import numpy as np

import scipy.ndimage as ndi

from skimage import io, color, util

import matplotlib.pyplot as plt

# Load and convert image to grayscale

image\_path = r"C:\Users\admin\Desktop\abdul kalam.jpg"

image = io.imread(image\_path)

gray\_image = color.rgb2gray(image) # Convert to grayscale

# Mean Filter

def mean\_filter(image, size=3):

return ndi.uniform\_filter(image, size=size)

# Median Filter

def median\_filter(image, size=5):

return ndi.median\_filter(image, size=size)

# Gaussian Filter

def gaussian\_filter(image, sigma=1.0):

return ndi.gaussian\_filter(image, sigma=sigma)

# Sobel Filter

def sobel\_filter(image):

sobel\_x = ndi.sobel(image, axis=0)

sobel\_y = ndi.sobel(image, axis=1)

magnitude = np.hypot(sobel\_x, sobel\_y)

# Normalize magnitude to [0, 1]

magnitude = (magnitude - np.min(magnitude)) / (np.max(magnitude) - np.min(magnitude))

return magnitude

# Apply filters

mean\_result = mean\_filter(gray\_image)

median\_result = median\_filter(gray\_image)

gaussian\_result = gaussian\_filter(gray\_image)

sobel\_result = sobel\_filter(gray\_image)

# Convert results to 8-bit format

def convert\_to\_ubyte(image):

return util.img\_as\_ubyte(image)

mean\_result\_ubyte = convert\_to\_ubyte(mean\_result)

median\_result\_ubyte = convert\_to\_ubyte(median\_result)

gaussian\_result\_ubyte = convert\_to\_ubyte(gaussian\_result)

sobel\_result\_ubyte = convert\_to\_ubyte(sobel\_result)

# Save results

io.imsave('mean\_filtered\_image.png', mean\_result\_ubyte)

io.imsave('median\_filtered\_image.png', median\_result\_ubyte)

io.imsave('gaussian\_filtered\_image.png', gaussian\_result\_ubyte)

io.imsave('sobel\_filtered\_image.png', sobel\_result\_ubyte)

# Display images using matplotlib

def display\_image(image, title):

plt.figure(figsize=(8, 8))

plt.imshow(image, cmap='gray')

plt.title(title)

plt.axis('off') # Hide axis

plt.show()

# Display results

display\_image(mean\_result, 'Mean Filter')

display\_image(median\_result, 'Median Filter')

display\_image(gaussian\_result, 'Gaussian Filter')

display\_image(sobel\_result, 'Sobel Filter')