# Birzeit University
## Department of Electrical & Computer Engineering
## First Semester, 2024/2025
## ENCS5343 Computer Vision
## Assignment#1
### Due Date November 15, 2024

**Comparing Simple Smoothing Filters with Advanced Filters on Noisy Images**

### 1. Objective

The aim of this assignment is to compare the performance of simple smoothing filters (Box filter, Gaussian filter, and Median filter) with advanced filters (e.g., adaptive filters) when applied to noisy images. You will evaluate them based on noise removal effectiveness, edge preservation, computational efficiency, and kernel size effects. Metrics such as Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) will be used for quantitative comparison. Additionally, the assignment will explore the influence of kernel size on the performance of each filter.

### 2. Background

Image denoising is a fundamental task in computer vision, and various filters are used for this purpose:

- **Box filter**: Averages pixel values uniformly over a local neighborhood, often causing edge blurring.
- **Gaussian filter**: Applies a weighted average based on distance from the center, which better preserves edges than the Box filter.
- **Median filter**: Replaces each pixel with the median of its neighborhood, highly effective for salt-and-pepper noise while preserving edges.
- **Adaptive filters**: Dynamically adjust filtering parameters based on local image statistics, offering superior noise removal with better edge preservation, but usually at a higher computational cost.

The size of the kernel (window) used in filtering plays a significant role in performance. Larger kernels provide more smoothing but may blur details and edges. Smaller kernels preserve more detail but might be less effective in reducing noise. This assignment will explore how varying kernel sizes impact filter performance.

### 3. Experiments

1. **Step 1: Generate or load noisy images**
   - Select a set of clean images from a public dataset or your own collection.
   - Add different types of noise (e.g., Gaussian noise, Salt-and-Pepper noise) with varying intensity levels (low, medium, high).

2. **Step 2: Apply filters**
   - Apply the following filters to each noisy image with different kernel sizes:
     - Simple filters: Box filter, Gaussian filter, and Median filter.
     - Advanced filters: Adaptive mean filter, adaptive median filter, and Bilateral filter.
   - Implement the filters using Python and OpenCV or a similar library.
3. **Step 3: Measure performance**
   - **MSE and PSNR**: Calculate the Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) for each filter at various kernel sizes.
   - **Edge preservation**: Use edge detection (e.g., Canny edge detector) to evaluate how well the filters preserve edges at different kernel sizes.
   - **Computational time**: Measure and report the time taken by each filter to process the images at different kernel sizes.
   - **Kernel size effect**: Analyze how varying kernel sizes (small, medium, large) impact the performance of each filter in terms of noise reduction, edge preservation, and processing speed.

## 4. Results

In your results section, include the following:

- **MSE and PSNR comparison**: Present the MSE and PSNR values for each filter across different noise levels and kernel sizes.
- **Edge preservation comparison**: Provide visual examples of edge maps for different filters, kernel sizes, and noise levels. Show how edge details are affected by increasing the kernel size.
- **Computational time comparison**: Create a table or chart that shows how the computational time varies with different kernel sizes for each filter.
- **Effect of kernel size**: Summarize how increasing or decreasing the kernel size impacts the balance between noise removal and edge preservation for each filter.

## 5. Discussion

In your discussion, focus on:

- **Noise removal**: Compare the effectiveness of the simple filters (Box, Gaussian, Median) and advanced filters in terms of MSE and PSNR across different kernel sizes.
- **Edge preservation**: Discuss how the filters perform in terms of preserving image edges and fine details, and how kernel size influences this.
- **Computational efficiency**: Analyze how the computational time scales with the kernel size for each filter and discuss the trade-offs between processing speed and filter performance.
- **Kernel size sensitivity**: Evaluate how sensitive each filter is to changes in kernel size, and the implications for selecting an appropriate kernel size in practical applications.
- **Exploring trade-offs**: Discuss the trade-offs between noise reduction, edge preservation, and computational cost for each filter type, particularly in relation to kernel size.

## 6. Deliverables

Your final submission should include:

1. Code
2. Report
    - **Introduction** (overview of the task and filters being compared)
    - **Experiments** (detailed steps including noise generation, filter application, and evaluation methods)
    - **Results** (MSE, PSNR, edge preservation, computational time, and kernel size effect comparisons)
    - **Discussion** (analysis of the results and key insights into the performance of different filters and kernel sizes)
    - **Conclusion** (summary of findings and recommendations)