TDT4195 Visuell Databehandling

Vegard Iversen and Sebastian Skogen Raa

October 2021

Image Processing - Assignment 2 - Visual Computing Fundamentals

Task 1: Theory

Task 1a

Question
Given a single convolutional layer with a stride of 1, kernel size of 5 × 5, and 6 filters. If I 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?

Answer
Can use this formula: $P = \frac{((S-1)\cdot W - S + F)}{2}$, with F = filter size, S = stride, W = input size. With stride = 1 we can simplify to $P = \frac{F-1}{2}$. With kernel size of 5x5 we get a padding of $P = \frac{5-1}{2} = 2$

Task 1b

Question
You are told that the spatial dimensions of the feature maps in the first layer are 504 × 504, 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). **Answer**

by using equations in the assignment

$$W_2 = \frac{W_1 - F_W + 2P_W}{S_W} + 1 \tag{1}$$

$$H_2 = \frac{H_1 - F_H + 2P_H}{S_H} + 1 \tag{2}$$

We can solve for F_W or F_H (since its a square) By doing this we get

$$\begin{aligned} 504 &= \frac{512 - F_W + 2\cdot 0}{1} + 1 \\ F_W &= 9 \end{aligned} \tag{3}$$

Therefor the kernel size is 9 × 9.

Task 1c

Question
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 504×504). Give the answer as (Height) $\times$ (Width). **Answer**

$$H_2 = \frac{H_1 - F_H + 2P_H}{S_H} + 1 = \frac{504-2+0}{2} + 1 = 252$$
$$W_2 = \frac{W_1 - F_W + 2P_W}{S_W} + 1 = \frac{504-2+0}{2} + 1 = 252$$

the output will be 252×252

Task 1d

Question
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) $\times$ (Width). **Answer**
$$H_2 = \frac{H_1 - F_H + 2P_H}{S_H} + 1 = \frac{252-3+0}{1} + 1 = 250$$
$$W_2 = \frac{W_1 - F_W + 2P_W}{S_W} + 1 = \frac{252-3+0}{1} + 1 = 250$$

Task 1e

Question
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.

Answer
Number of parameters in layer 1 (Conv2D) and assuming input image is 32x32x1 (greyscale): $(5 \cdot 5 \cdot 1) + 1) \cdot 32 = 832$
Layer 2: $(3 \cdot 3 \cdot 32) + 1) \cdot 64 = 18496$
Layer 3: $(3 \cdot 3 \cdot 64) + 1) \cdot 128 = 73856$
Layer 4: $(32 \cdot 32 \cdot 2 + 1) \cdot 64 = 131136$
Layer 5: $(64 + 1) \cdot 10 = 650$

Task 2a

Question
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. By looking at the final train/validation loss/accuracy, do you see any evidence of overfitting? Shortly summarize your reasoning.

Answer

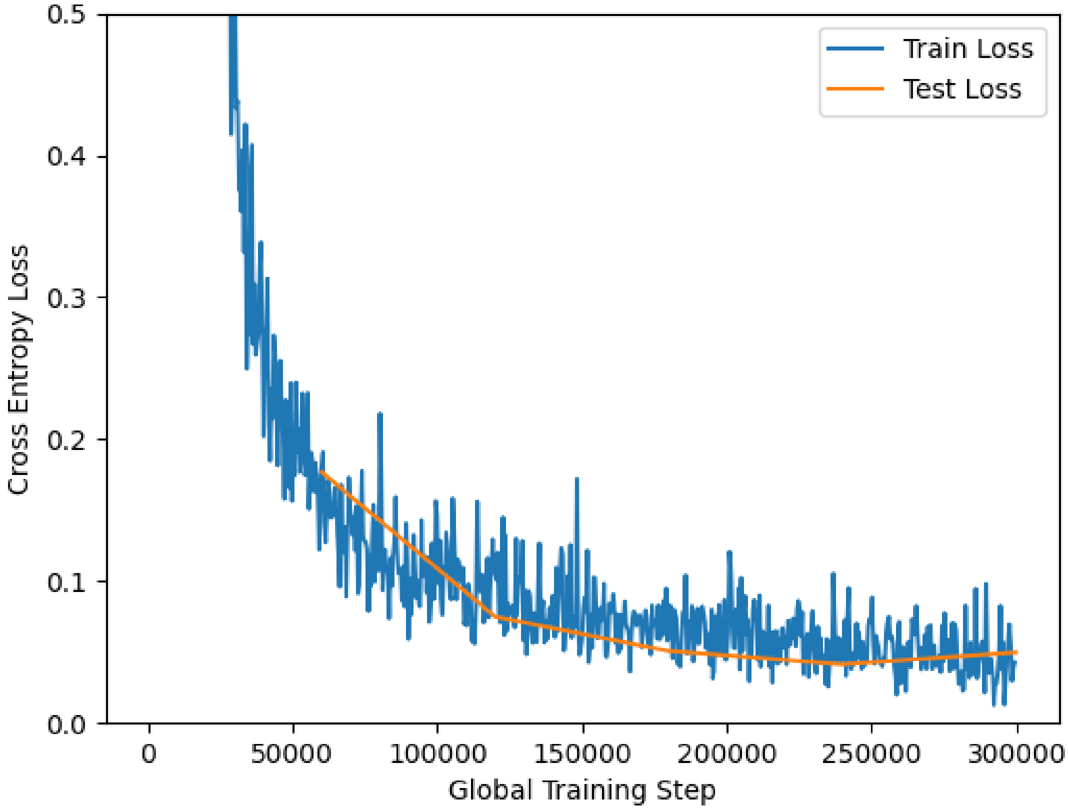

Figure 1: 2a

Final Test loss: 0.04972277681608714. Final Test accuracy: 0.9832. A small increase in loss in the end seems to indicate some overfitting.

Task 2b

Question
The optimizer in pytorch is the method we use to update our gradients. Till now, we have used standard stochastic gradient descent (SGD). Understanding what the different optimizers do is out of the scope of this course, but we want to make you aware that they exist. 1 Adam is one of the most popular optimizers currently. Change the SGD optimizer to Adam (use torch.optim.Adam instead of torch.optim.SGD), and train your model from scratch. Use a learning rate of 0.001. 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]).

Answer

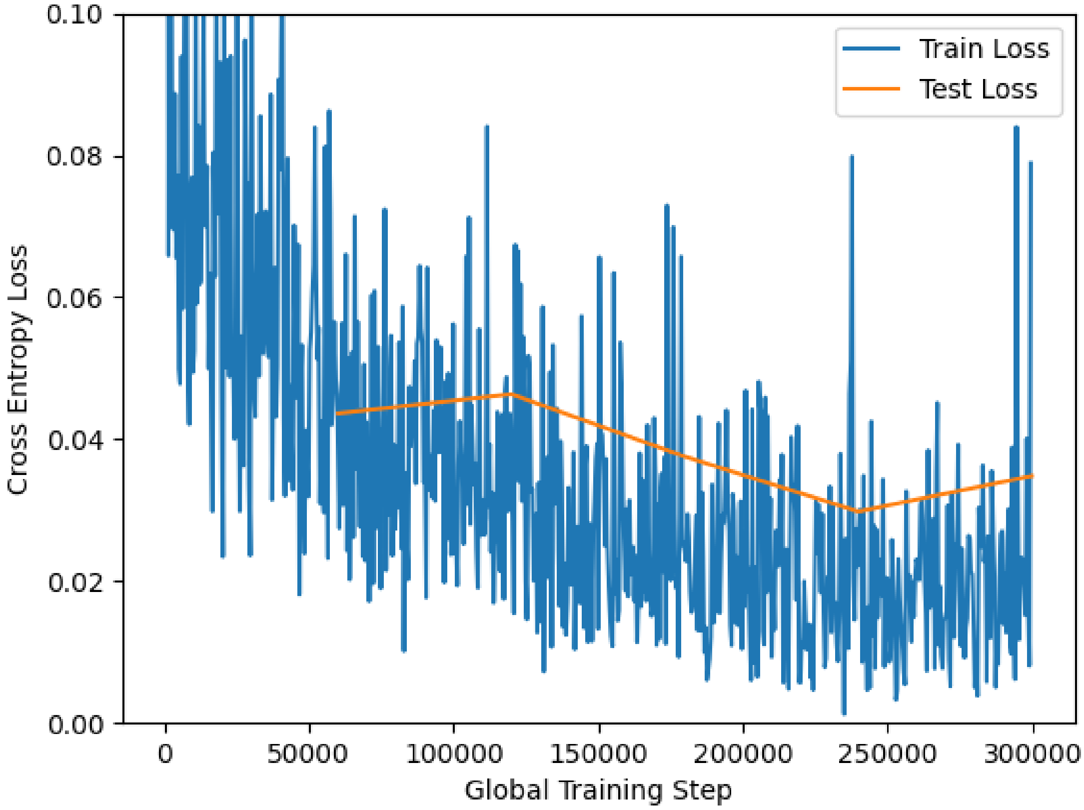

Figure 2: Again a small increase in loss in the end seems to indicate some overfitting, however it is harder to tell.

Final Test loss: 0.03481642094069175. Final Test accuracy: 0.9902.

Task 2c

Question
Interpreting CNNs is a challenging task. One way of doing this, is to visualize the learned weights in the first layer as a K × K × 3 image 2 , where K is the kernel size.. Understanding what the filter does can be difficult. Therefore, we can visualize the activation by passing an image through a given filter. The result of this will be a grayscale image. 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

Answer

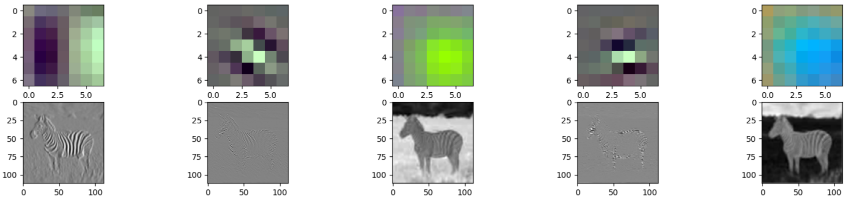

Figure 3: 2c: Interpreting CNNs.

Task 2d

Question
Looking at the visualized filter, and its corresponding activation on the zebra image, describe what kind of feature each filter extracts. Explain your reasoning. **Answer**

- The first filter extracts the vertical edges
- The 2 filter extracts the diagonal edges
- The 3 filter extracts green color component
- The 4 filter extracts the small details
- The 5 filter extract blue color component

Task 3a

Question
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.
Answer
Horizontal lines becomes vertical points in frequency domain therefor since the further away from the center the higher the frequency. The same goes for vertical lines, but becomes horizontal points instead.

1. 1a $\rightarrow$ 2e
2. 1b $\rightarrow$ 2c
3. 1c $\rightarrow$ 2f
4. 1d $\rightarrow$ 2d
5. 1f $\rightarrow$ 2a
6. 1f $\rightarrow$ 2a

Task 3b

Question
What are high-pass and low-pass filters? **Answer**
A high-pass filter is a filter that lets high frequencies pass and blocks low frequencies. Low-pass filter is a filter that lets low frequencies pass and blocks for high frequencies.

Task 3c

Question
The amplitude —Fg— 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.

Answer
b is i low pass filter since its lets the frequencies near the center pass (white).
a is a high pass filter since it lets the higher frequencies in the corners pass.

One can think of the image as a coordinate system, where the center is close to zero (low frequencies) and the edges are higher values (high frequencies).

Task 4a

Question
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 —Ff— of the transform. Make sure to shift the zero-frequency component to the center before displaying the amplitude. 8 Implement this in the function convolve_im in task4a.pytask4a.ipynb. The high-pass and low-pass filter is already defined in the starter code. You will observe a "ringing" effect in the filtered image. What is the cause of this? **Answer**

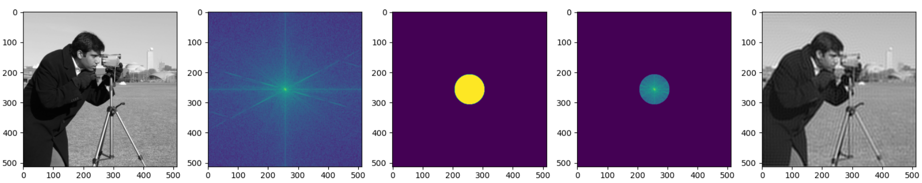

Figure 4: 4a high-pass filter

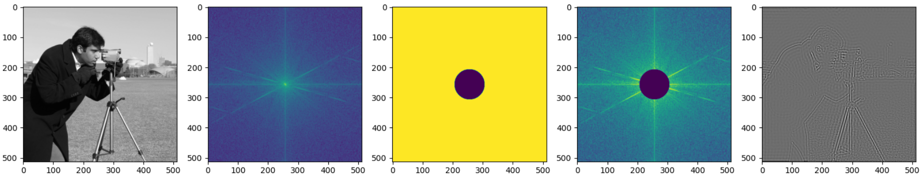

Figure 5: 4a low-pass filter

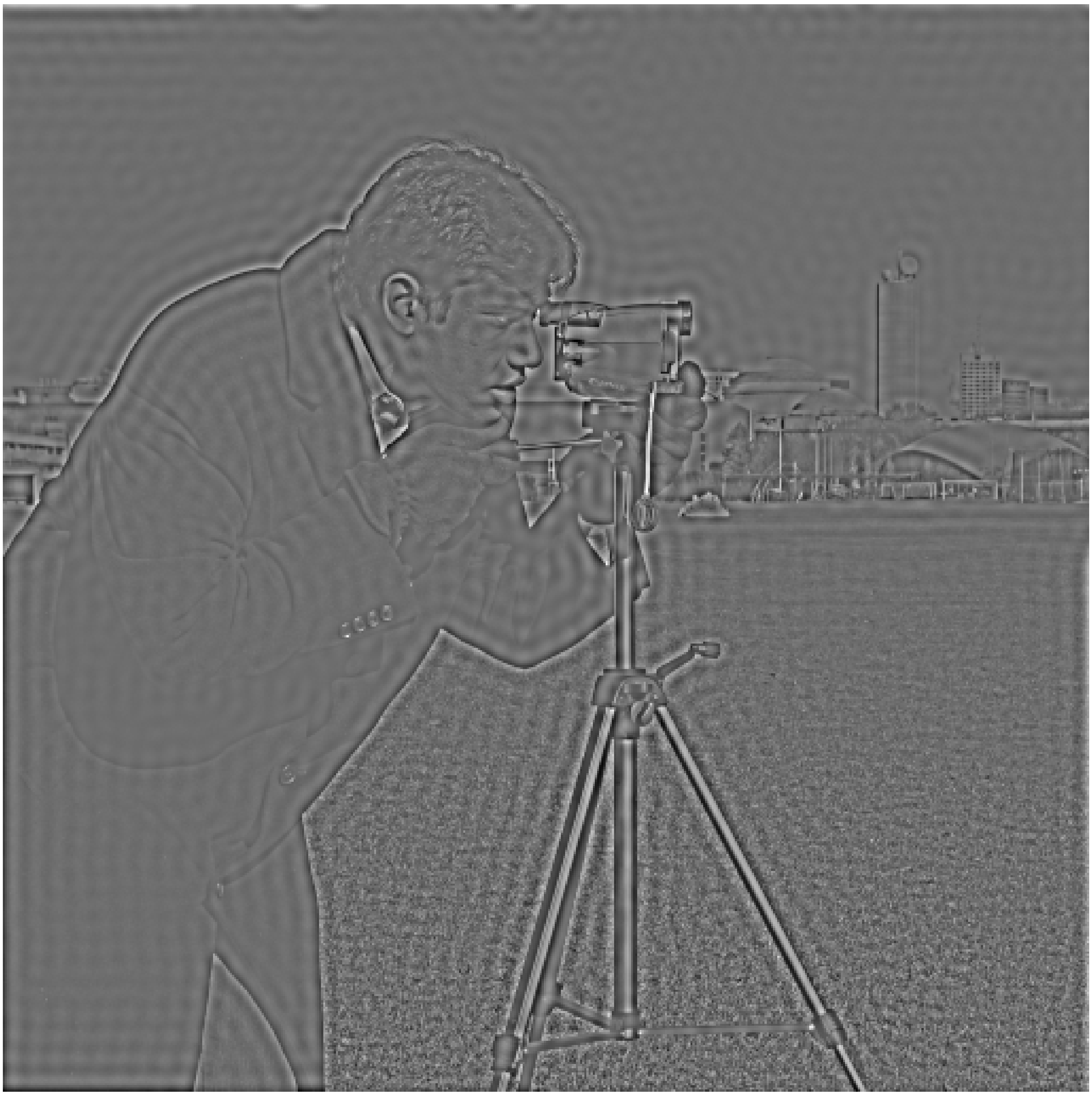

Figure 6: 4a high-pass cameraman

Figure 7: 4a low-pass cameraman

Task 4b

Question
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). Include in your report the filtered images and the before/after amplitude —Ff— of the transform. Make sure to shift the zero-frequency component to the center before displaying the amplitude. Implement this in the function convolve_im in task4b.pytask4b.ipynb. The gaussian and sobel filter are already defined in the starter code. **Answer**

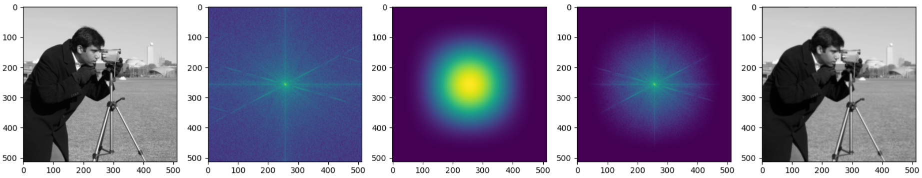

Figure 8: 4b Gaussian

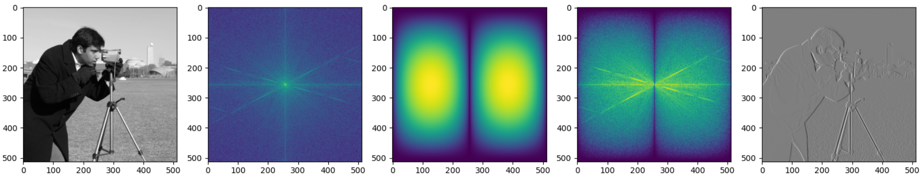

Figure 9: 4b sobel

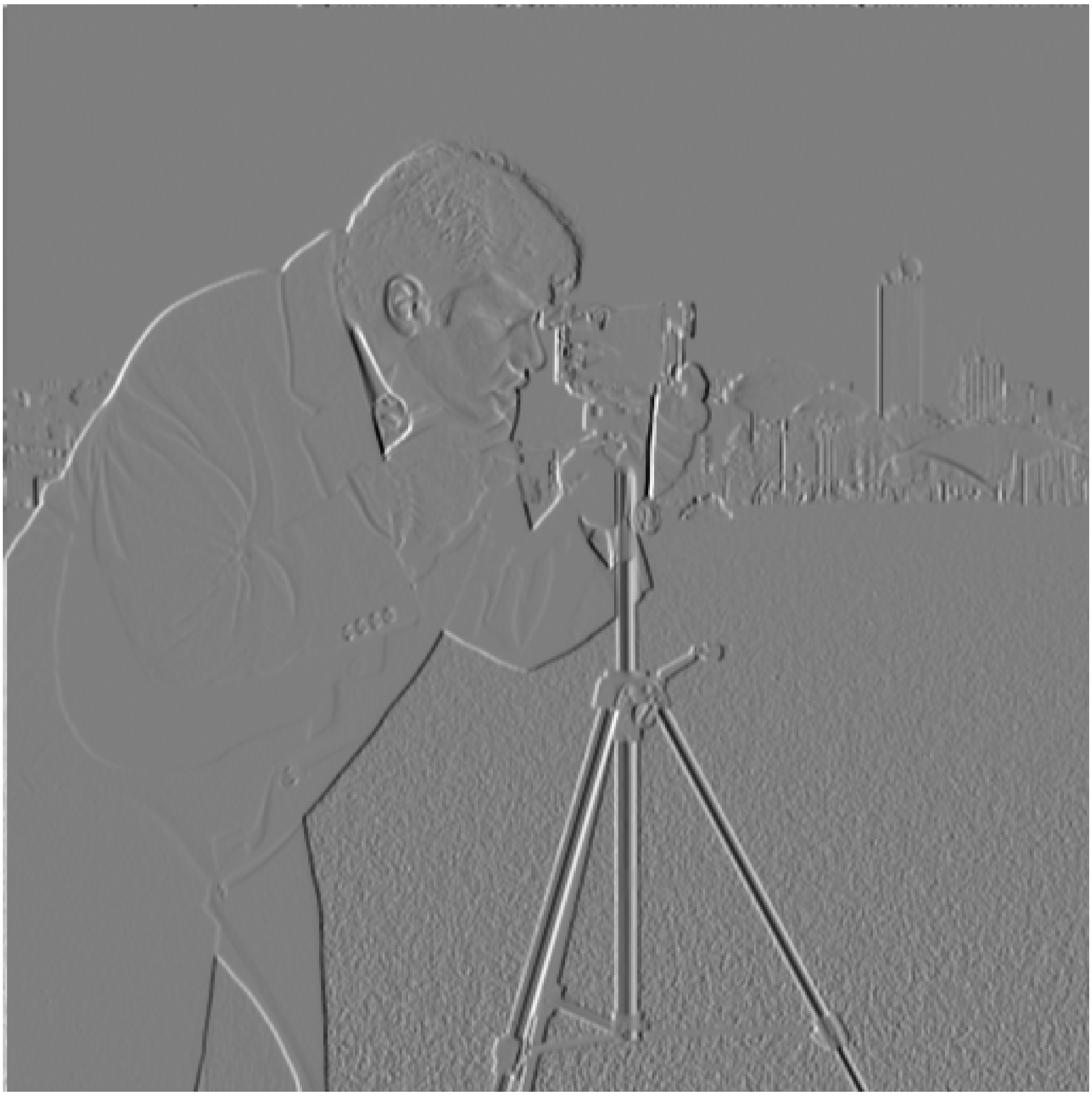

Figure 10: 4b sobel

Figure 11: 4b Gaussian

Task 4c

Question
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. Also, the result you should expect can be seen in Figure 5b Include the filtered result in your report. Implement this in the file task4c.pytask4c.ipynb **Answer**

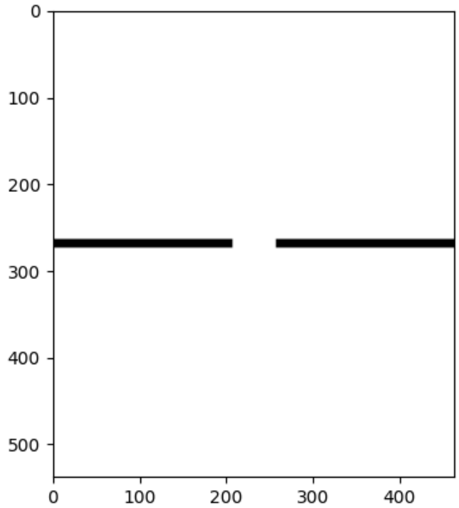

Figure 12: The filter used.

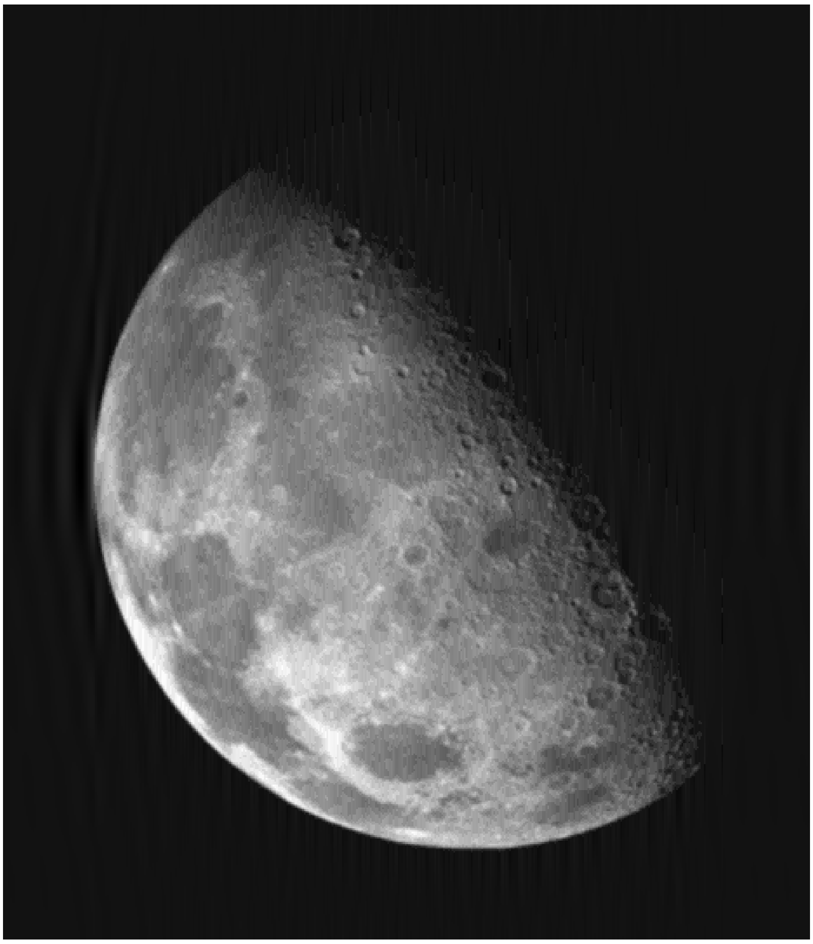

Figure 13: The filtered moon.

Task 4d

Question
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. You will use the frequency domain to extract a binary image which draws a rough line describing the rotation of each document. From this, we can use a hough transform to find a straight line intersecting most of the points in the binary image. When we have this line, we can easily find the rotation of the line and the document. Your task is to generate the binary image by using the frequency spectrum. See Figure 6 which shows you what to expect. We've implemented most of the code for you in this task; you only need to alter the function create_binary_image in task4d.pytask4.ipynb. Include the generated image in your report (similar to Figure 6).

Answer

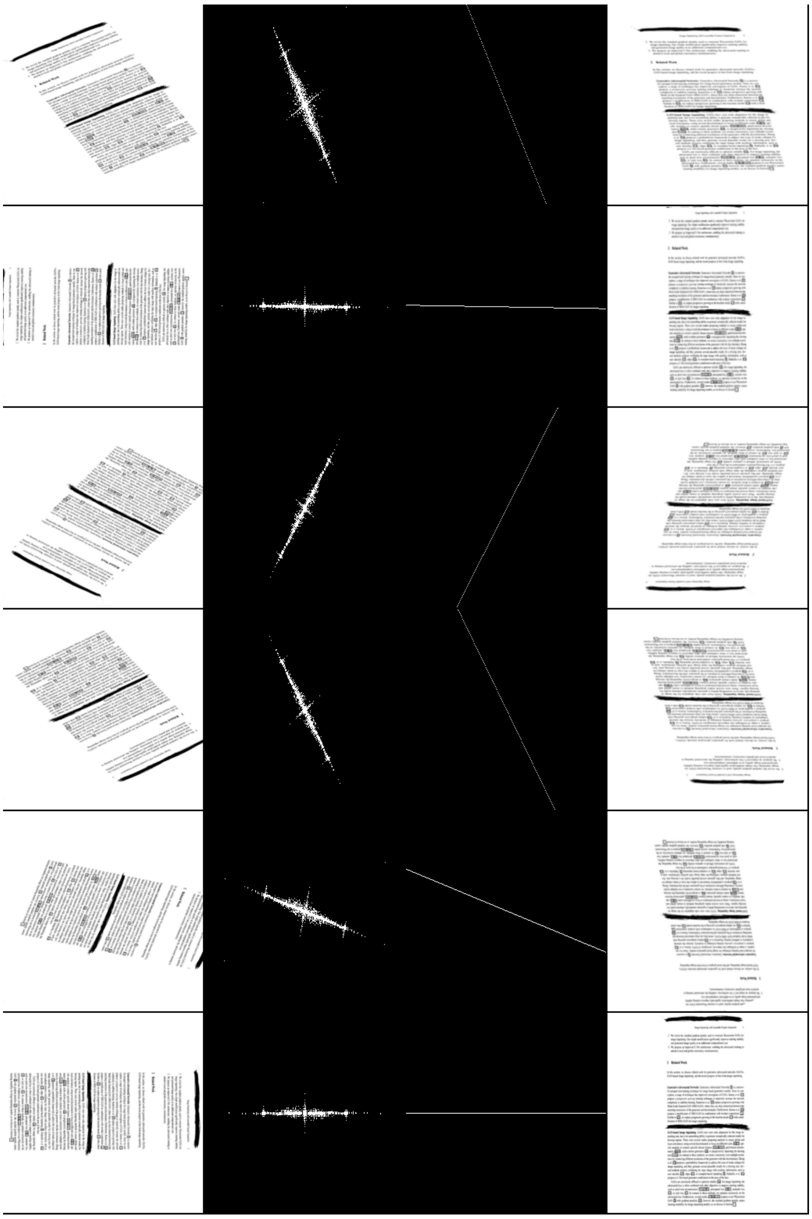

Figure 14: Rotation correction using Yen threshold on frequency domain.