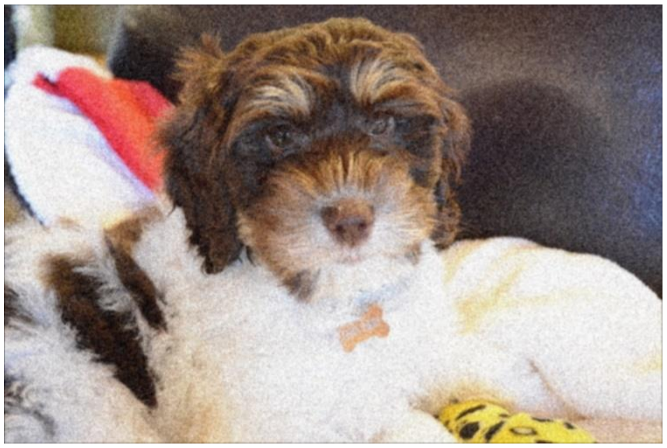
Kewen Peng

CSC 391 Project 1

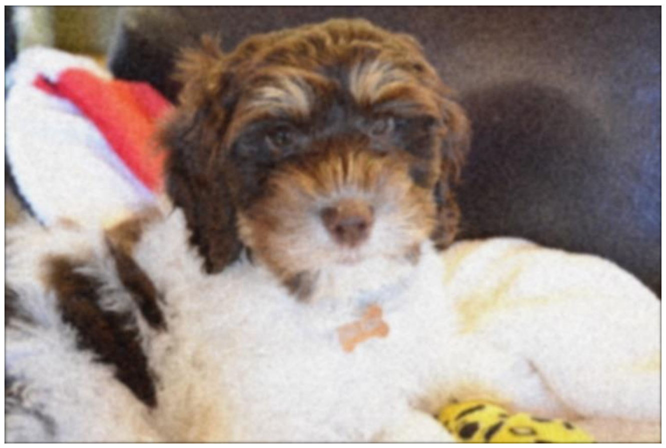
In this project, we use OpenCV to conduct experiments on image processing. The OpenCV package in Python allows us to apply spatial and frequency filters on an image for various goals (i.e. denoising, smoothing, edge detection). We also compared the difference between the ideal low/high pass filter and Butterworth low/high pass filter.

3.2.1 Spatial filter

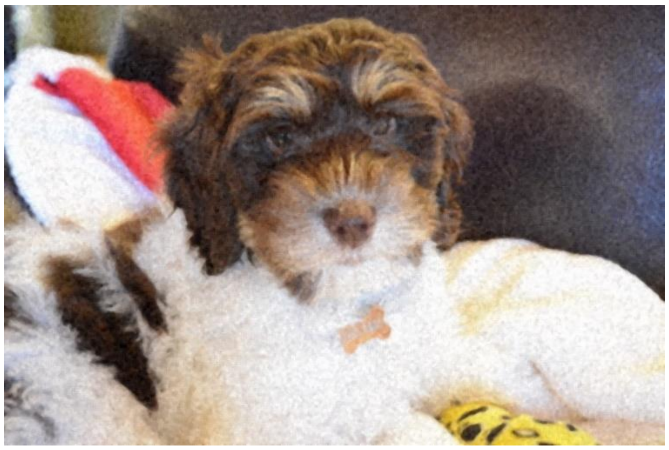
Gaussian 5x5



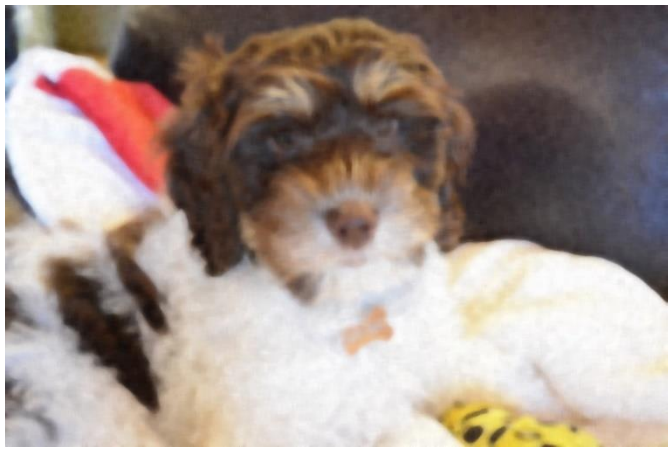
Gaussian 11x11



Median 5x5



Median 11x11

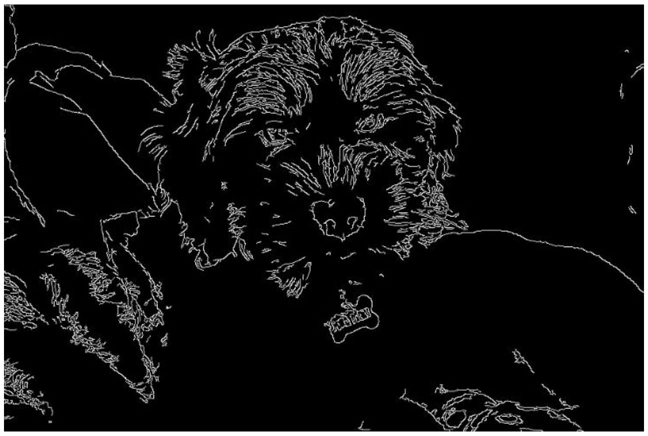


In both filters, with better denoising performance, the resolution gets lower. This is because with a filter of larger size, the filter will use more neighbors to impute the pixel. This leads to a better denoising result while also blurring edges.

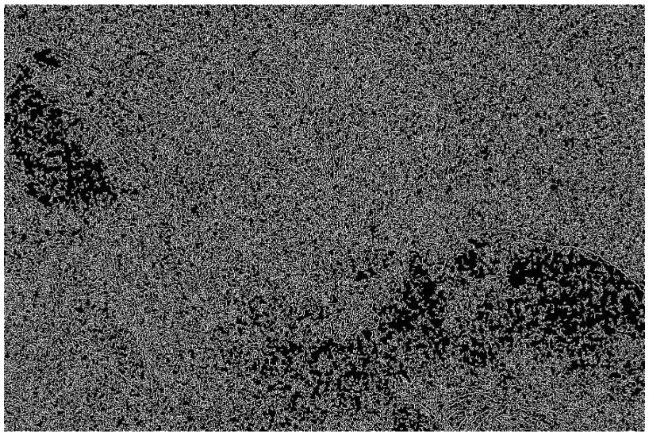
Kewen Peng

3.2.2 Canny edge detection

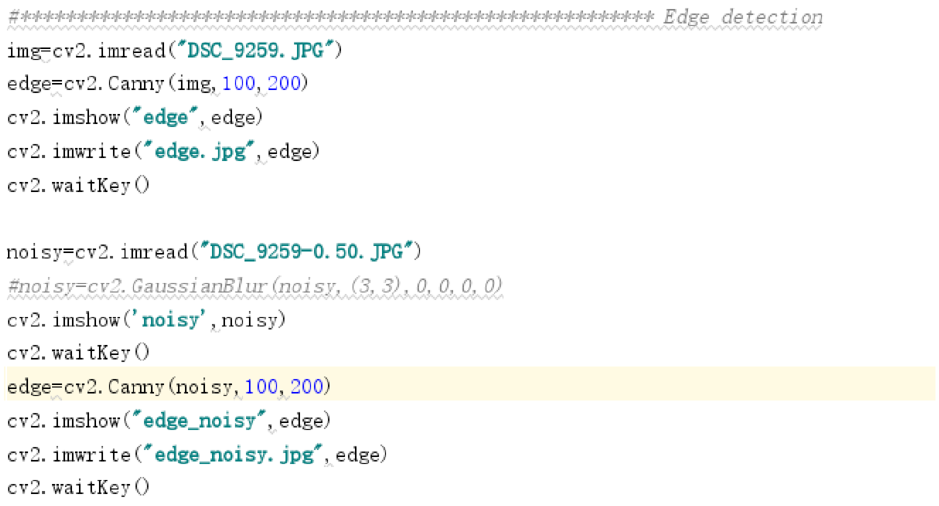
Original



Noisy

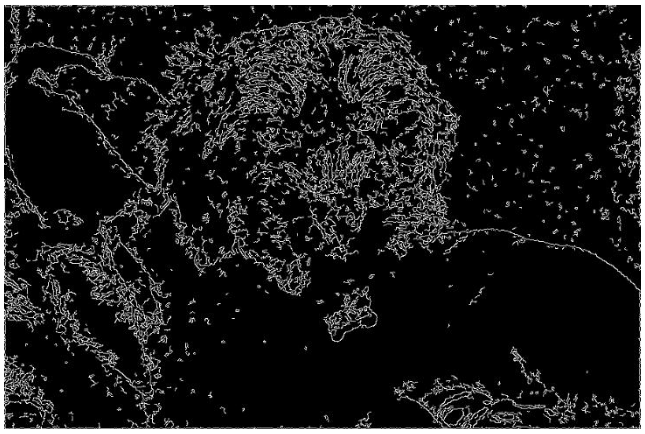


With Canny edge detection of the same parameter, it is harder to detect desired edges from the noisy image. Codes shown as below:



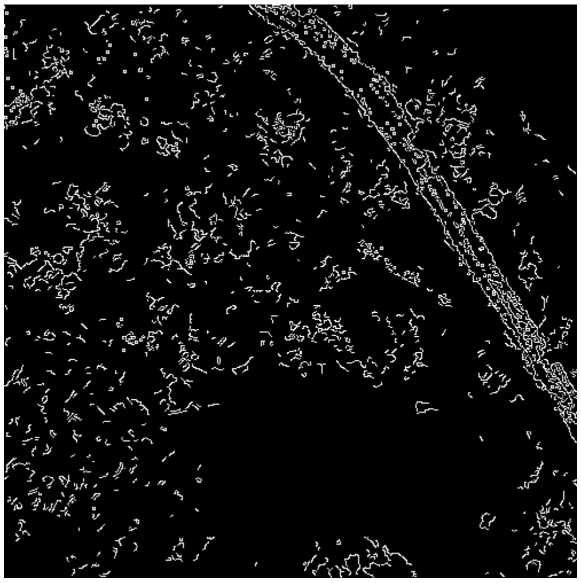
```
#****************************************************************** Edge detection
img=cv2.imread("DSC_9259.JPG")
edge=cv2.Canny(img,100,200)
cv2.imshow("edge",edge)
cv2.imwrite("edge.jpg",edge)
cv2.waitKey()

noisy=cv2.imread("DSC_9259-0.50.JPG")
#noisy=cv2.GaussianBlur(noisy,(3,3),0,0,0,0)
cv2.imshow('noisy',noisy)
cv2.waitKey()
edge=cv2.Canny(noisy,100,200)
cv2.imshow("edge_noisy",edge)
cv2.imwrite("edge_noisy.jpg",edge)
cv2.waitKey()
```

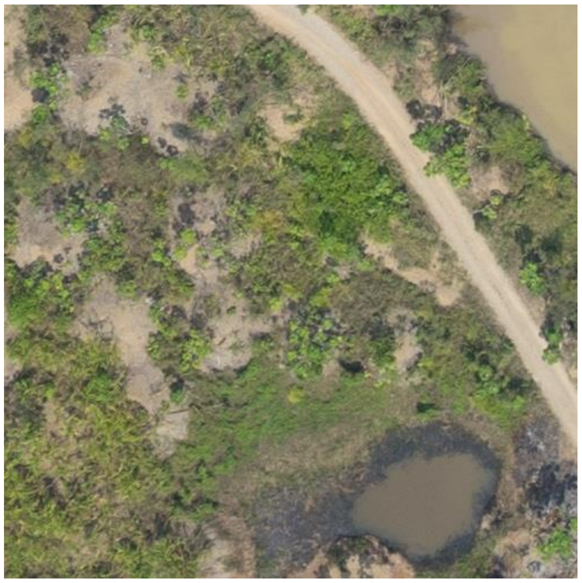
But if we apply GaussianBlur filter to the noisy image first, the result gets better.



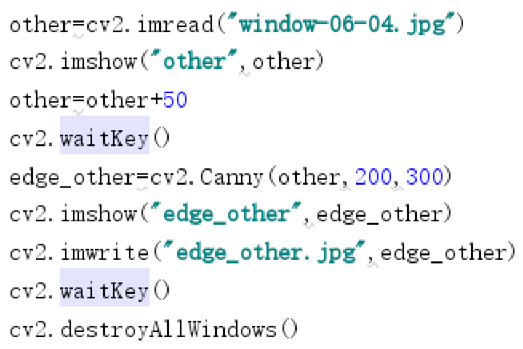
Kewen Peng

Field image





In order to better extract edges from the noisy field image, I first apply Gaussian filter to the noisy image and then try the edge detection.

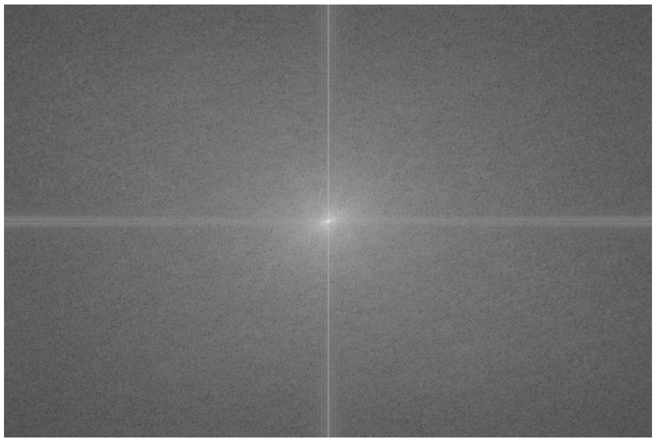


```
other=cv2.imread("window-06-04.jpg")
cv2.imshow("other",other)
other=other+50
cv2.waitKey()
edge_other=cv2.Canny(other,200,300)
cv2.imshow("edge_other",edge_other)
cv2.imwrite("edge_other.jpg",edge_other)
cv2.waitKey()
cv2.destroyAllWindows()
```

Kewen Peng

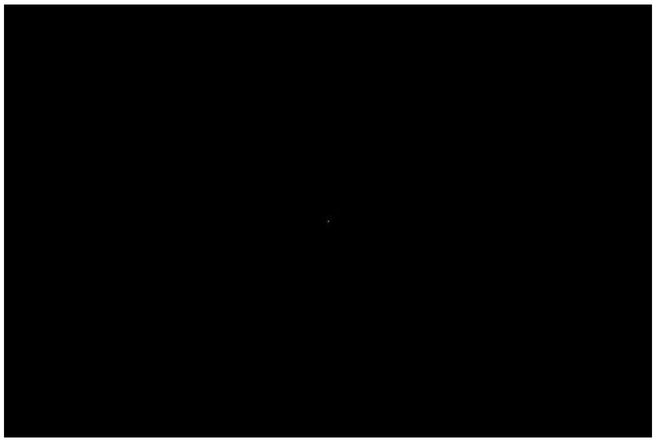
4.1 2-D DFT

Log(Magnitude+1) of DFT



Original gray image

Magnitude



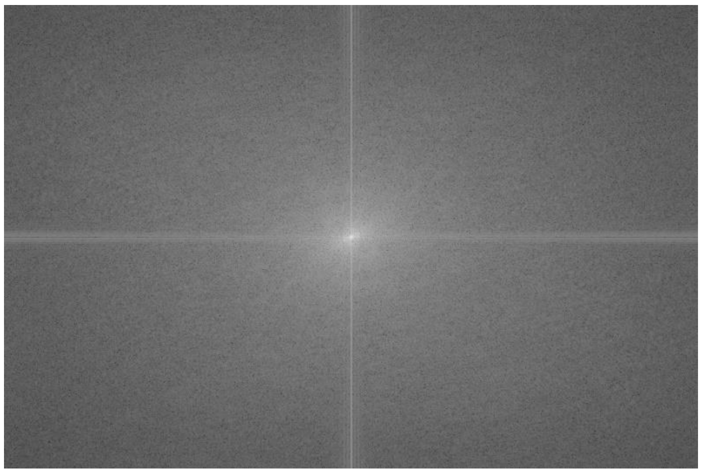
Kewen Peng

4.2 Frequency analysis

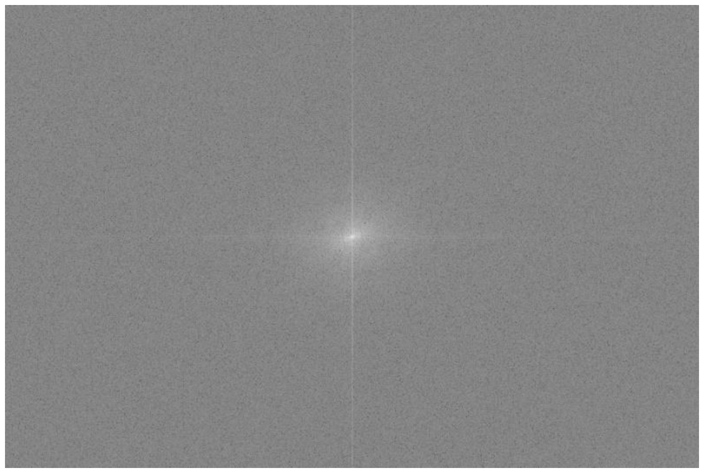
a) When the image is smoother (less artificial content), the coefficient decays faster, because smoother images tend to have less high-frequency “ingredients.

b) The noisy one gives the Fourier coefficients of higher frequency a greater magnitude compared to the original image.

DFT of original puppy image



DFT of 0.5-noisy puppy image



c) Because the image has strong edges along that direction, which is, in this case, along the axis.

d) Removing coefficient far away from the center will make the image blurred. (Like an ideal low pass filter)

Removing coefficients near the center only leaves high frequencies, which is some specific edges. (Like an ideal high pass filter)

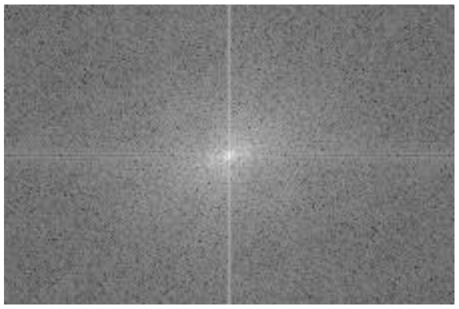
When smoothing an image, we can use low pass filter, and when we want edge detection, we can apply the high pass filter.

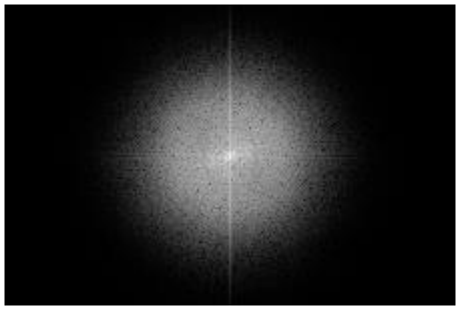
Kewen Peng

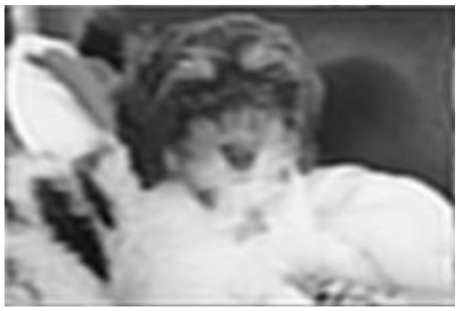
5.1 Butterworth low pass filter

Original LogMagnitude

Butterworth low pass LogMagnitude



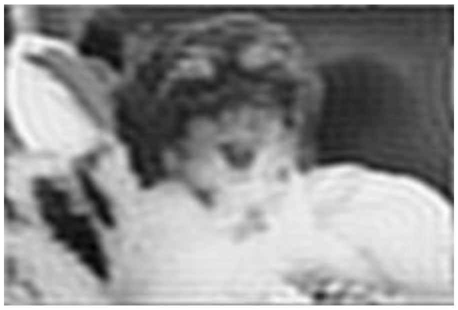




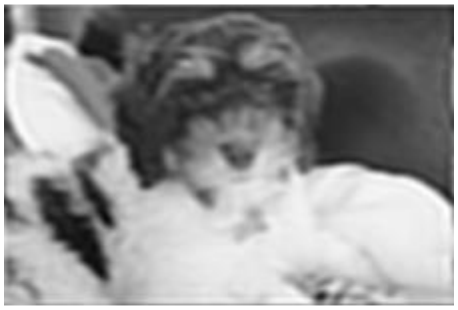
Kewen Peng

5.2 Compare ideal and Butterworth filter

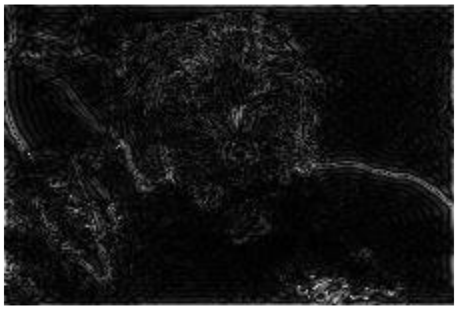
Ideal low pass



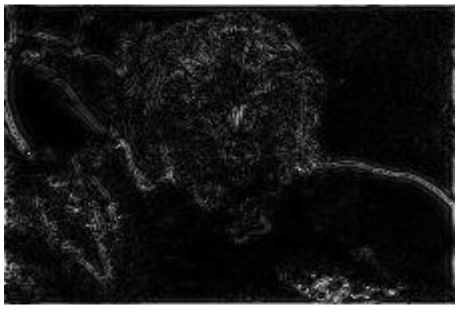
Butter low pass



Ideal high pass



Butter high pass



By my observation, the ideal filters contains more noise in the image than Butterworth filters do. This is because Butterworth low/high pass filter removes frequencies out of the threshold in a smoother way, while the ideal low/high pass filter obtains a very sharp decay around the cutoff frequency.