# TDT4195: Visual Computing Fundamentals

## Image Processing - Assignment 2 Report

**10th November 2023**

**Oluwatobi OJEKANMI**

## 1 Convolutional Neural Networks Theory

Given

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

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

### (1a.)

Substituting the given values of $F_H = F_W = 7$ and $S_H = S_W = 1$ into equations $\boxed{1}$ and $\boxed{2}$, we have

$$\begin{aligned} W_2 &= W_1 - 7 + 2P_W + 1 \\ W_2 &= W_1 - 6 + 2P_W \end{aligned}$$

$$\begin{aligned} H_2 &= H_1 - 7 + 2H_W + 1 \\ H_2 &= H_1 - 6 + 2H_W \end{aligned}$$

Therefore, for $W_1 = W_2$ and $H_1 = H_2$,

$$2P_W - 6 = 0 \quad \text{and} \quad 2P_H - 6 = 0$$
$$P_W = P_H = \mathbf{3}$$

Therefore, for the conditions $W_1 = W_2$ and $H_1 = H_2$ to be satisfied, a padding of 3 on all sides or a total $6 \times 6$ padding is required.

### (1b.)

Substituting the given values of $H_1 = W_1 = 512$, $H_2 = W_2 = 506$, $P_H = P_W = 0$, and $S_H = S_W = 1$ into equations $\boxed{1}$ and $\boxed{2}$, we have

$$\begin{aligned} 506 &= \frac{512 - F_W + 2(0)}{1} + 1 \\ F_W &= 513 - 507 \\ F_W &= \mathbf{7} \end{aligned}$$

$$\begin{aligned} 506 &= \frac{512 - F_H + 2(0)}{1} + 1 \\ F_H &= 513 - 507 \\ F_H &= \mathbf{7} \end{aligned}$$

Therefore a 7 × 7 kernel size is required to obtain the given output dimension from the given input size and convolution layer details (i.e., padding and strides).

## (1c.)

If subsampling is done using neighborhoods of size 2 × 2 with a stride of 2, the spatial dimensions of the pooled feature maps in the first layer can be calculated using equations $\boxed{1}$ and $\boxed{2}$ as follows:

$$W_2 = \frac{506 - 2 + 2(0)}{2} + 1 = \frac{504}{2} + 1$$
$$W_2 = 252 + 1 = \mathbf{253}$$

$$H_2 = \frac{506 - 2 + 2(0)}{2} + 1 = \frac{504}{2} + 1$$

$$H_2 = 252 + 1 = \mathbf{253}$$

Therefore, the spatial dimensions of the pooled feature maps in the first layer are 253 × 253.

## (1d.)

If the spatial dimensions of the convolution kernels in the second layer are 3 × 3, and no padding is used with a stride of 1, the size of the feature maps in the second layer can be calculated using equations $\boxed{1}$ and $\boxed{2}$ as follows:

$$W_2 = \frac{253 - 3 + 2(0)}{1} + 1$$
$$W_2 = 250 + 1 = \mathbf{251}$$

$$H_2 = \frac{253 - 3 + 2(0)}{1} + 1$$
$$H_2 = 250 + 1 = \mathbf{251}$$

Therefore, the spatial dimensions of the feature maps in the second layer are 251 × 251.

## (1e.)

The number of parameters in a convolution layer can be determined using

$$N_{parameters} = ((F_H \times F_W \times C_1) + 1) \times C_2$$

Where $F_H$ is the filter height, $F_W$ is the filter width, $C_1$ is the number of input image channels or feature maps, and $C_2$ is the number of filters in the layer.

Therefore, the number of parameters in the given network and a grey-scale image of dimension 32 × 32 can be determined as follows

1. Compute the number of parameters in Convolutional Layer 1

$$N_1 = ((5 \times 5 \times 1) + 1) \times 32 = \mathbf{832}$$

2. Compute the number of parameters in Convolutional Layer 2

$$N_2 = ((3 \times 3 \times 32) + 1) \times 64 = \mathbf{18496}$$

3. Compute the number of parameters in Convolutional Layer 3

$$N_3 = ((3 \times 3 \times 64) + 1) \times 128 = \mathbf{73856}$$

4. Determine the input size to the linear layers The input to the flattening layer is a $4 \times 4 \times 128$ tensor. Therefore, the output vector would be of dimension $2048 \times 1$. This will be the input dimension into the linear layer.

5. Compute the number of parameters in Linear Layer 1

$$N_4 = ((2048 + 1) \times 64 = \mathbf{131136}$$

6. Compute the number of parameters in Linear Layer 2

$$N_5 = ((64 + 1) \times 10 = \mathbf{650}$$

7. Compute the total number of parameters in the network

$$\begin{aligned} N_{total} &= (N_1 + N_2 + N_3 + N_4 + N_5) \\ &= 832 + 18496 + 73856 + 131136 + 650 \\ &= \mathbf{224970} \end{aligned}$$

Similarly, if the input image has 3 channels (i.e., RGB), only the number of parameters in the convolution layer 1 will change from 832 to 2432. As such, the total number of parameters for this new network will be $224970 + 2432 - 832 = 226570$.

Therefore, the total number of network parameters is 224970 for a grey-scale image input and 226570 for an RGB image input.

## 2 Convolutional Neural Networks Programming

### (2a.)

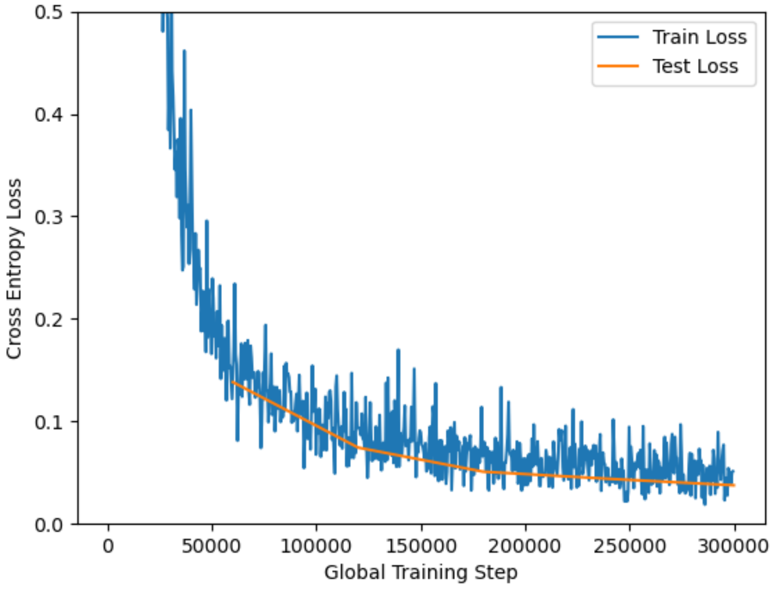


Figure 1: Train and Test Losses of the given Network

In Figure 1, there is no indication of overfitting as both the training and test losses have decreased during the course of fitting the model to the training dataset. Overfitting occurs when the training loss decreases while the test loss remains constant or increases. In this case, the model shows a balanced performance on both datasets.

## (2b.)

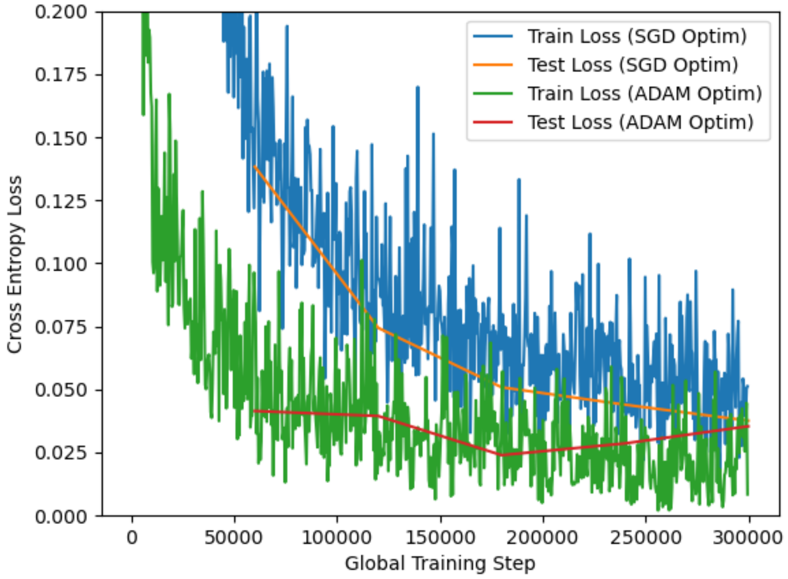


Figure 2: Train and Test Losses for both SGD and ADAM Optimizers

Figure 2 shows that the ADAM optimizer enabled a quicker convergence of the losses for this problem as compared to the SGD optimizer.

## (2c.)

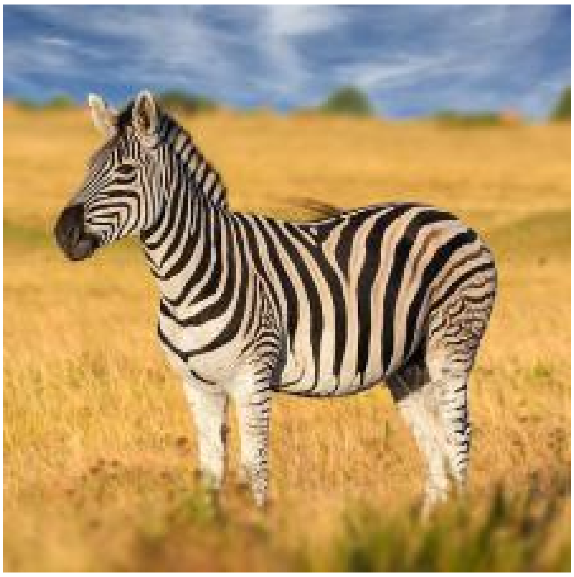


Figure 3: Given Zebra Image

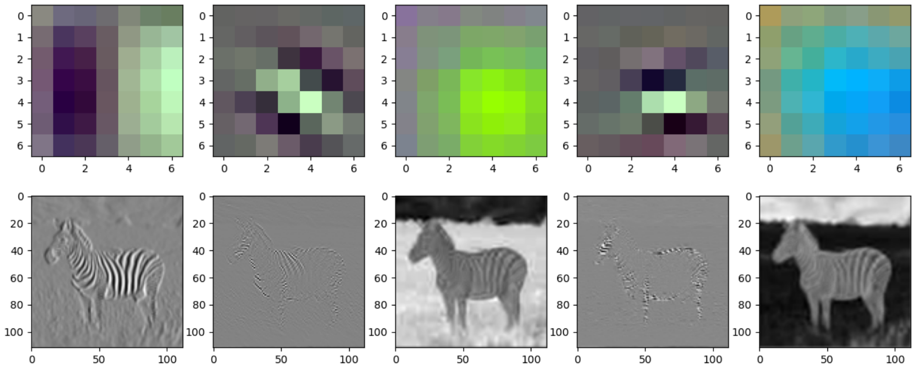


Figure 4: ResNet Filters and Activations on the Zebra Image

### (2d.)

The description of what the filters do are

1. Filter 1 detects vertical edges as its activation on the zebra image shows vertical edges such as the stripes of the zebra, whereas other edges such as the horizontal boundary between the land and sky are not visible
2. Filter 2 detects diagonal black stripes as its activation on the zebra images shows majorly the presence of stripes
3. Filter 3 segments the image and detects green objects or grass as its activation on the zebra image shows that the grass has a high probability (i.e., pixels for the activated/detected object have values close to 1 or white region) whereas the sky and zebra have relatively low probability (i.e., pixels for the undetected object have values close to 0 or black region)
4. Filter 4 detects horizontal edges as its activation on the zebra image shows horizontal edges
5. Filter 5 also segments the image but detects blue objects or the sky. Therefore its activation on the zebra image shows that the sky has a high probability (i.e., pixels for the activated/detected object have values close to 1 or white region) whereas the grass and zebra have relatively low probability (i.e., pixels for the undetected object have values close to 0 or black region)

## 3 Filtering in the Frequency Domain Theory

### (3a.)

The Fourier Transform spectrum for all images shows 2 bright spots symmetrically placed about another bright spot at the center. The bright spot at the center of these images is the origin of the frequency coordinate system. The u-axis runs left to right through the center and represents the horizontal component of frequency. The v-axis runs bottom to top through the center and represents the vertical component of frequency. Therefore, horizontal lines in the spatial domain get vertical dots in the frequency domain while vertical lines in the spatial domain get horizontal dots in the frequency domain. Also, high-frequency spatial images (closely packed lines with small spacing) have wider dots in the frequency domain, while low-frequency spatial images have tightly packed dots in the frequency domain.

Therefore, the correct mapping of these spatial and frequency images is as follows:

- 1a → 2e
- 1b → 2c
- 1c → 2f
- 1d → 2b
- 1e → 2d
- 1f → 2a

### (3b.)

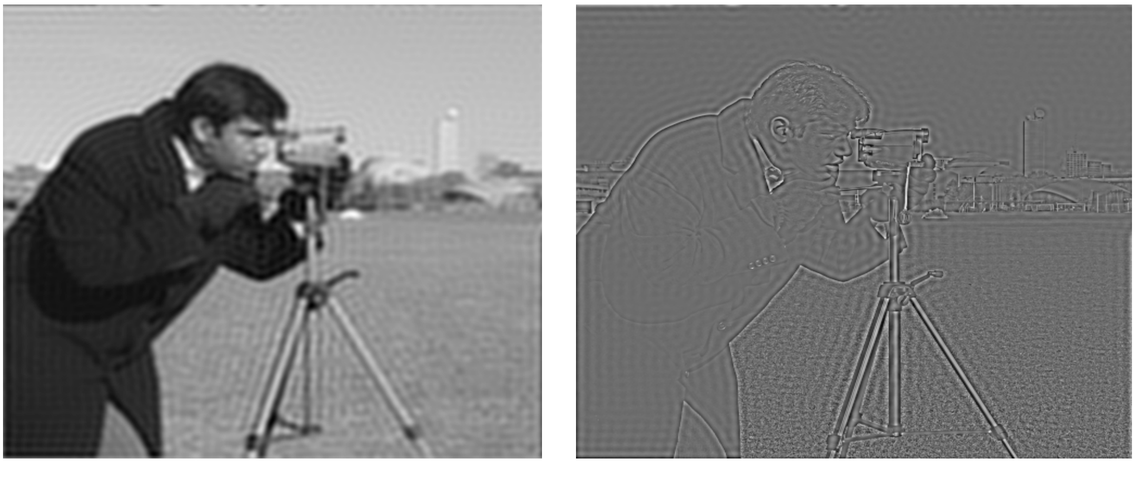
Applying a low-pass filter on an image's Fourier Transform Spectrum removes all high frequencies from the image's frequency spectrum, while a high-pass filter removes all low frequencies from the image's frequency spectrum.
Low-pass filters only preserve details that do not change rapidly. Therefore it is often used as a "blurring" or "smoothing" filter as it averages out rapid changes in intensity.
On the other hand, how-pass filters preserve details that change rapidly (e.g., edges). Therefore it is often used as an "edge detection" filter.

### (3c.)

The center of an image's frequency spectrum is the origin of the frequency coordinate system. Therefore, the distant a frequency is from the center, the higher it is. Therefore, image (a) is a high-pass filter since the filter aims to keep frequencies that are far from the center (white regions), while image (b) is a low-pass filter since the filter aims to keep only low frequencies concentrated around the center/origin (white spot at the center).

## 4 Filtering in the Frequency Domain Programming

### (4a.)



(a) Low-pass filtered image

(b) High-pass filtered image

Figure 5: Filtered Camera Man Images

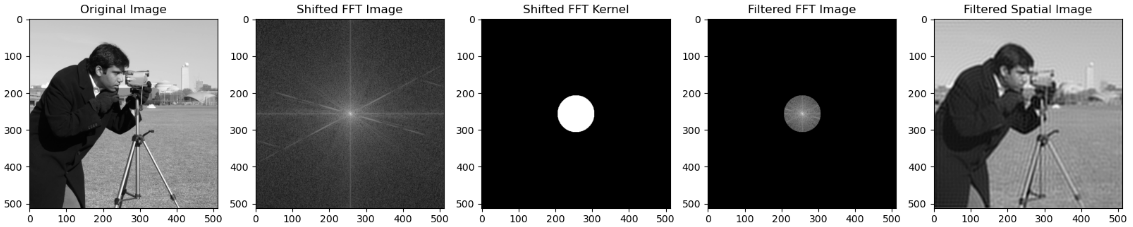


Figure 6: Low Pass Filtering of the Camera Man

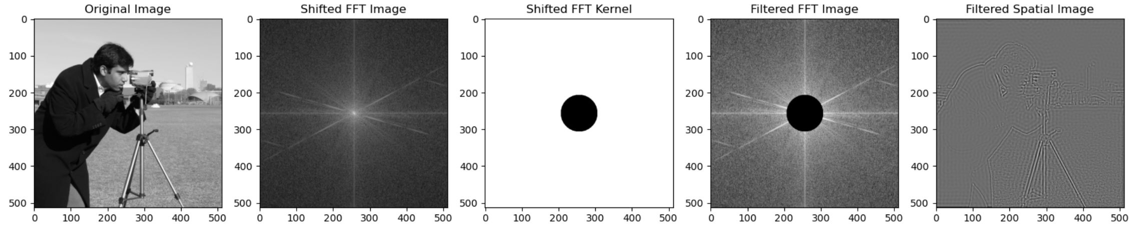
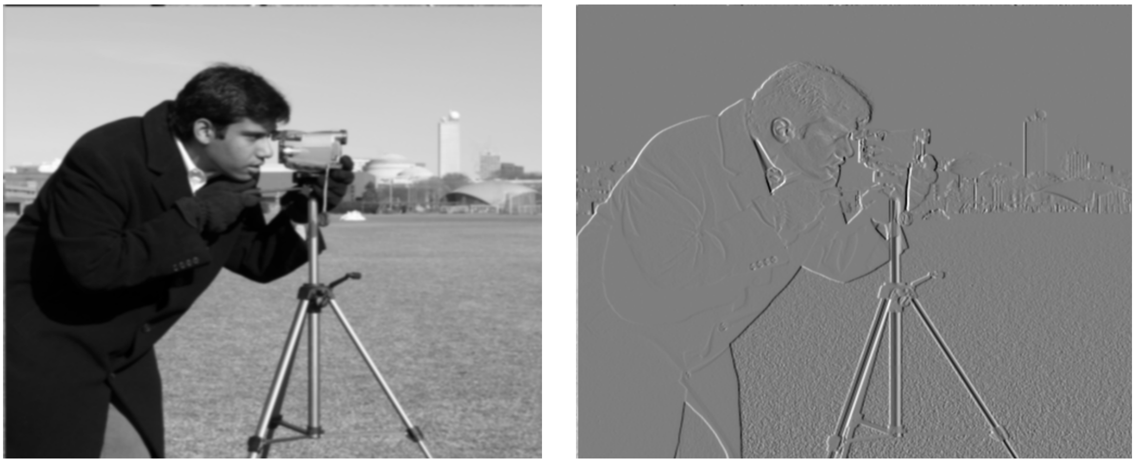


Figure 7: High Pass Filtering of the Camera Man

Figures 5 shows the filtered images in the spatial domain while Figures 6 and 7 show the transformation from the original image to the final filtered spatial image given both low-pass and high-pass filters respectively.
Consequently, the ringing effect observed in the filtered images is due to the loss of high-frequency information in the filtered image.

## (4b.)



(a) Gaussian filtered image

(b) Sobel filtered image

Figure 8: Filtered Camera Man Images

Figures 8 shows the filtered images in the spatial domain while Figures 9 and 10 show the transformation from the original image to the final filtered spatial image given both Gaussian and Sobel filters respectively.

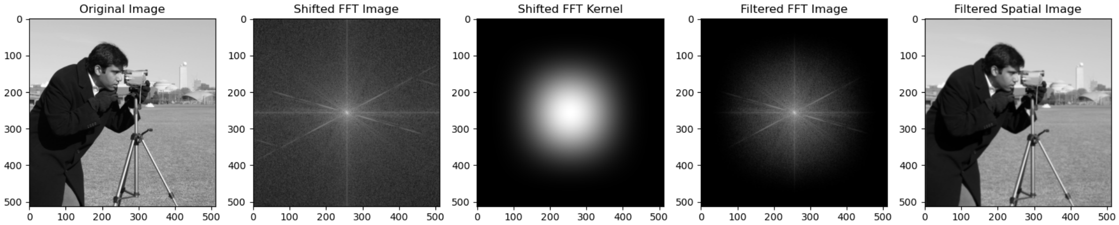


Figure 9: Gaussian Filtering of the Camera Man

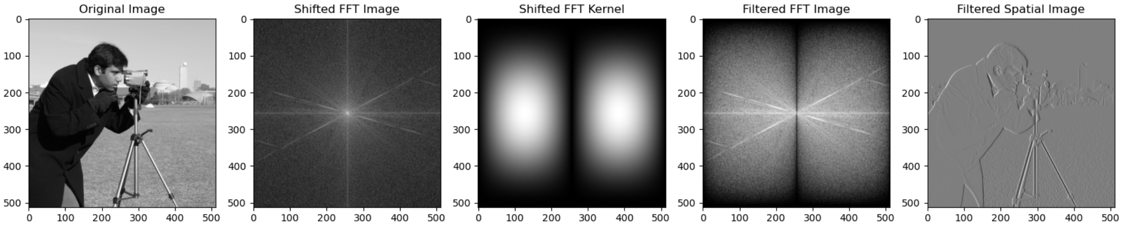


Figure 10: Sobel Filtering of the Camera Man

## (4c.)

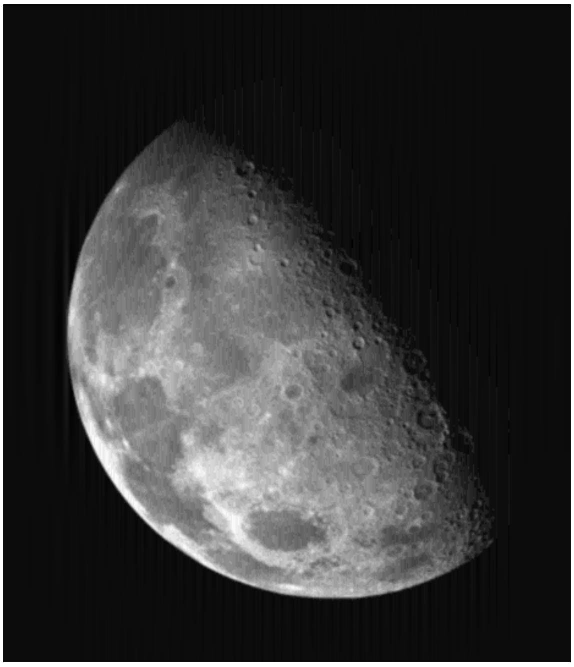


Figure 11: Filtered Spatial Moon

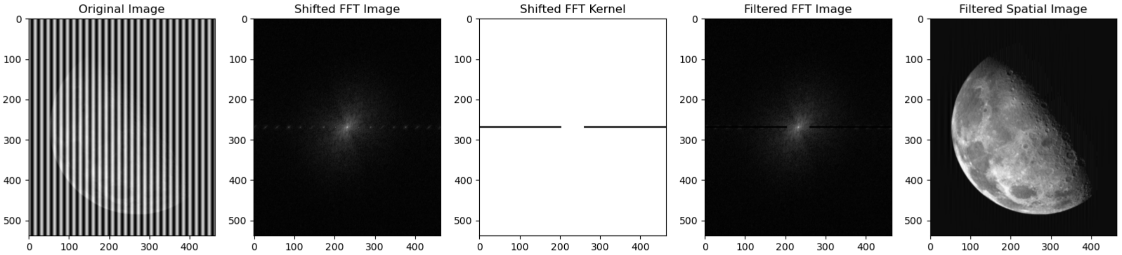


Figure 12: Filtering of the Noisy Moon

(4d.)