# *IAgnel Charities*

# Fr. C. Rodrigues Institute of Technology, Vashi

**Department of Electronics and Telecommunication Engineering**

**SUB: Image Processing & Machine Vision Lab (ECL603)**

**FH 2024**

Dear Student,

This lab is about learning how to perform frequency domain processing on a given digital image. The objectives of this lab are:

1. To visualize image in frequency domain and relate frequency with intensity change
2. To create a Gaussian LPF by using Euclidian distance concept.
3. To perform filtering in frequency domain using the designed filter.
4. To visualize filtered image in spatial domain.
5. To relate the concept of cut-off frequency with visual appeal of filtered image.
6. To be able to modify the code for a given filtering operation.

**Task 1. Read the following paragraph about frequency domain processing and get your doubts clarified from the instructor.**

Basically, here, image is looked upon as a matrix in two-dimensions and a two-dimensional transform is applied to convert it into an object in frequency domain. You must be wondering what is “frequency” in an image. So, the term frequency is associated with changes in intensity in an image. If an image has too many edges in it, we say that the image has high frequency content. In contrast, if an image is very smooth, the gray levels changing gradually, then we say that the image has low frequency content. Now, what is the utility of looking at images from the frequency domain? So, if you remember from the knowledge acquired about noise in previous semesters, any type of noise is generally a high frequency signal. It is true for noises in image too. So, we would “filter out” noise by designing a ……. …pass filter (fill in the blank with Low/ High) for the image. Does that make sense?

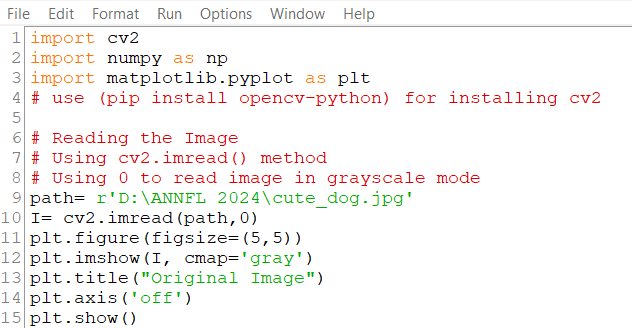
How do we design a filter for images? So, there are ways to design a high-pass, low-pass or band-pass filter for filtering images in frequency domain. The procedure comprises of the following steps.

1. Convert image in frequency domain (which is called FFT of the image)
2. Perform center-shifting of the FFT in order to bring low frequencies to central area of the FFT image.
3. Create desired filter (a black-and-white image) according to the function required.
4. Multiply the FFT in step ii with the filter created in step iii.
5. Take inverse FFT of the resulting image

Though the steps look very simple, there is much to each step. Let us first study what do you mean by FFT of an image.

**Task 2. Reading an image and grasping the concept of “image in frequency domain” by performing FFT on it.**

Study the following piece of code, and execute it in your IDE terminal with correct syntax. Note that Python is indent sensitive, so one extra space and tour output is gone for a toss. Note the command provided in the comment on line 4 to install the cv2 package, which is necessary for image processing in python. Remember to give YOUR image path on line 9. Note the ‘0’ which is required for grayscale display, on line 10. Lines 11 to 15 are mandatory lines for displaying any figure with proper title, in proper screen-size.

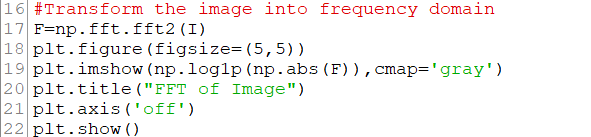


Code piece 1

1. Read the image ‘cute dog.jpg’ from the relevant folder, and display this original image. (code piece 1) Paste your output here.

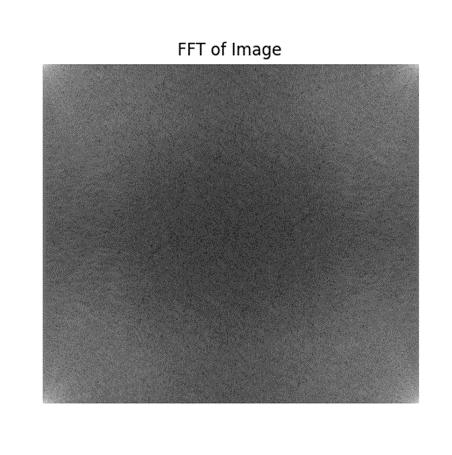


1. Convert the image into frequency domain as shown in code piece 2.

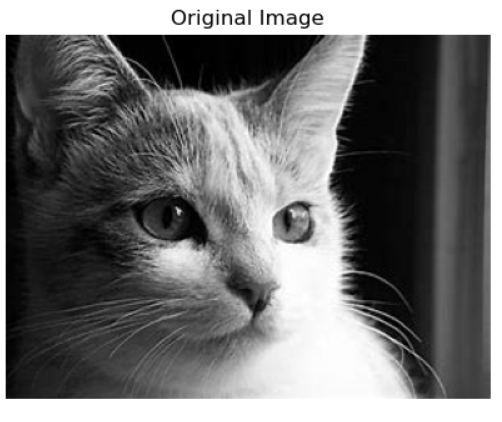


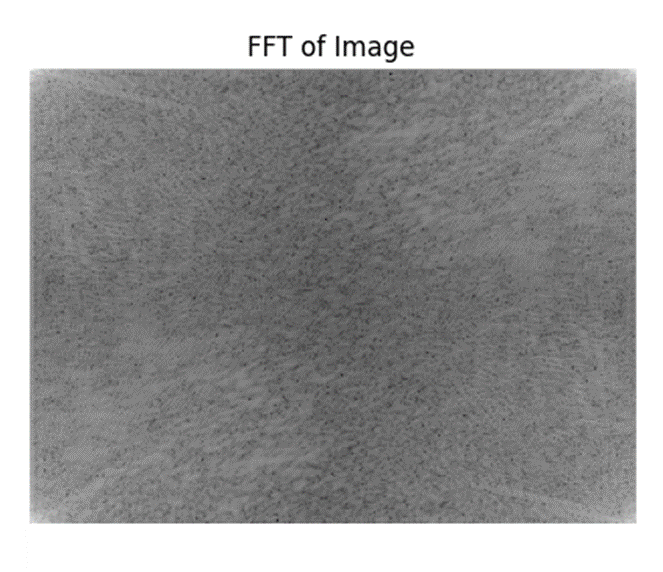
Code piece 2

1. Paste your output here.

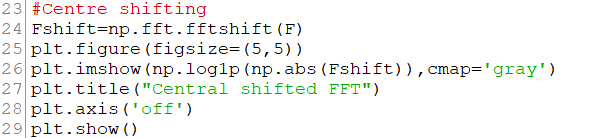


1. Note that line 17 is the use of the fft2 function in numpy.
2. Write your observations about the white color content in the FFT image. Is there too much white content in corners? Is the centre black?
3. Change the image to “cute\_cat” (uploaded in your folder, shown below). Paste the image of FFT of cute\_cat here. Now state what is your observation about the FFT.





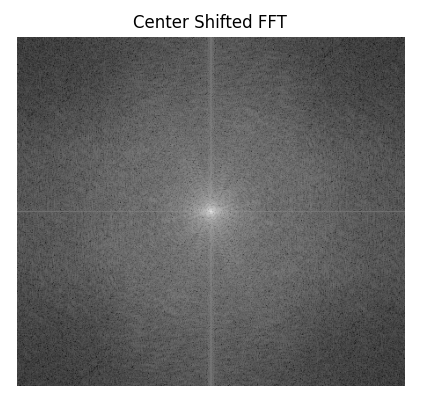
1. Follow code piece 3 to perform center shifting of the FFT.



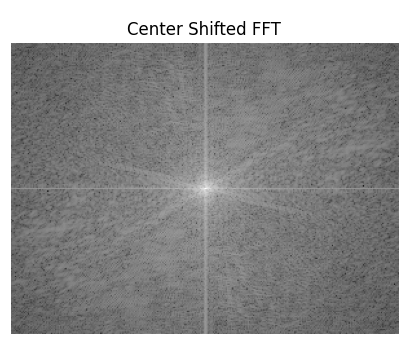
Code piece 3

1. State your observation about the change you observe in the FFT
2. Perform the same task on cute\_cat image and re-state your observation.
3. Paste the center shifted FFT images here.

FOR DOG :-

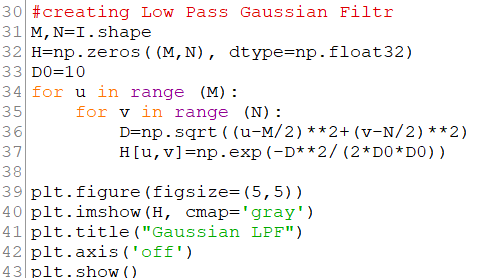


FOR CAT :-



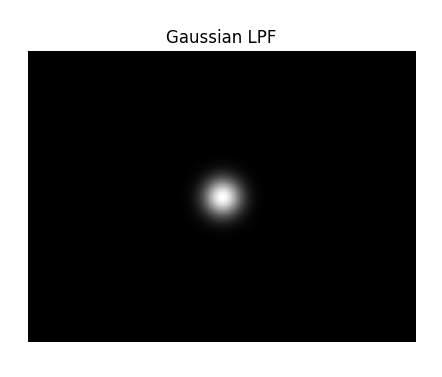
**Task 3. Create a Gaussian filter using the Euclidian distance, and the filter formula**.

Refer to code piece 4 and implement the Gaussian Low Pass Filter.



Code piece 4

Paste the output here.

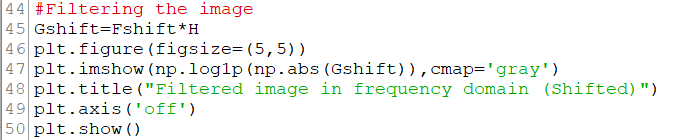


Answer the following questions.

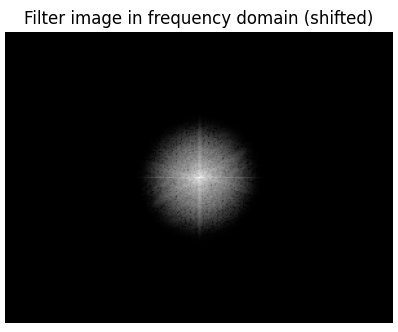
1. What does the command I.shape do?
2. What is the size of the matrix H in line 32 and what does it contain?
3. What is D0?
4. What are you computing in line 36?
5. From line 37, write down the formula for the filter H(u,v).
6. State your observation about the filter figure, related to smoothness of the intensity change.
7. Justify that this is a LPF.

**Task 4: Filtering the image and displaying in frequency domain.**

Follow code piece 5 and paste the output below it.



Code piece 5

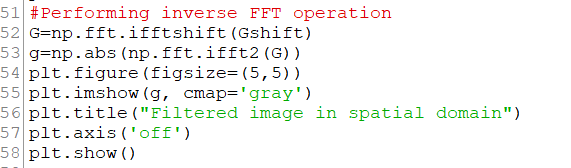


Answer the following questions.

1. Describe what the variable Gshift in line 44 is about.
2. Why is ‘abs’ (meaning absolute value) used in line 47?
3. State your observation about the image that you see.

**Task 5: Performing inverse FFT operation and displaying the filtered image in spatial domain**

Follow code piece 6 and paste the output.



Code piece 6



Answer the following questions.

1. What does the variable G in line 52 represent?
2. In which line actually inverse fft performed?
3. What is the effect of changing variable G in line 53 to Gshift?

**Additional Tasks:**

1. Select an image with lot of high frequency content and observe how this program works on it.
2. Change value of D0 to 5 and then to 25 and state your observations.
3. State the changes that will take place in line 37 (code piece 4) if one wants to design an ideal low pass filter.

**Conclusion: Write in your own words, how fairly the objectives stated in the beginning of write-up are achieved for you, through this experiment.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Self-Learning is the Best Learning\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***