Cody Costa

EE263

8/4/2025 resubmission

Homework 2

PYTHON IMAGE RESULTS:

Please see attached code files to assignment submission for step-by-step procedure

PROBLEM 1)

Input Images: Clean Lena image

Salt+Pepper and Gaussian noise Lena (used for filtering)

Output images:

Median Filter

Smoothing Filter

Combined Filter

IN THIS CASE, THE MEDIAN FILTER GIVES THE SHARPEST IMAGE OF THE 3 FILTERS

PROBLEM 2)

Input Image: Lena with random noise

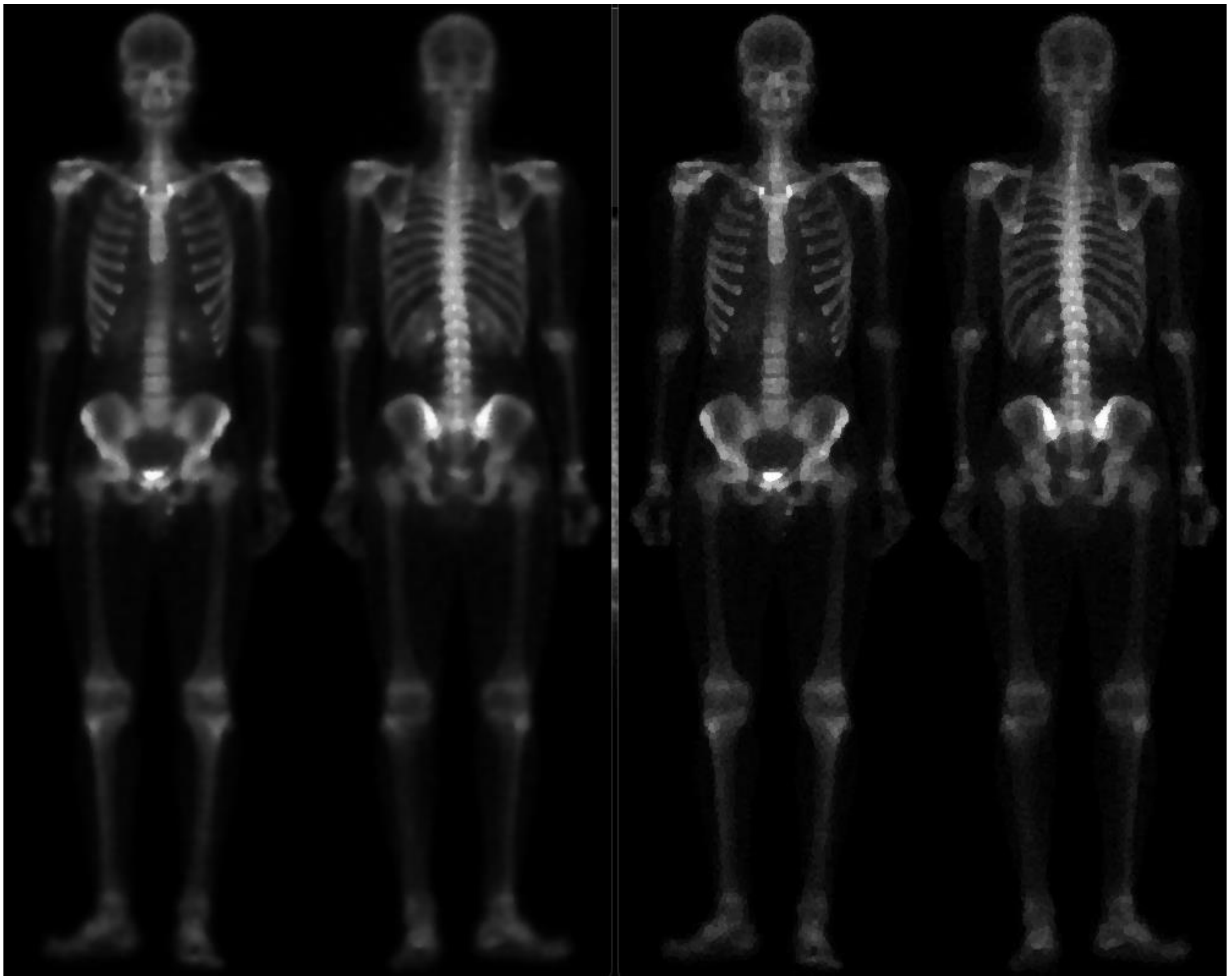
Output Image: Smoothed, filtered image

PROBLEM 3)

Input Image:

Output Images:

Bilateral Filter (left) Nagao Matsuyama filter (right)



PROBLEM 4)

Input Image:

Output Images:

Standard Bilateral Filter

G(D) / I computation

G(D) / I^2 computation

IN THIS CASE, THE CUSTOM VECTORIZED BILATERAL FILTERS PRESERVE EDGES BETTER THAN THE REGULAR FILTER DOES

PROBLEM 5)

Computations on sample section of LENA image are as follows:

```python
''' PROBLEM 5:  ENTROPY '''

# 8×8 image matrix
G = np.array([
    [139, 144, 149, 153, 155, 155, 155, 155],
    [144, 151, 153, 156, 159, 156, 156, 156],
    [150, 155, 160, 163, 158, 156, 156, 156],
    [159, 161, 162, 160, 160, 159, 159, 159],
    [159, 160, 161, 162, 162, 155, 155, 155],
    [161, 161, 161, 161, 160, 157, 157, 157],
    [162, 162, 161, 163, 162, 157, 157, 157],
    [162, 162, 161, 161, 163, 158, 158, 158]
])

# RMSE approximation (Gaussian noise sigma=5)
sigma = 5
rmse = sigma
psnr = 20 * np.log10(255 / rmse)

# Entropy
unique, counts = np.unique(G, return_counts=True)
probs = counts / G.size
entropy = -np.sum(probs * np.log2(probs))

print("RMSE:", rmse)            # 5
print("PSNR (dB):", psnr)       # 34.15 or 10pi
print("Entropy:", entropy)      # 3.57
```