Image Processing Filters - Full Code PDF

---------- Below are all Python codes included in the notebook.

```
# IMPORTS
import numpy as np
import matplotlib.pyplot as plt
from scipy.ndimage import binary_erosion, binary_dilation, median_filter, maximum_filter, minimum_filter

# FREQUENCY DOMAIN SETUP
img = np.ones((256,256)) * 128 + np.random.randint(-50,50,(256,256))
img = np.clip(img,0,255)
F = np.fft.fftshift(np.fft.fft2(img))
M, N = img.shape
u, v = np.meshgrid(np.arange(-N//2, N//2), np.arange(-M//2, M//2))
D = np.sqrt(u**2 + v**2)

# IDEAL LPF
D0 = 40
H = (D <= D0).astype(float)
ideal_lpf = np.abs(np.fft.ifft2(np.fft.ifftshift(H * F)))

# BUTTERWORTH LPF
D0 = 40; n = 2
H = 1 / (1 + (D/D0)**(2*n))
bw_lpf = np.abs(np.fft.ifft2(np.fft.ifftshift(H * F)))

# GAUSSIAN LPF
D0 = 40
H = np.exp(-(D**2) / (2*D0**2))
gaussian_lpf = np.abs(np.fft.ifft2(np.fft.ifftshift(H * F)))

# IDEAL HPF
H = (D > 40).astype(float)
ideal_hpf = np.abs(np.fft.ifft2(np.fft.ifftshift(H * F)))

# BUTTERWORTH HPF
D0 = 40; n = 2
H = 1 / (1 + (D0/(D+1e-8))**(2*n))
bw_hpf = np.abs(np.fft.ifft2(np.fft.ifftshift(H * F)))

# GAUSSIAN HPF
D0 = 40
H = 1 - np.exp(-(D**2)/(2*D0**2))
gaussian_hpf = np.abs(np.fft.ifft2(np.fft.ifftshift(H * F)))

# MORPHOLOGY SETUP
img = np.zeros((200,200))
img[50:150, 50:150] = 1
se = np.ones((5,5))

# EROSION
eroded = binary_erosion(img, structure=se)

# DILATION
dilated = binary_dilation(img, structure=se)

# OPENING
opening = binary_dilation(binary_erosion(img, se), se)

# CLOSING
closing = binary_erosion(binary_dilation(img, se), se)

# ORDER FILTERS SETUP
noisy = img + np.random.normal(0,0.3,img.shape)

# MEDIAN FILTER
median_img = median_filter(noisy, size=3)
```

```
# MAX FILTER
max_img = maximum_filter(noisy, size=3)

# MIN FILTER
min_img = minimum_filter(noisy, size=3)

# MIDPOINT FILTER
midpoint = generic_filter(noisy, lambda v: (np.max(v) + np.min(v)) / 2, size=3)
```