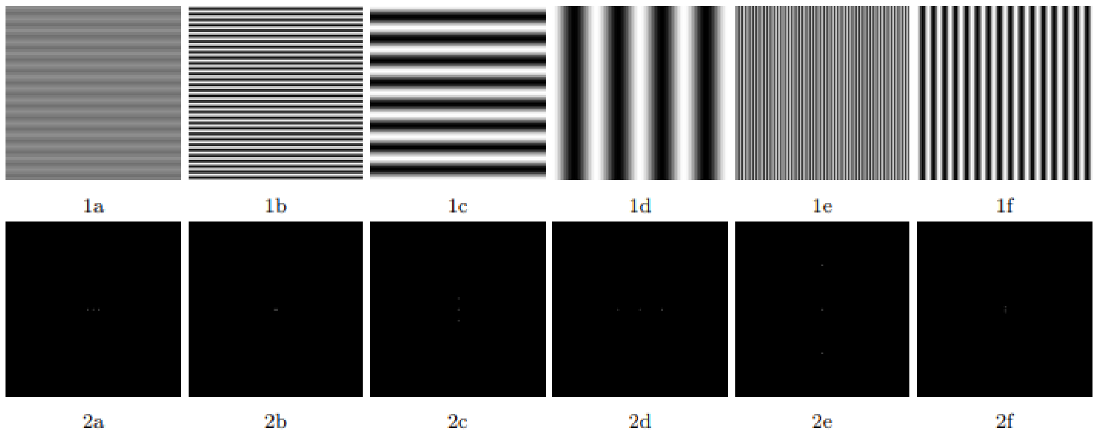
- The first filter seem to extract some vertical edges. Looks realistic considering the kernel. The second filter seems to extract some diagonal edges. The third filter extracts green colors from the picture. The fourth one seems to extract horizontal lines which in this case corrsponded to the contours of the zebra. The last and fifth filter extracts blue colors from the image.

Task 3: Theory, Filtering in the Frequency Domain [1 points]

Task 3.a

[0.6pt] Given the images in the spatial and frequency domain in Figure 3, pair each image in the spatial domain (first row) with a single image in the frequency domain (second row). Explain your reasoning.

- First of all we can distinguish pictures which varies sideways (x-direction). This means that 1d, 1e and 1f belongs to 2a, 2b and 2d. Since the discrete fourier transform is mirrored for negative frequencies we can study only positive ones. An impulse/point far away from the center in the frequency domain indicates high frequencies. With this reasoning we get that F{1e}=2b, F{1f}=2a, F{1d}=2d. Similarly we get that: F{1a}=2f, F{1b}=2c, F{1c}=2e.



Task 3a.

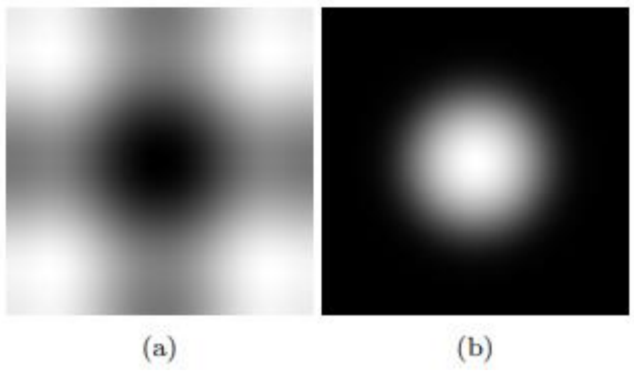
Task 3.b

[0.1pt] What are high-pass and low-pass filters?

- High-pass filters are filters that removes low frequencies and let high frequencies pass. These will have low values in the middle and large ones close to the edges. Low-pass filters is the opposite which instead amplify low frequencies compared to high which is attenuated. Low-pass filters will instead have high values in the center and low in the edges.

Task 3.c

[0.3pt] The amplitude |F{g}| of two commonly used convolution kernels can be seen in Figure 4. For each kernel (a, and b), figure out what kind of kernel it is (high- or low-pass). Shortly explain your reasoning.



Task 3c.

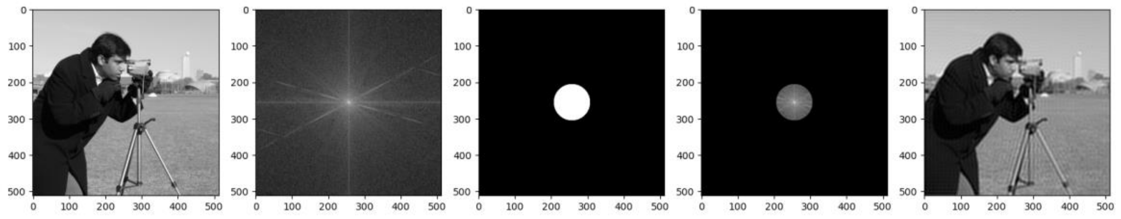
- Just looking at equation 3 we can easily see that the lowest frquencies are mapped onto lowest u and v values. In other words if we were to create a low-pass kernel we would use values close to 1 close to the center of the kernel (in the frequency domain) and values closer to 0 at the edges. Similar reasoning as in the previous exercise. Hence (a) is the fourier transform of a high-pass and (b) is the fourier transform of a low-pass.

Task 4: Programming, Filtering in the Frequency Domain [2 points]

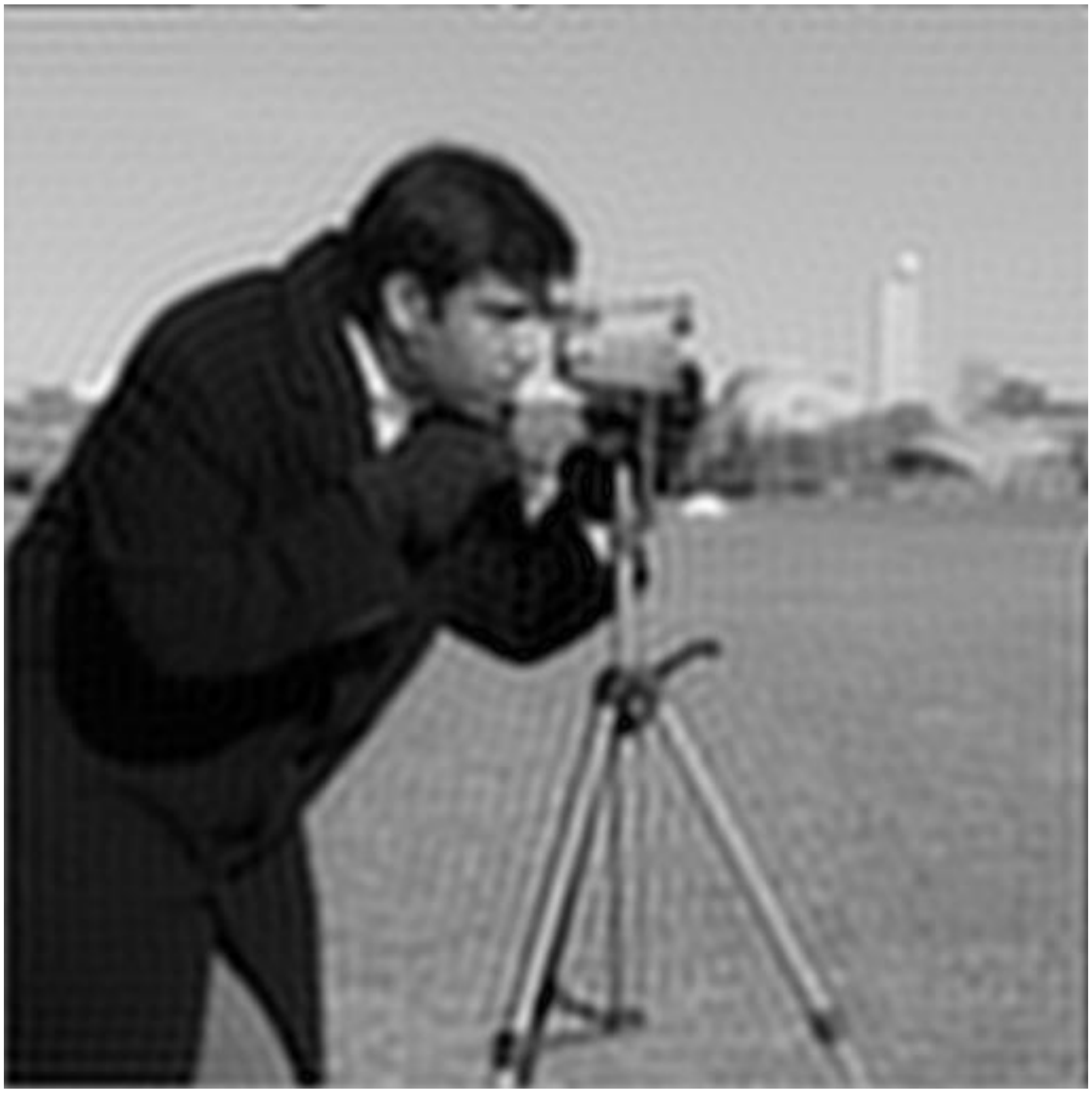
Task 4.a

[0.5pt] Implement a function that takes an grayscale image, and a kernel in the frequency domain, and applies the convolution theorem (seen in Equation 4). Try it out on a low-pass filter and a high-pass filter on the grayscale image ”camera man”(im = skimage.data.camera()). Include in your report the filtered images and the before/after amplitude |F{f}| of the transform.

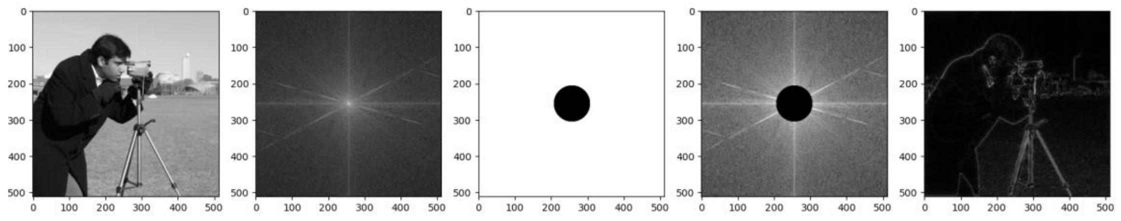
- The ringing effekt will be due to frequency domain aliasing. This happens for large frequencies and is explained by the sampling theorem. This can be solved by zero padding the image to double size.



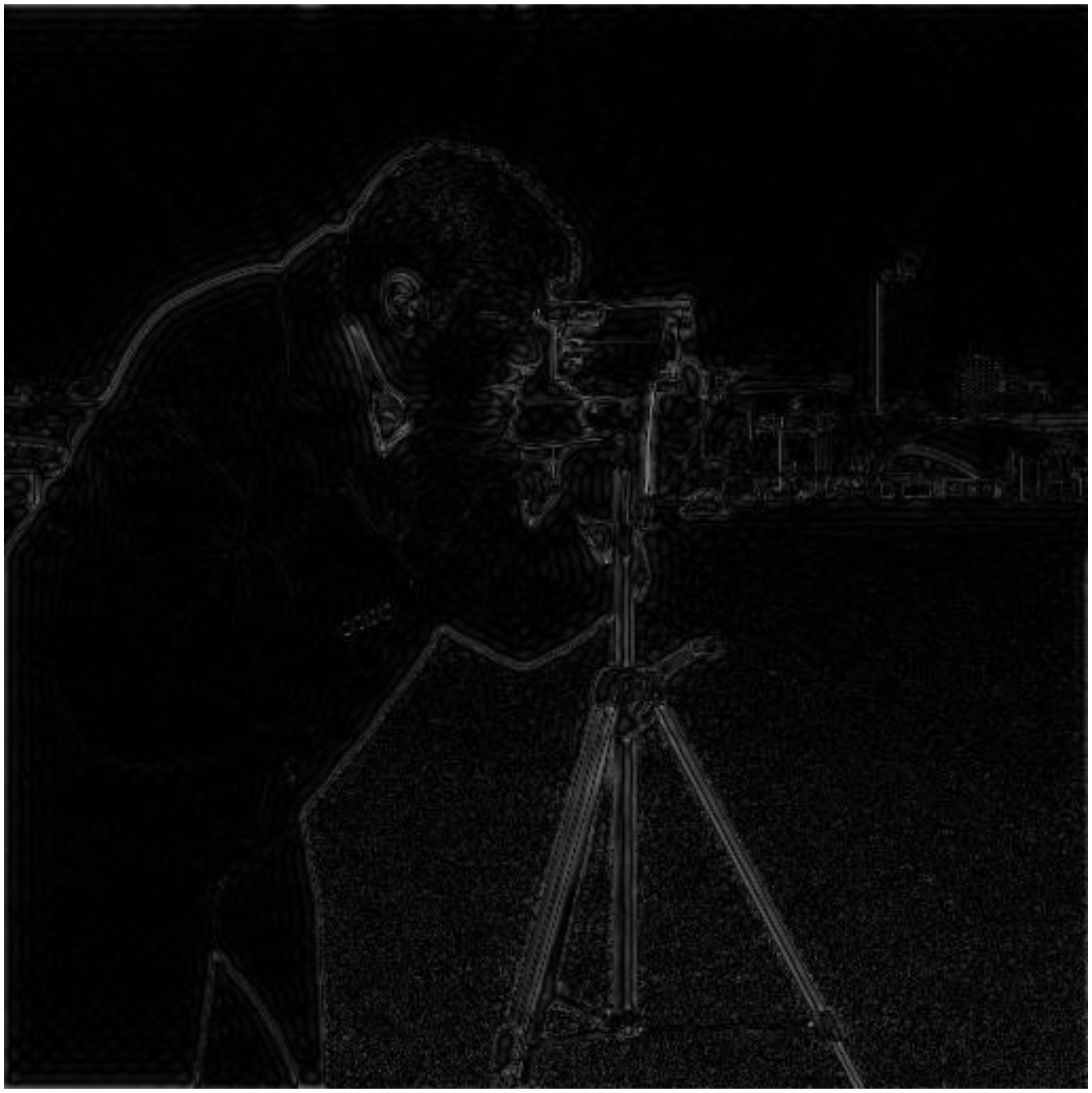
Low pass filtering.



Close look low pass camera man.



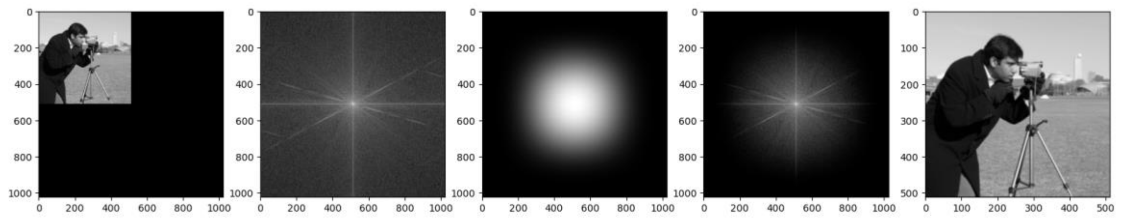
High pass filtering.


Close look high pass camera man.

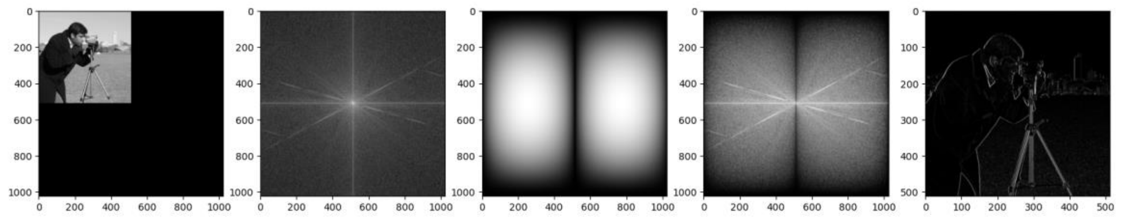
Task 4.b

[0.2pt] Implement a function that takes an grayscale image, and a kernel in the spatial domain, and applies the convolution theorem. Try it out on the gaussian kernel given in assignment 1, and a horizontal sobel filter (Gx).

- As we can see the gaussian kernel has kind of a lowpass effect and the sobel kernel has vertical edge detection. (I some sense highpass effect).



Gaussian kernel camera man.

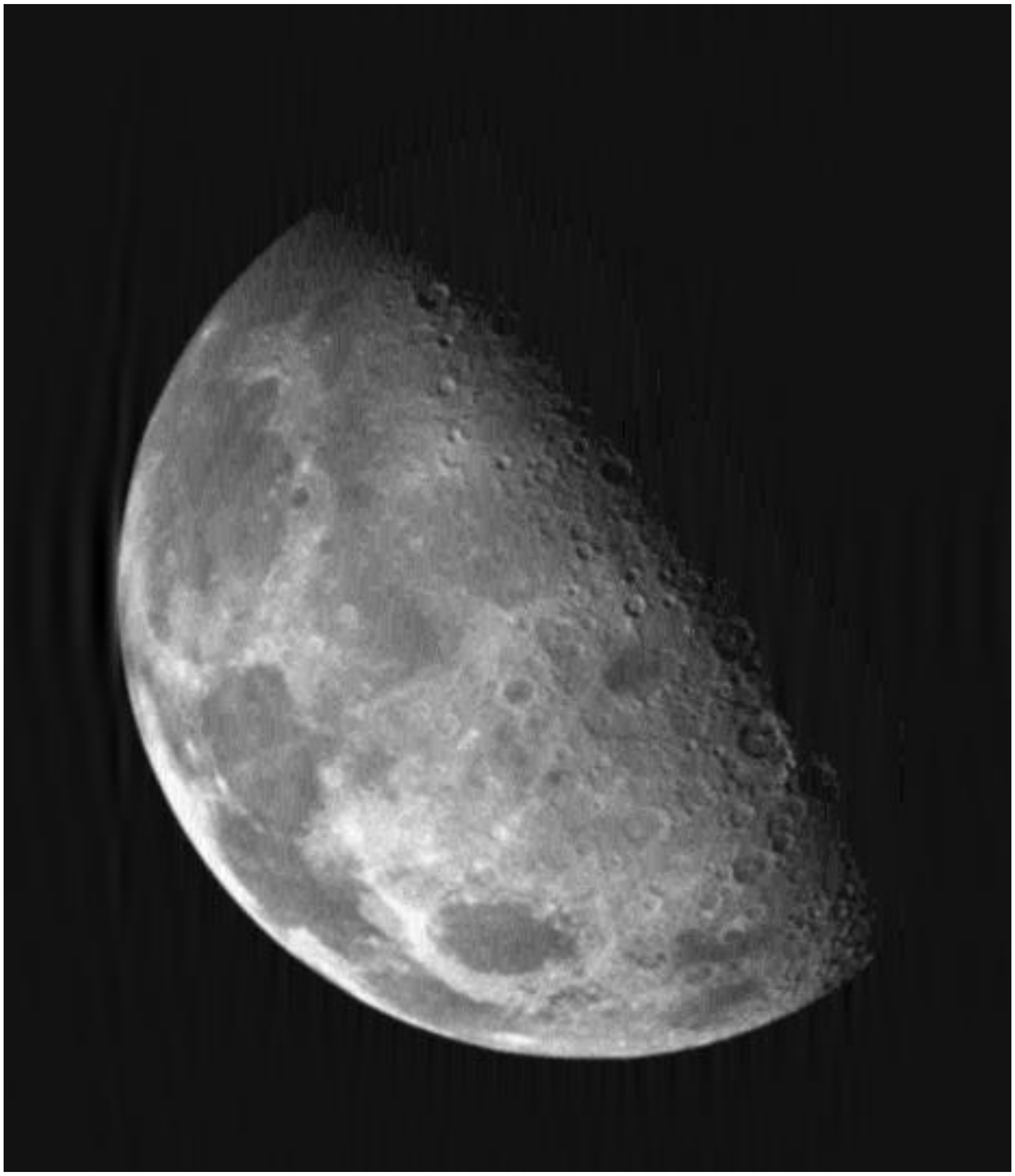


Close look high pass camera man.

Task 4.c

[0.7pt] Use what you've learned from the lectures and the recommended resources to remove the noise in the image seen in Figure 5a. Note that the noise is a periodic signal.

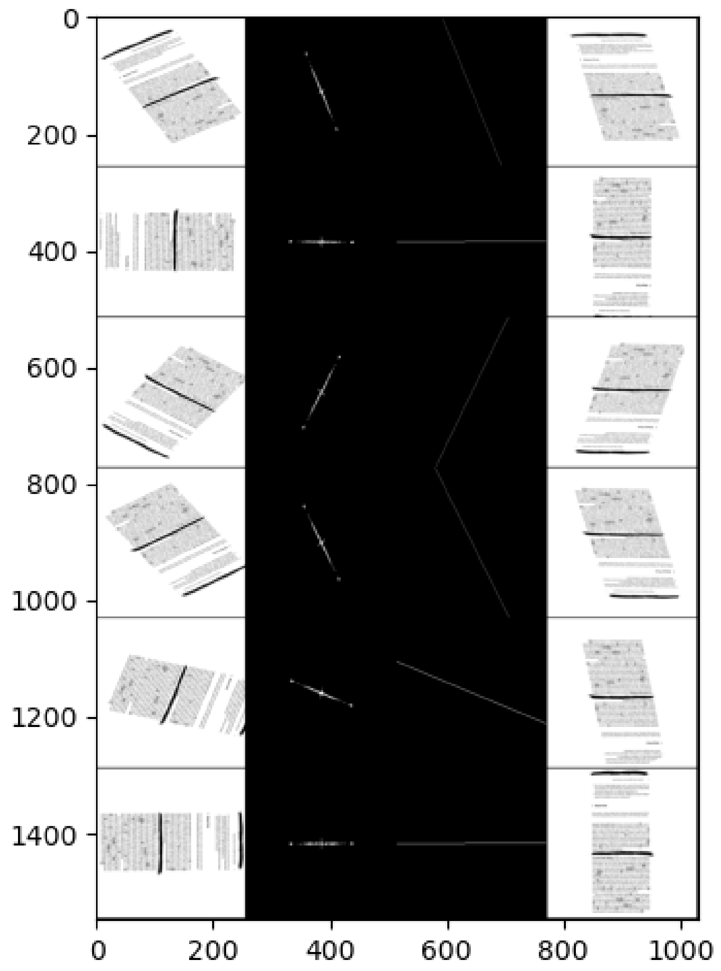
- By taking the fourier transform we can find som noise. By applying a allpass filter with a zero line we can remove that noise.



Close look high pass camera man.

Task 4.d

[0.6pt] Now we will create a function to automatically find the rotation of scanned documents, such that we can align the text along the horizontal axis.



Close look high pass camera man.