EEE 309/EEE 311 SIGNALS AND SYSTEMS

INTERDISCIPLINARY TERM PROJECT REPORT

PROJECT NO. #7

Image Smoothing using Convolution

GROUP MEMBERS		
Name - Surname	**Student No.**	**Department**
Arifali Baghirli	210201848	CENG
Emre Aghan	230201010	CENG
Ulash Sarp Kaya	220201035	CENG
Shukran Umay Uslu	220201034	CENG
Huseyin Kilic	200201036	CENG
Amine Wachich	210201942	CENG
Abdulrahman Said Usman	220202926	EEE
Nursena Guler	220202016	EEE

GRADE				
PÇ4.1	**PÇ4.2**	**PÇ6.2**	**PÇ7.3**	**TOTAL**
/ 10	**/ 10**	**/ 30**	**/ 50**	**/ 100**

1. Introduction

Image smoothing is a fundamental operation in signal and image processing, widely used to reduce noise and enhance visual quality. In its most basic form, smoothing is achieved by applying convolutional filters that average or weight neighboring pixels to produce a cleaner representation of an image. The specific technical problem investigated in this project is the removal of noise from a real-world photograph using convolution-based smoothing techniques, namely the Gaussian filter and the Box (mean) filter.

This problem is significant because noise reduction is a critical preprocessing step in applications such as computer vision, medical imaging, satellite image interpretation, and feature extraction. Effective smoothing improves the performance of subsequent processing steps by reducing unwanted disturbances while preserving essential structural information.

The measurable goals of this project are:

• Resize an input photograph to 500×500 pixels.

• Convert the image from RGB to grayscale.

• Apply Gaussian smoothing using a convolutional kernel.

• Apply Box (mean) smoothing using a convolutional kernel.

• Analyze visual differences between the original and smoothed images.

The scope of this project is limited to two convolution-based smoothing algorithms and does not include edge detection, sharpening, color-space conversions beyond grayscale, or advanced denoising techniques such as median filtering or bilateral filtering.

2. Theoretical Background

Convolution is a mathematical operation that combines two signals to produce a third signal. In image processing, convolution is performed between an image I(x, y) and a kernel K(m, n). The discrete 2D convolution formula is:

```
(I * K)(x, y) = Σ Σ I(x - m, y - n) · K(m, n)
```

where K is a small matrix (kernel) that slides across the image to compute weighted sums of neighboring pixel intensities.

Gaussian smoothing uses a kernel defined by the 2D Gaussian function:

```
    G(x, y) = 1/(2πσ²) · exp(-(x² + y²) / (2σ²))
```

This kernel gives higher weight to central pixels, producing smooth and natural blurring while preserving overall structure.

Box filtering, or mean filtering, uses a kernel where all weights are equal. For an N×N kernel:

```
    K = 1/N²  [all ones matrix]
```

This filter replaces each pixel with the average of its surrounding neighborhood.

Both filters reduce high-frequency components (noise), making images smoother.

3. Methodology

The input signal for this project is a real photograph captured using a mobile device. The output signals are the resized image, grayscale image, Gaussian-smoothed image, and box-smoothed image. The following methodology was applied:

1. Image Acquisition

   A natural scene photo was captured and stored as `realPhoto.jpg`.

2. Preprocessing Steps

   • Resize image to 500×500 pixels.

   • Convert RGB image to grayscale.

3. Smoothing Algorithms

   • Gaussian Blur using radius = 3.

   • Box Blur (mean filter) using radius = 3.

4. Tools and Libraries

- Programming Language: Python
- Libraries: Pillow (PIL), Matplotlib
- Execution: VS Code terminal

The data was generated directly from the input photograph using programmatic transformations and filters.

4. Results and Discussion

Four images were produced from the experiment:

1. Resized image (500×500)
2. Grayscale image
3. Gaussian-filtered image
4. Box-filtered image

Gaussian Filter Results:

The Gaussian blur produced a smooth, natural-looking output. Noise was reduced effectively, and important features remained recognizable. The weighted nature of the Gaussian kernel prevents excessive loss of detail.

Box Filter Results:

The Box filter created a stronger smoothing effect but also introduced more noticeable blurring of edges. Since all neighboring pixels are weighted equally, the filter tends to flatten regions more aggressively, reducing local contrast.

Comparison:

- Gaussian filter preserves structure better.
- Box filter smooths uniformly but may over-blur important details.
- Both filters reduce high-frequency noise as expected from convolution theory.

No major implementation errors were encountered. Slight differences from theoretical expectations occurred due to finite kernel size and digital rounding effects.

5. Conclusion

This project successfully achieved all objectives defined in the Introduction. A real-world photograph was resized, converted to grayscale, and processed using Gaussian and Box convolution filters. Both smoothing techniques reduced noise, with Gaussian filtering providing more natural results.

Key takeaways include the importance of convolution in image processing, the distinction between weighted and uniform smoothing, and the observable impact of kernel design on image quality.

Future improvements may include experimenting with median filters, bilateral filters, edge-preserving smoothing, or comparing smoothing effects across color channels.

6. Statement of Contributions

- Emre Ağan — Assisted with theoretical background research and mathematical formulations.
- Ulaş Sarp Kaya — Supported methodology design and code structure.
- Şükran Umay Uslu — Helped analyze smoothing results and image differences.
- Arifali Baghirli — Implemented the Python code, performed experiments, generated outputs.
- Hüseyin Kılıç — Contributed to writing and formatting the report.
- Amine Wachich — Prepared figure descriptions and visual interpretation.
- Nursena Güler — Prepared literature review and reference formatting.
- Abdulrahman Said Usman — Helped with proofreading and final document alignment.

REFERENCES

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Pearson, 2018.

[2] S. Marchand-Maillet and Y. M. Sharaiha, *Binary Digital Image Processing*. Academic Press, 2000.

[3] Python Pillow Library Documentation:

[4] Matplotlib Documentation: https://matplotlib.org

APPENDIX #1

Python code used in the project:

```
# Image Smoothing using Convolution
from PIL import Image, ImageFilter, ImageOps
import matplotlib.pyplot as plt
import os
# Ensure assets/ folder exists
os.makedirs("assets", exist_ok=True)
# 1. Load Image
img = Image.open("assets/realPhoto.jpg")
# 2. Resize image to 500x500 pixels
img_resized = img.resize((500, 500))
# 3. Convert image to grayscale
img_gray = ImageOps.grayscale(img_resized)
# 4. Apply Gaussian Smoothing
img_gaussian = img_gray.filter(ImageFilter.GaussianBlur(radius=3))
# 5. Apply Box (Mean) Filter
img_box = img_gray.filter(ImageFilter.BoxBlur(radius=3))
# 6. Save all output images into assets/
img_resized.save("assets/resized_500x500.jpg")
img_gray.save("assets/grayscale.jpg")
img_gaussian.save("assets/gaussian_filtered.jpg")
img_box.save("assets/box_filtered.jpg")
# 7. Display Results
```

```
plt.figure(figsize=(12, 8))

plt.subplot(2, 2, 1)

plt.imshow(img_resized)

plt.title("Resized 500x500")

plt.axis("off")

plt.subplot(2, 2, 2)

plt.imshow(img_gray, cmap="gray")

plt.title("Grayscale")

plt.axis("off")

plt.subplot(2, 2, 3)

plt.imshow(img_gaussian, cmap="gray")

plt.title("Gaussian Filtered")

plt.axis("off")

plt.subplot(2, 2, 4)

plt.imshow(img_box, cmap="gray")

plt.title("Box Filtered")

plt.axis("off")

plt.tight_layout()

plt.show()
```

Github Repository:

The full project, including source code, assets, and documentation, is also available on GitHub:

Github Link