Lab Work: Applying Convolution Filters to Images

YOUB OUBODIB
Master's Program: IAA
Polydisciplinary Faculty of Ouarzazate

June 19, 2025

Abstract

This lab work investigates the use of convolution filters on grayscale and RGB images to extract and enhance image features. Implemented in Python with libraries such as NumPy, OpenCV, and Matplotlib, various kernels were applied—including blur, Sobel (horizontal and vertical), sharpening, and randomly generated filters of sizes 3×3, 5×5, and 7×7. Standard test images were used to evaluate the impact of these filters on features such as edges, smoothness, and detail enhancement. Results highlight the visual differences across filters and demonstrate their significance in image processing tasks.

This lab demonstrated the power and flexibility of convolution filters in image processing. Results showed kernel type and size significantly influence outcomes. The source code is available at: `https://github.com/youbodib/TP_Convolution_IMG`.

1 Introduction

Convolution is a fundamental technique in image processing. It applies a kernel to an image by sliding it and calculating new pixel values as a weighted sum of neighbors:

$$(I * K)(i,j) = \sum_{m=-k}^{k} \sum_{n=-k}^{k} I(i+m, j+n) \cdot K(m,n)$$

This project applies several filters (blur, Sobel, sharpen, random) to both grayscale and RGB images using Python.

2 Materials and Methods

2.1 Materials

- Python 3.x
- Libraries: NumPy, OpenCV, Matplotlib
- Images from hlevkin.com

Kernels Used

Blur (3×3):

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Sobel Horizontal:

$$\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}$$

Sobel Vertical:

$$\begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix}$$

Sharpen:

$$\begin{bmatrix}0 & -1 & 0\\ -1 & 5 & -1\\ 0 & -1 & 0\end{bmatrix}$$

Random filters of sizes 3×3, 5×5, and 7×7 were generated with a fixed seed.

2.2 Methods

1. **Loading:** Using a function to load grayscale or RGB images via OpenCV.
2. **Convolution:** A universal function applied kernels to images with padding.
3. **Implementation:** Custom functions for each filter type.
4. **Visualization:** Side-by-side comparison using Matplotlib.
5. **Saving Results:** Images saved via `cv2.imwrite`.

3 Results

- **Blur:** 3×3 kernels smoothed images; larger sizes increased blurriness.
- **Sobel:** Detected horizontal/vertical edges effectively; grayscale was clearer.
- **Sharpen:** Enhanced texture and edge detail, especially in color images.
- **Random:** Unpredictable effects, showing distortion or feature emphasis.

4 Discussion

All filters affected both grayscale and RGB images, with predictable outcomes. Random filters introduced noise or new patterns. Edge filters worked better in grayscale, while sharpening was more vivid in RGB.

5 Conclusion

This lab demonstrated the power and flexibility of convolution filters in image processing. Results showed kernel type and size significantly influence outcomes. The source code is available at: `https://github.com/youbodib/TP_Convolution_IMG`.