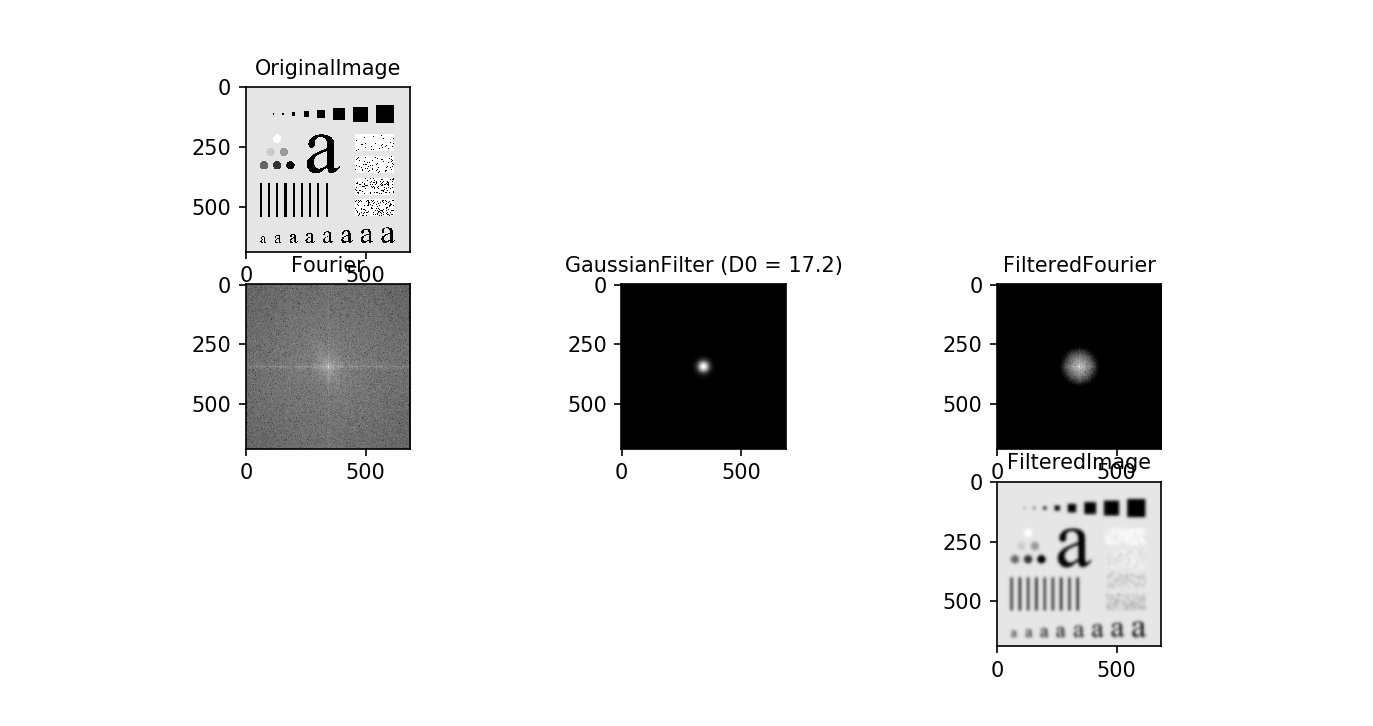
<Q1. Gaussian low-pass filter >

Perform Fourier transform to the given image (character.tif). Make the kernel of the Gaussian low-pass filter and apply it to the image. Perform the inverse Fourier transform to get the filtered image.

==================source code==========================

import matplotlib.pyplot as plt  
from skimage import io, filters, feature  
import numpy  
from scipy import fftpack, signal  
import scipy  
  
def gkern(kernlen, nsig):  
 # Return 2D Gaussian Kernel  
 gkern1d = signal.gaussian(kernlen, std=nsig).reshape(kernlen, 1)  
 kernel = numpy.outer(gkern1d, gkern1d)  
 return kernel  
  
# Load image file  
fpath = "C:/Users/fxk/PycharmProjects/tenjumh/Computer Vision/191018/"#'./PycharmProjects/tenjumh/Computer Vision/191018/'  
image = io.imread(fpath + 'character.tif').astype('float64')  
# Fourier Transform  
imageFFT = fftpack.fft2(image)  
imageFFT = fftpack.fftshift(imageFFT)  
  
# Gaussian Filter  
sizeI, sizeJ = image.shape  
d0 = sizeI/40 # cutoff fregquency, equivalent with sigma  
  
GaussianFilter = gkern(sizeI, d0) # kernel  
  
# Apply filter  
imageFFTFiltered = numpy.multiply(imageFFT, GaussianFilter)  
  
# Inverse Fourier Transform  
imageIFFT = fftpack.ifft2(imageFFTFiltered)  
imageFiltered = scipy.absolute(imageIFFT)  
  
# Plot  
plt.figure(figsize=(10, 10), dpi=150)  
plt.subplot(3, 3, 1)  
plt.imshow(image, cmap='gray')  
plt.title('OriginalImage', {'fontsize': 10})  
plt.subplot(3, 3, 4)  
plt.imshow(scipy.log10(1+scipy.absolute(imageFFT)), cmap='gray')  
plt.title('Fourier', {'fontsize': 10})  
plt.subplot(3, 3, 5)  
plt.imshow(GaussianFilter, cmap='gray')  
plt.title('GaussianFilter (D0 = ' + str(d0) + ')', {'fontsize': 10})  
plt.subplot(3, 3, 6)  
plt.imshow(scipy.log10(1+scipy.absolute(imageFFTFiltered)), cmap='gray')  
plt.title('FilteredFourier', {'fontsize': 10})  
plt.subplot(3, 3, 9)  
plt.imshow(imageFiltered, cmap='gray')  
plt.title('FilteredImage', {'fontsize': 10})  
  
plt.show()

================== output ========================================



<Q2. Gaussian high-pass filter >

Perform the same analyses with the Gaussian high-pass filter.

==================source code==========================

import matplotlib.pyplot as plt  
from skimage import io, filters, feature  
import numpy  
from scipy import fftpack, signal  
import scipy  
  
def gkern(kernlen, nsig):  
 # Return 2D Gaussian Kernel  
 gkern1d = signal.gaussian(kernlen, std=nsig).reshape(kernlen, 1)  
 kernel = numpy.outer(gkern1d, gkern1d)  
 return kernel  
  
# Load image file  
fpath = "C:/Users/fxk/PycharmProjects/tenjumh/Computer Vision/191018/"#'./PycharmProjects/tenjumh/Computer Vision/191018/'  
image = io.imread(fpath + 'character.tif').astype('float64')  
  
# Fourier Transform  
imageFFT = fftpack.fft2(image)  
imageFFT = fftpack.fftshift(imageFFT)  
  
# Gaussian Filter  
sizeI, sizeJ = image.shape  
d0 = sizeI/40 # cutoff fregquency, equivalent with sigma  
  
GaussianFilter = 1- gkern(sizeI, d0) # kernel(Gaussian High Pass Filter)  
  
# Apply filter  
imageFFTFiltered = numpy.multiply(imageFFT, GaussianFilter)  
  
# Inverse Fourier Transform  
imageIFFT = fftpack.ifft2(imageFFTFiltered)  
imageFiltered = scipy.absolute(imageIFFT)  
  
# Plot  
plt.figure(figsize=(10, 10), dpi=150)  
plt.subplot(3, 3, 1)  
plt.imshow(image, cmap='gray')  
plt.title('OriginalImage', {'fontsize': 10})  
plt.subplot(3, 3, 4)  
plt.imshow(scipy.log10(1+scipy.absolute(imageFFT)), cmap='gray')  
plt.title('Fourier', {'fontsize': 10})  
plt.subplot(3, 3, 5)  
plt.imshow(GaussianFilter, cmap='gray')  
plt.title('GaussianFilter (D0 = ' + str(d0) + ')', {'fontsize': 10})  
plt.subplot(3, 3, 6)  
plt.imshow(scipy.log10(1+scipy.absolute(imageFFTFiltered)), cmap='gray')  
plt.title('FilteredFourier', {'fontsize': 10})  
plt.subplot(3, 3, 9)  
plt.imshow(imageFiltered, cmap='gray')  
plt.title('FilteredImage', {'fontsize': 10})  
  
plt.show()

================output===========================================

