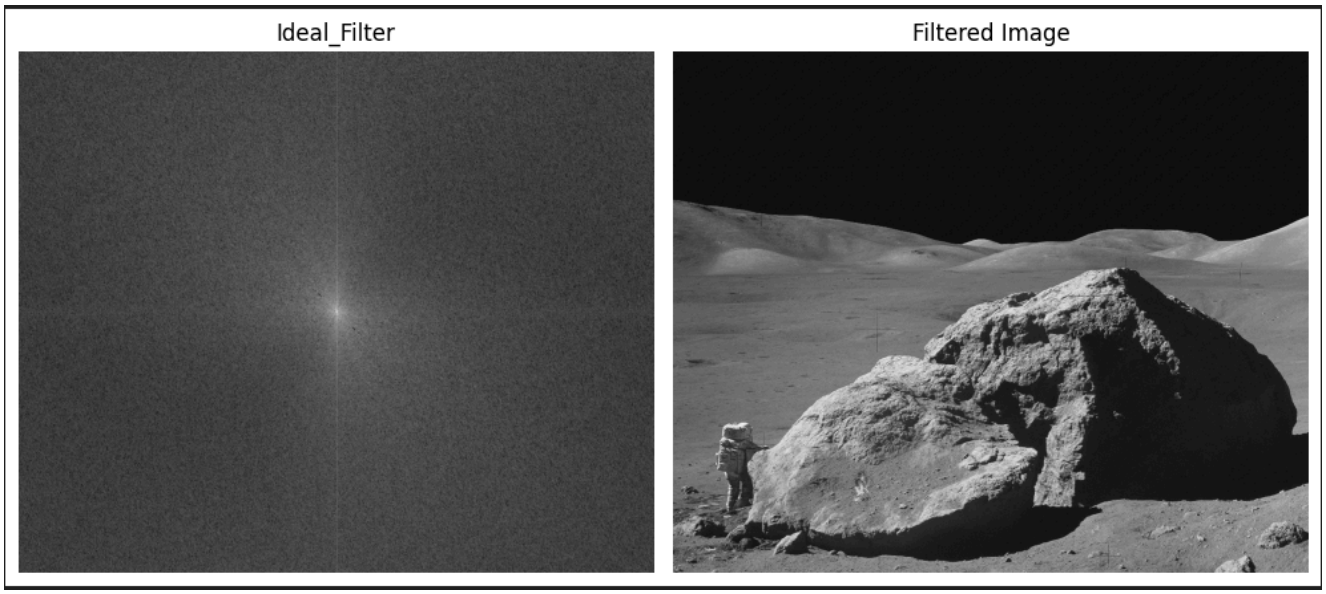
DIP Homework 2

Jayden Fassett

This assignment was done in a jupyter notebook. All code is in the attached jupyter file and is commented for convenience.

Question 1

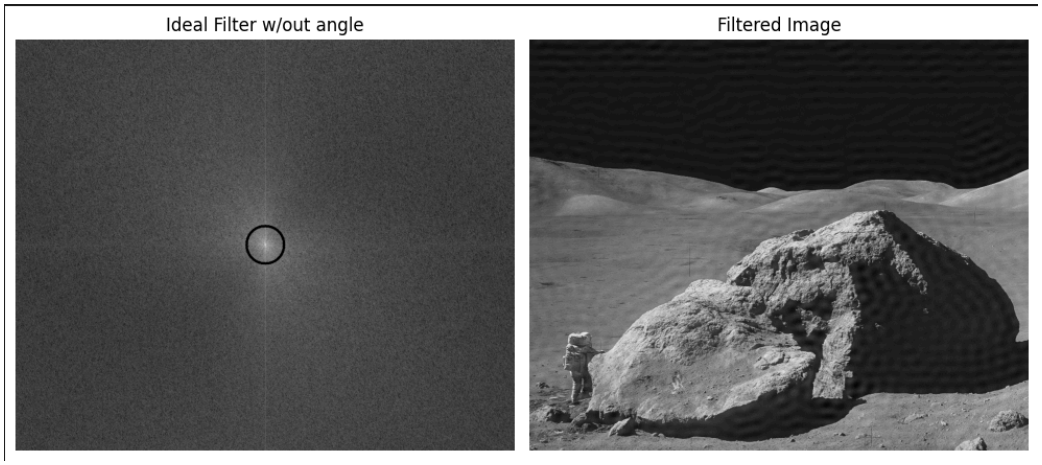


I cleaned the image using a custom-built ideal filter. It was initially designed as a standard band-reject filter, but I noticed that the noise was on a -45 degree line, so I added that component in. I was able to get a very clean image.

This is the code used to construct the filter. It builds two circles on a matrix of ones, and gets the difference. Every value that falls into the difference is zeroed out, giving a band. Then I add another if statement to determine if they are at a -45 degree angle.

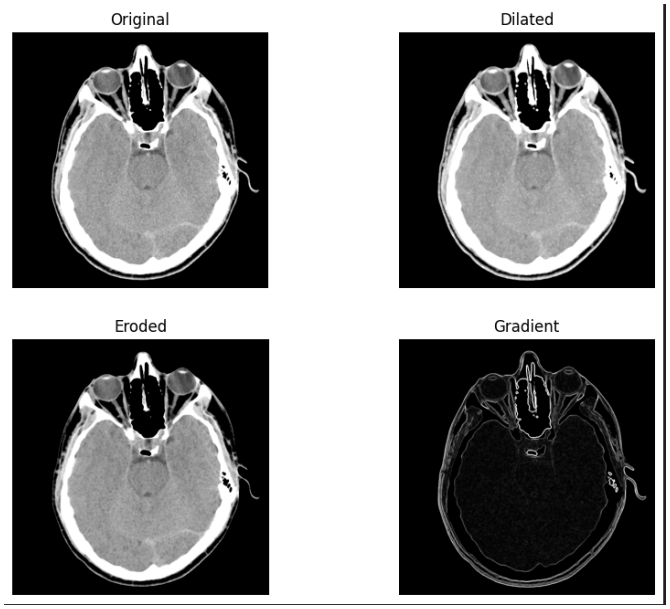
```
def ideal(img, bigrad=40, smallrad=35):
    h, w = img.shape
    #Array of ones the same size as the image. The idea is that you zero out only the necessary values
    radiusfilter = np.ones((h, w))
    center = (h / 2, w / 2)

    for idx in range(h):
        for idx2 in range(w):
            # If within radius and not on the -45 degree line
            distance = dist((idx, idx2), center)
            if distance >= smallrad and distance <= bigrad:
                # Check if the point is on the -45 degree line
                # if dist((idx,idx2),center) > smallrad:
                if idx - idx2 == center[0] - center[1]:
                    radiusfilter[idx, idx2] = 0
    return radiusfilter
```

This is what it looked like without the angle component for reference:



Question 2

Question 2 was quite simple in comparison. I was able to recreate the image using basic scipy functions.




```
import skimage.morphology as mp

img2_dilated = mp.dilation(img2)
img2_eroded = mp.erosion(img2)
images = [img2,img2_dilated,img2_eroded,img2_dilated - img2_eroded]
labels = ["Original","Dilated","Eroded","Gradient"]
fig, axes = plt.subplots(2, 2, figsize=(10, 8))
for i, ax in enumerate(axes.flatten()):
    ax.imshow(images[i], cmap='gray')
    ax.axis('off')

    if labels is not None:
        ax.set_title(labels[i])
```

✓ 0.3s