Assignment 2

Birk Nøhr Dissing
Jakob Damgaard Olsen

February 20, 2023

1. Pixel-wise contrast enhancement:

1.1

Figure 1 shows how changing gamma effects the image. We can see that a gamma less than 1 results in the image getting more bright, while a gamma greater than 1 results in the image being made darker. This is due to the nature of the gamma transformation, where it either compresses or stretches the dynamic range depending on the value of gamma. To do these corrections the following code was used:

```
def gammaFunc(image,gamma):
    return 255*(image/255)**gamma
```

Figure 1: Gamma correction applied to a gray scale image. Figure depicts the original image compared to the gamma < 1 and gamma > 1 corrections of 'cameraman.tif'.

1.2

The function used to apply the gamma transformation on a RGB image can be seen below:

```python
def gammaFuncRGB(image,gamma):
    img = np.copy(image)
    for i in range(image.shape[2]):
        img[:,:,i] = 255*(img[:,:,i]/255)**gamma
    return img
```

A general function was also made which can handle both gray scale images and RGB images. This function is seen below.

```python
def gammaFunc(image,gamma):
    img = np.copy(image)
    try:
        for i in range(image.shape[2]):
            img[:,:,i] = 255*(img[:,:,i]/255)**gamma
        return img
    except:
        return 255*(image/255)**gamma
```

Applying this function to the 'autumn.tif' image we get the result in figure 2.

Figure 2: Gamma correction applied to a RGB image. Figure depicts the original image compared to the gamma < 1 and gamma > 1 corrections of 'autumn.tif'.

1.3

Comparing figure 2 and 3 we see that using the gamma correction, with gamma less than 1, on the "Value" channel of the HSV image results in red channel appearing more red, than if one would apply the same gamma filter on the RGB channels. The opposite effect can also be seen for when gamma is greater than 1. Here we see that correcting the "Value" channel results in a dimmer red, while correcting the RGB values result in a more vibrant red.

Therefore if you want to use a gamma < 1 correction, the most correct one seems to be correcting the RGB values, while if you want to correct with gamma > 1, then it is more correct to transform to HSV first and then apply the correction to the 'Value' channel.

Figure 3: Gamma correction applied to the 'Value' channel of a HSV image and converted back into RGB. Figure depicts the original image compared to the gamma < 1 and gamma > 1 corrections of 'autumn.tif'.

2. Image filtering and enhancement:

2.1

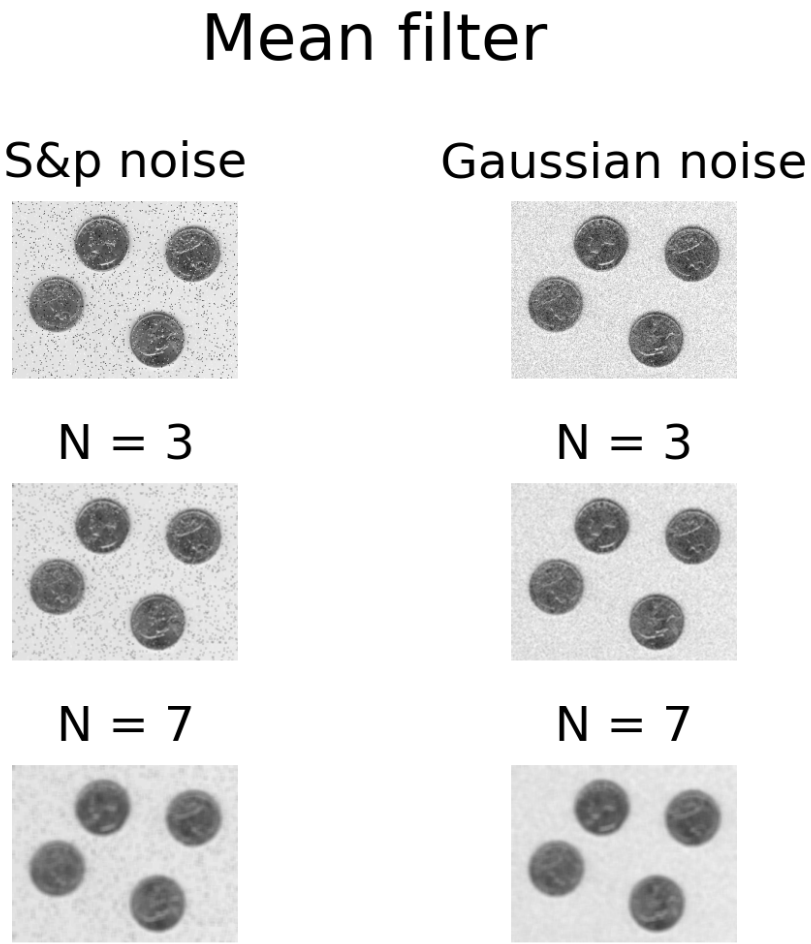

Figure 4: The mean filter was applied to the noisy images. The kernel sizes used was $N = 3$, and $N = 7$.

Salt and pepper noise and Gaussian noise was added to the picture "eight.tif". A mean and median filter with various kernel size was applied to the noisy images. The effect of the mean filter can be seen in figure 4 while the effect of the median filter can be seen in figure 5. From both of the filters it can be seen that increasing the kernel size also reduces the amount of noise in the picture. This comes at a cost of decreasing sharpness as the images gets more blurred with increasing kernel size. The mean filter performs about the same for salt and pepper, and Gaussian noise, however the median filter performs much better on salt and pepper noise than on Gaussian noise. On salt and pepper noise the median filter also performs better than the mean filter although on Gaussian noise the performance between the two filters are similar. The reason the median filter performs so well on the salt and pepper noise is that the median is immune to large outliers which the salt and pepper noise is.

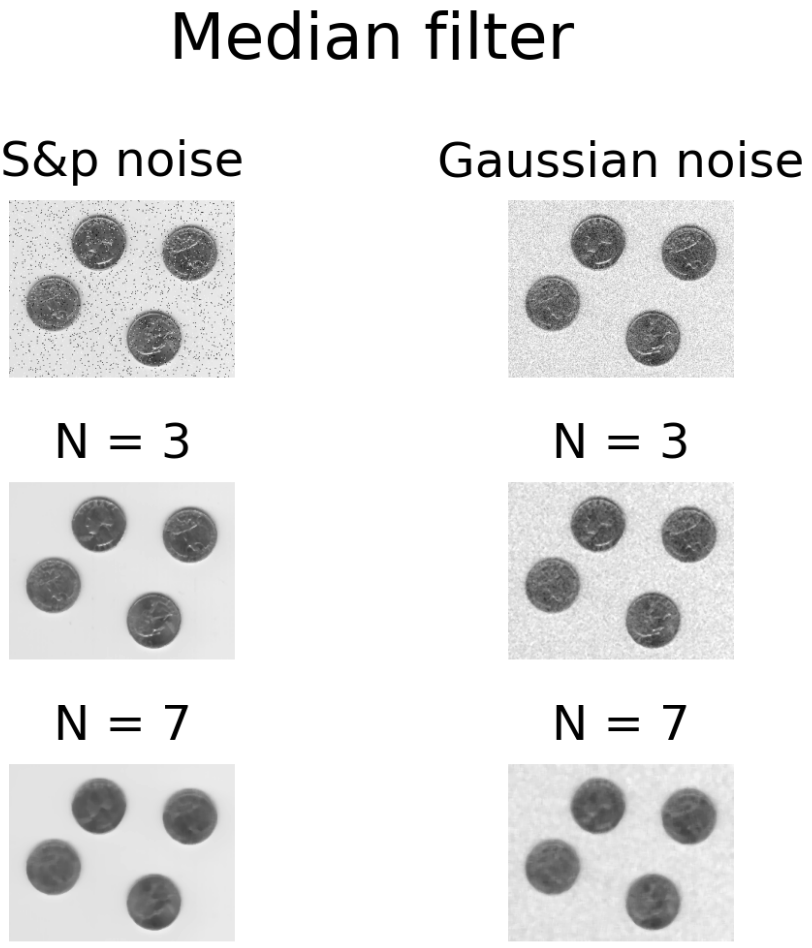

Figure 5: The median filter was applied to the noisy images. The kernel sizes used was $N = 3$, and $N = 7$.

The time it took to run 100 executions of each filter with kernel sizes 1-25 was measured and can be seen in figure 6. Here it can be seen that the execution time of the median filter scales exponentially while the execution time for the mean filter scales linearly. This is due to it being possible to linearly separate the kernel for the mean filter resulting in 2N operations instead of N^2 operations for kernels which are not linearly separable.

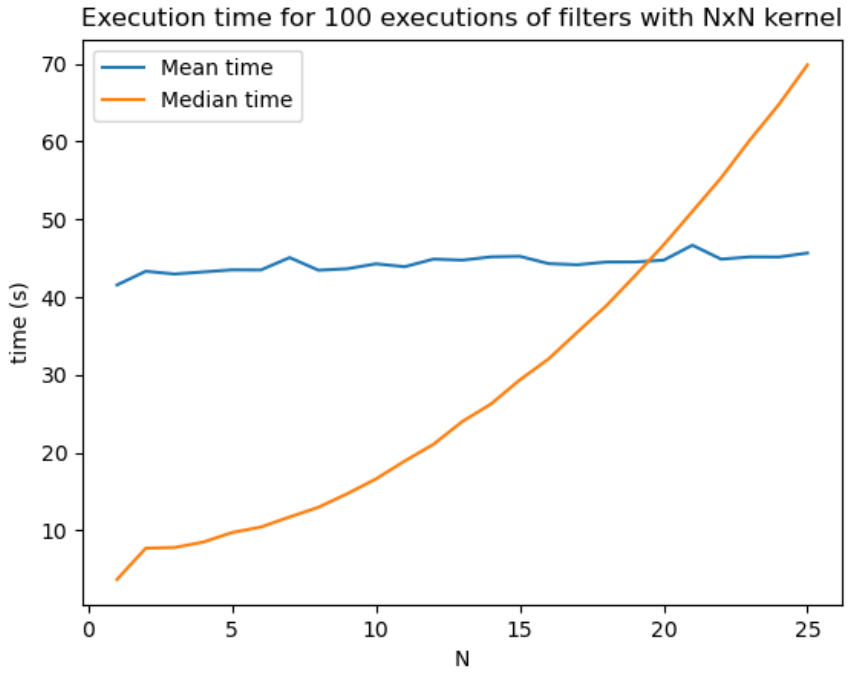

Figure 6: The time it took for 100 executions of the two filters is plotted against the size of the kernel. The median filter scales exponentially with kernel size while the mean filter scales linearly.

2.2

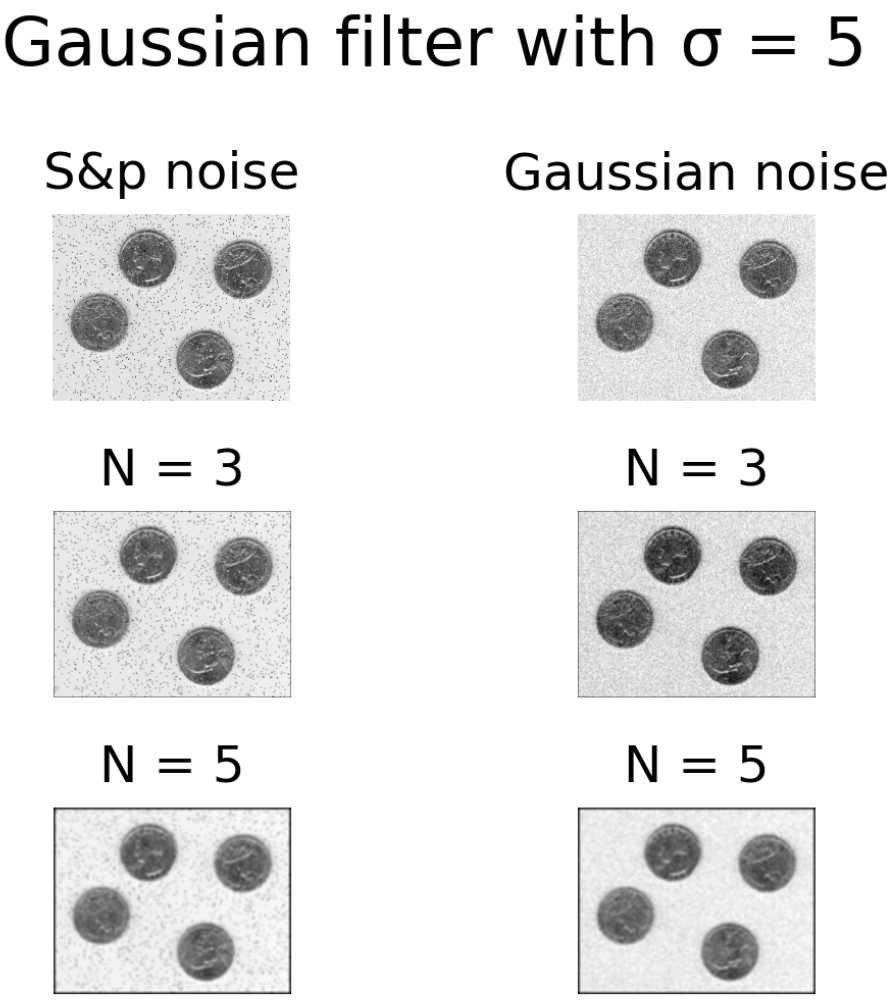

Figure 7

The effect of Gaussian filters with varying kernel size and $\sigma = 5$ can be seen in figure 7. When the kernel size, N, increases a black border appears on the image. This is due to the image not being padded, and the intensity of the pixels where the kernel can not be applied is set to 0.

2.3

The effect of Gaussian filters for increasing values of σ while the filter kernel size scales as $N = 3\sigma$ can be seen in figure 8. We see that as σ is increased, the blurring effect that the Gaussian filter has indeed does partially remove noise both the SP noise and Gaussian noise. But comparing the effect it has on the two types of noise we see that it is better at handling Gaussian noise compared to SP noise. We also see that for $\sigma = 1$ the sharpness of the image is still at a point where we are easily able to see depth changes in the coins, but for $\sigma = 3$ some of this depth/edges is lost due to the filtering blurring it out. So from this we can see that generally as σ is increased we reduce the noise in the image while also blurring it more. Therefore one needs to consider the value of σ when applying a Gaussian filter to not blur the image to much, while still removing the unwanted noise.

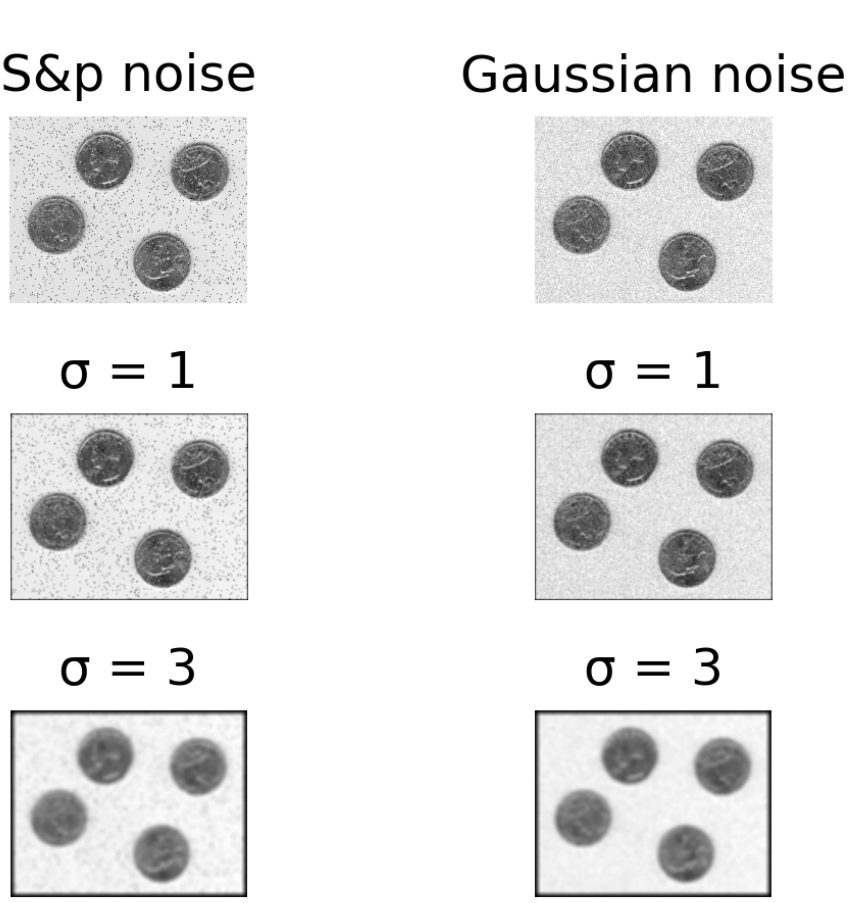

Figure 8

3. Complex numbers:

3.1

$$\text{i.} \quad d = a + b = (a_1 + b_1) + (a_2 + b_2)i = d_1 + d_2 i \tag{1}$$
$$\textbf{with} \quad d_1 = (a_1 + b_1), d_2 = (a_2 + b_2)$$

$$\text{ii.} \quad d = a - b = (a_1 - b_1) + (a_2 - b_2)i = d_1 + d_2 i \tag{2}$$
$$\textbf{with} \quad d_1 = (a_1 - b_1), d_2 = (a_2 - b_2)$$

$$\text{iii.} \quad d = a \cdot b = a_1 \cdot (b_1 + b_2 i) + a_2 i \cdot (b_1 + b_2 i) = a_1 b_1 + a_1 b_2 i + a_2 b_1 i + a_2 b_2 i^2 = \tag{3}$$
$$(a_1 b_1 - a_2 b_2) + (a_1 b_2 + a_2 b_1)i = d_1 + d_2 i$$
$$\textbf{with} \quad d_1 = (a_1 b_1 - a_2 b_2), d_2 = (a_1 b_2 + a_2 b_1)$$

$$\text{iv.} \quad d = \frac{a}{b} = \frac{a_1 + a_2 i}{b_1 + b_2 i} \cdot \frac{b_1 - b_2 i}{b_1 - b_2 i} = \frac{(a_1 b_1 + a_2 b_2) + (a_1 b_2 + a_2 b_1) i}{b_1^2 + b_2^2} = \tag{4}$$
$$\frac{(a_1 b_1 + a_2 b_2)}{b_1^2 + b_2^2} + \frac{(a_1 b_2 + a_2 b_1) i}{b_1^2 + b_2^2} = d_1 + d_2 i$$
$$\textbf{with} \quad d_1 = \frac{(a_1 b_1 + a_2 b_2)}{b_1^2 + b_2^2}, d_2 = \frac{(a_1 b_2 + a_2 b_1)}{b_1^2 + b_2^2}$$

3.2

$$d = \sqrt{-3} = \sqrt{3 \cdot (-1)} = \sqrt{3} \cdot i = d_1 + d_2 i \tag{5}$$
$$\textbf{with} \quad d_1 = 0, d_2 = \sqrt{3}$$

3.3

$$\text{i.} \quad d = a \cdot b = a_r e^{i a_\theta} \cdot b_r e^{i b_\theta} = a_r b_r e^{i a_\theta + i b_\theta} = a_r b_r e^{i \cdot (a_\theta + b_\theta)} = d_r e^{i d_\theta} \tag{6}$$
$$\textbf{with} \quad d_r = (a_r b_r), d_\theta = (a_\theta + b_\theta))$$

$$\text{ii.} \quad d = \frac{a}{b} = \frac{a_r e^{i a_\theta}}{b_r e^{i b_\theta}} = \frac{a_r}{b_r} e^{i a_\theta - i b_\theta} = \frac{a_r}{b_r} e^{i \cdot (a_\theta - b_\theta)} = d_r e^{i d_\theta} \tag{7}$$
$$\textbf{with} \quad d_r = \left(\frac{a_r}{b_r} \right), d_\theta = (a_\theta - b_\theta))$$

3.4

$$(a_r e^{i a_\theta})^* = a_r e^{i(-a_\theta)} \tag{8}$$

3.5

$$\text{i.} \quad d = a + b = a_r e^{i a_\theta} + b_r e^{i b_\theta} = a_r(\cos(a_\theta) + i \sin(a_\theta)) + b_r(\cos(b_\theta) + i \sin(b_\theta)) = \tag{9}$$
$$(a_r \cos(a_\theta) + b_r \cos(b_\theta)) + i(a_r \sin(a_\theta) + b_r \sin(b_\theta)) = d_1 + i d_2$$
$$\textbf{with} \quad d_1 = (a_r \cos(a_\theta) + b_r \cos(b_\theta)), d_2 = (a_r \sin(a_\theta) + b_r \sin(b_\theta))$$

$$\text{i.} \quad d = a - b = a_r e^{i a_\theta} - b_r e^{i b_\theta} = a_r(\cos(a_\theta) + i \sin(a_\theta)) - b_r(\cos(b_\theta) - i \sin(b_\theta)) = \tag{10}$$
$$(a_r \cos(a_\theta) - b_r \cos(b_\theta)) + i(a_r \sin(a_\theta) - b_r \sin(b_\theta)) = d_1 + i d_2$$
$$\textbf{with} \quad d_1 = (a_r \cos(a_\theta) - b_r \cos(b_\theta)), d_2 = (a_r \sin(a_\theta) - b_r \sin(b_\theta))$$

3.6

Finding d_r and d_θ.

$$d_r = \sqrt{a_1^2 + a_2^2}, \quad d_\theta = \arctan\left(\frac{a_2}{a_1} \right) \tag{11}$$

Writing the complex number in polar form.

$$d = \sqrt{a_1^2 + a_2^2} e^{i \arctan\left(\frac{a_2}{a_1}\right)} = d_r e^{i d_\theta} \tag{12}$$

4. Fourier Transform - Practice:

Using scipy.fft.fft2 as fft2 and scipy.fft.fftshift as fftshift, a function was created that calculates the power spectrum of the image :

```
def powerSpec(image):
    ft = fft2(np.copy(image))
    ftshift = fftshift(ft)
    ps = ftshift**2
    return ps
```

Using this function on the image "trui.png" we get the result displayed in figure 9. From the power spectrum we can see multiple features. One important on is the "cross" that goes through the middle of the image. This feature indicates the orientation of the image, We can also see this by calculating the power spectrum of the original image after applying a rotation to it. This can be seen in figure 10 where we indeed see that this cross shows the orientation of the image.

Another feature that can be seen is halfway out into the corner from the middle, here we see areas which are lighter, indicating more signals at that frequency. The frequency at these points are from mid to high frequency signals. These areas might therefore be a representation of the color changes or edges of the objects in the image.

The power spectrum and fft also both show us this property of their symmetry due to the fft having an imaginary component.

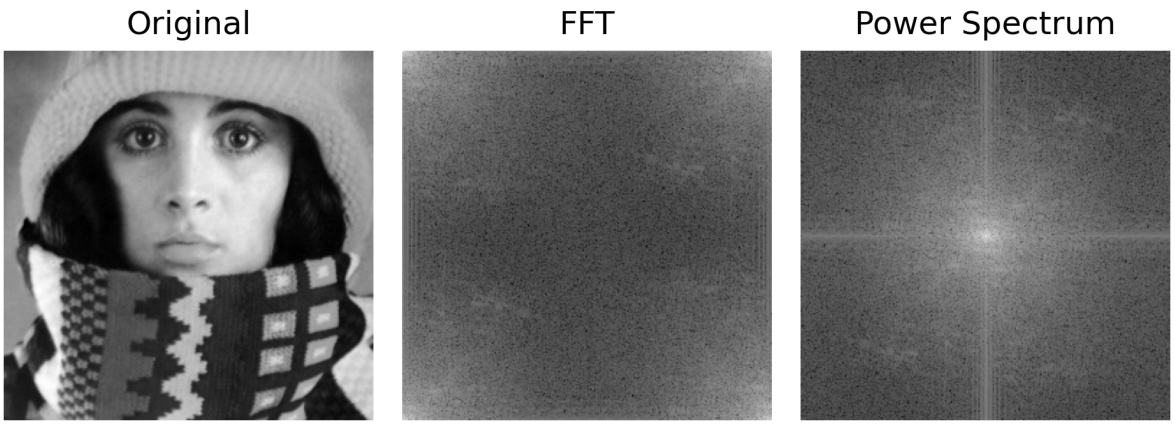

Figure 9: Power spectrum of the image "trui.png". Left figure shows the original image while the center and right figure shows FFT and power spectrum plotted as $log_{10}(1 + abs(x))$, where x is either the FFT or power spectrum.

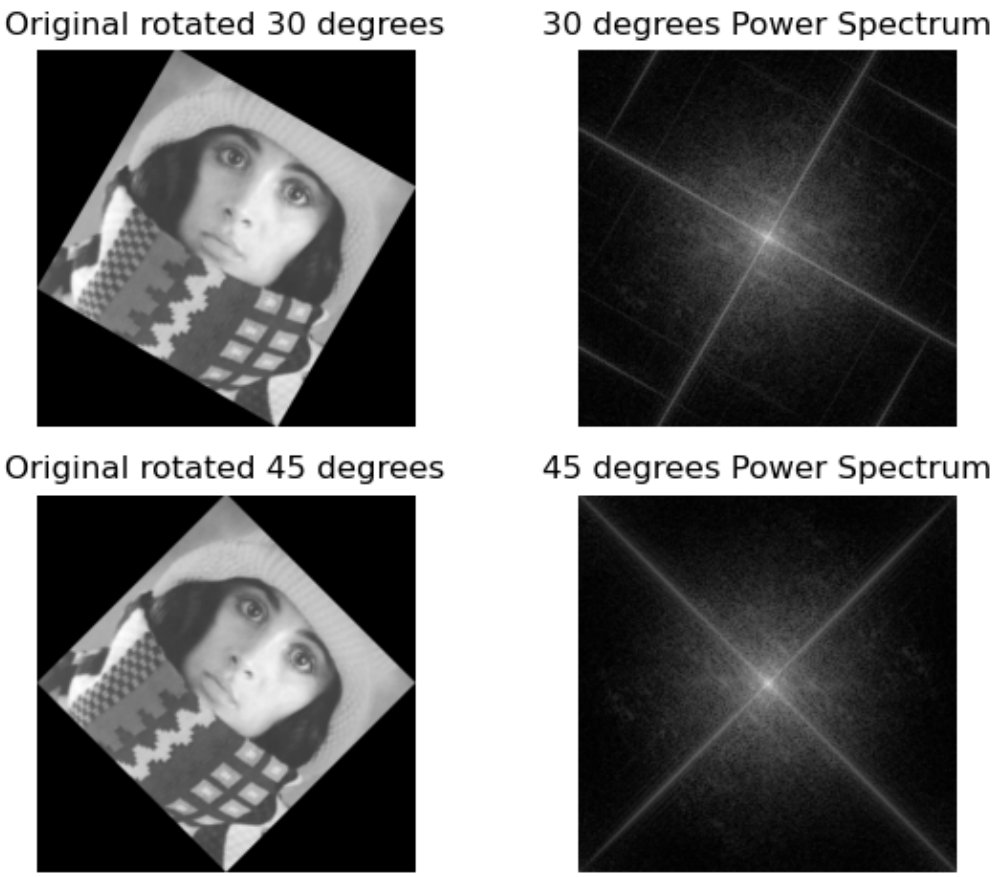

Figure 10: Power spectrum's of the rotated image "trui.png". Left figure shows the altered image, and the right figure shows corresponding the power spectrum.