Generative AI Question 02

Kissa Zahra[1 [i21−0572]]

FAST National University of Computer and Emerging Sciences, Islamabad H-11, Pakistan kissasium@gmail.com

Abstract. Implementing and analyzing 2D convolution operations from scratch without relying on deep learning libraries. Various kernels were applied to grayscale images to explore how different convolution parameters affect image processing outcomes. In this report we will compares convolution versus correlation operations, examines the effects of different kernel types (including edge detection, blurring, and sharpening kernels), and investigates the impact of kernel size, stride, and padding.

Keywords: Image Processing · Convolution · Kernels · Edge Detection · Image Filtering · Computer Vision

1 First Section

1.1 Introduction

Convolution operations form the backbone of modern computer vision and image processing techniques. Despite their widespread implementation in high-level deep learning frameworks, a fundamental understanding of the underlying mathematical operations is essential for computer vision researchers and practitioners. This report details the implementation of 2D convolution from scratch and analyzes its various applications in image processing.

Convolution works by sliding a kernel (a small matrix of weights) over an input image and computing element-wise multiplications and summations at each position. This simple yet powerful operation allows for various image transformations, from edge detection to blurring and sharpening, depending on the kernel used.

The objectives of this study are:

1. To implement a flexible convolution function from scratch that supports various parameters
2. To analyze the effects of different kernels on image processing
3. To compare convolution and correlation operations
4. To examine how parameters like kernel size, stride, and padding affect the output

1.2 Methodology

Implementation of Convolution Function We implemented a generalized 2D convolution function from scratch that processes grayscale images with the following customizable parameters:

- Input image: A grayscale image to be processed
- Kernel: A user-defined kernel matrix (defaults to a random kernel if none is provided)
- Kernel size: The dimensions of the kernel matrix
- Stride: The step size for sliding the kernel across the image
- Padding: Option for "valid" (no padding) or "same" (zero-padding to maintain original dimensions)
- Mode: Option to perform either convolution (kernel flipped) or correlation (kernel as-is)

Experimental Setup For this experiments, we used a standard grayscale test image and applied various kernels to analyze their effects:

Fig. 1. original image

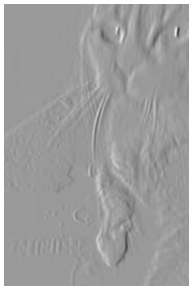

Fig. 2. Vertical Edge Detection

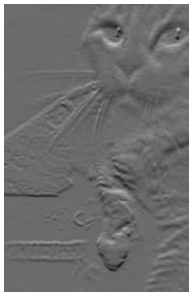

Fig. 3. Horizontal Edge Detection

Fig. 4. Blur Image

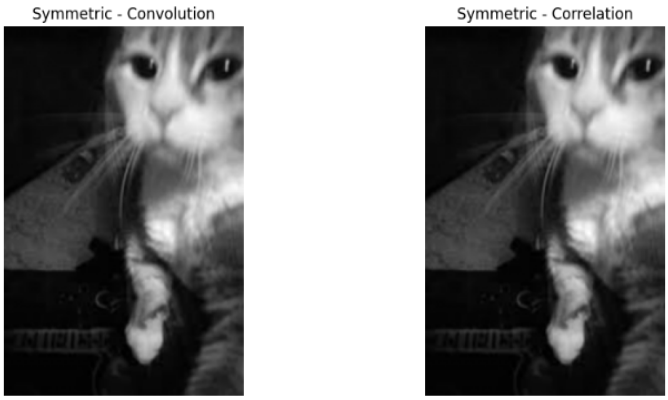

Fig. 5. Symmetric kernel Convolution vs. Correlation

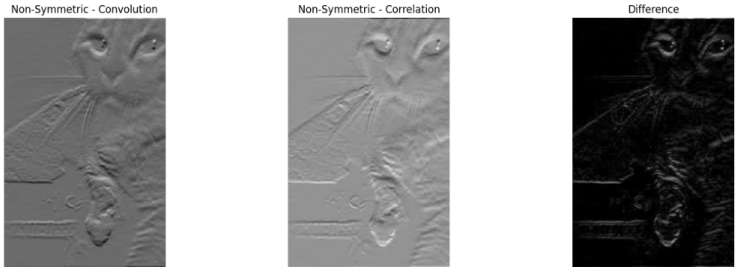

Fig. 6. Non-Symmetric kernel Convolution vs. Correlation

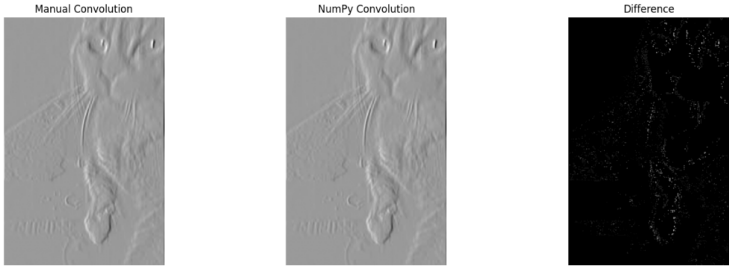

Fig. 7. Manual vs NumPy Convolution

Fig. 8. Kernel sizes: 3×3, 5×5, and 7×7

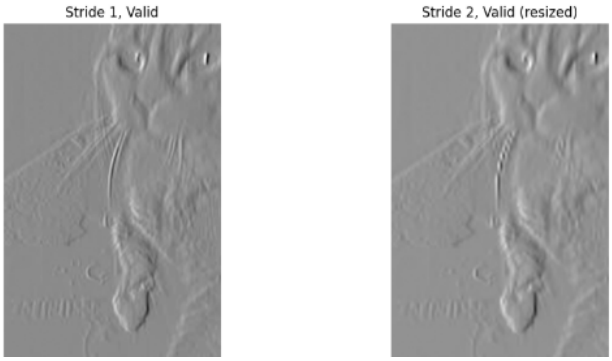

Fig. 9. Effect of stride and padding

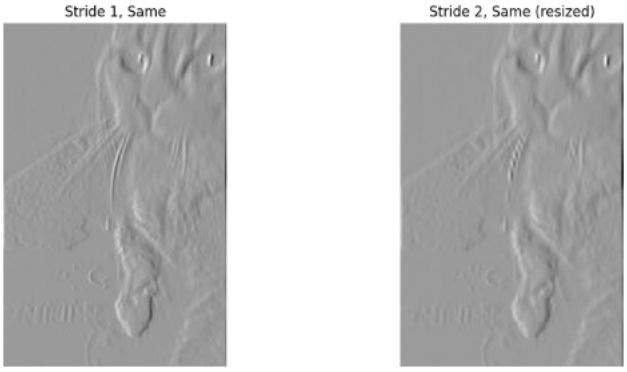

Fig. 10. Effect of stride and padding

2 Second Section

2.1 Conclusion

Blurring Kernels Blurring kernels reduce detail and noise by averaging neighboring pixels. Box blur kernels apply uniform averaging across the filter region, creating a simple smoothing effect but often introducing blocky artifacts at higher kernel sizes.

Sharpening Kernels Sharpening kernels enhance edges by amplifying the difference between pixels and their surroundings. Basic sharpening kernels boost center pixels while slightly reducing surrounding ones, creating a local contrast enhancement that emphasizes details but can amplify noise.

Impact of Kernel Parameters; Kernel Size, Stride Values, and Padding Options Kernel size significantly impacts processing results. Larger kernels (5×5, 7×7) incorporate more contextual information, making them effective for broad feature detection and stronger blur effects, but they reduce fine detail and increase computational cost. Stride values determine processing density across the image. A stride of 2 offered a reasonable compromise between efficiency and detail preservation for most applications.

Convolution vs. Correlation Analysis When using symmetric kernels (such as a 2×2 matrix with all values of 0.25 or a Gaussian kernel), convolution and correlation produce identical results regardless of operation mode. Non-symmetric kernels (like gradient operators) yield directionally opposite responses between convolution and correlation.