Morphological Image Processing Report

Md. Al-Amin Babu
ID: 2110676134

1 Objective

Six morphological operations are performed on a binary image using both OpenCV (built-in) and user-defined implementations: *erosion*, *dilation*, *opening*, *closing*, *top-hat*, and *black-hat*. Results are compared per operation.

2 Data and Settings

- Input: a grayscale image binarized by threshold > 127 into $\{0, 1\}$.
- Iterations: set to 1 for all built-in operations.
- Structuring elements (all 5×5):
  - Rectangular: all ones.
  - Elliptical: disk-like mask.
  - Cross-shaped: plus sign (center row/column).
  - Diamond-shaped: Manhattan-radius ≈ 2 mask. *Note: in this implementation, the elliptical and diamond masks are identical, so their effects coincide.*

3 Methods

Built-in (OpenCV): The functions `erode`, `dilate`, and `morphologyEx` (OPEN, CLOSE, TOPHAT, BLACKHAT) were used with `iterations` = 1.
User-defined (from scratch):

- **Erosion:** Output pixel = 1 if all pixels under kernel positions = 1.
- **Dilation:** Output pixel = 1 if any pixel under kernel positions = 1.
- **Opening/Closing:** erosion→dilation and dilation→erosion.
- **Top-hat/Black-hat:** $I - (I \circ B)$ and $(I \bullet B) - I$ on binary images.

4 Results (Comparison)

Each figure shows: top row = OpenCV built-in, bottom row = user-defined. Outputs are visually very similar, confirming correctness of the manual routines. Any minor difference comes from display scaling ($\{0, 1\}$ vs $0-255$), not from the morphology itself.

5 Conclusion

Built-in and manual implementations with 5×5 structuring elements and one iteration produce near-identical outcomes. Structuring element shape controls the strength and geometry of the effect: rectangular > ellipse/diamond > cross in aggressiveness, with cross preserving straight strokes and ellipse/diamond smoothing corners.

6 Reference (Code Link)

Explained script (kept hidden in this report):
`https://github.com/Al-Amin134/Digital-Image-Processing/blob/main/morphology.py`

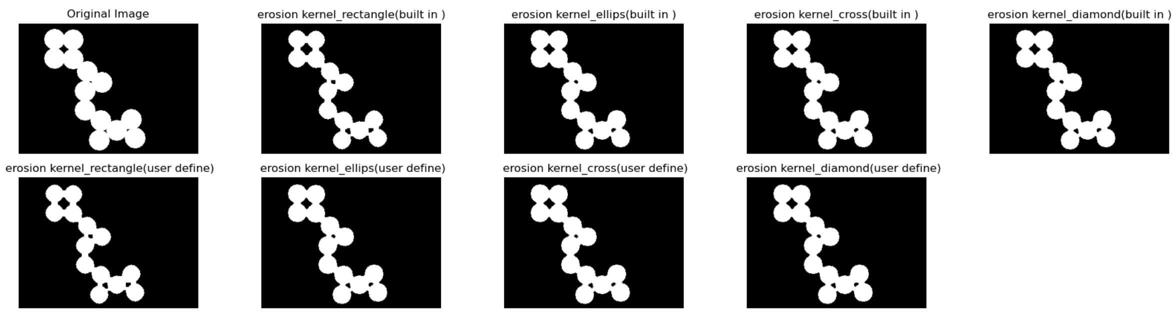

Figure 1: Erosion: Both methods thin bright regions and remove details smaller than the 5×5 SE. Rectangle is most aggressive; cross preserves straight strokes. Built-in vs manual: near-identical.

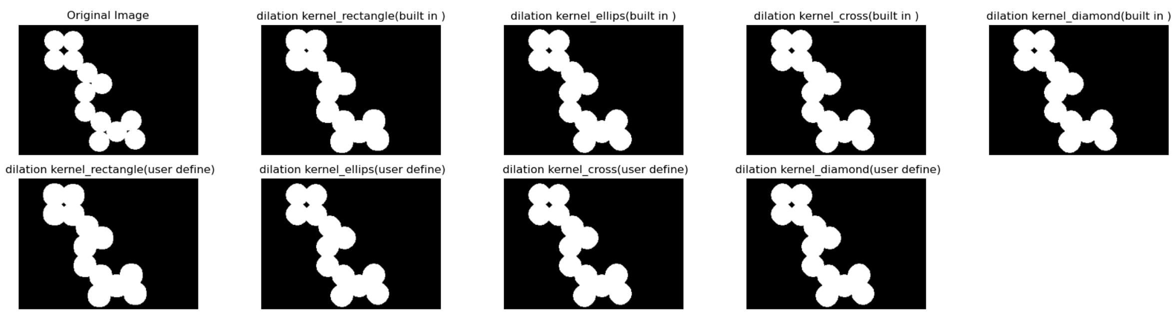

Figure 2: Dilation: Bright regions expand consistently in both methods; small gaps are bridged. Rectangle dilates most; cross favors vertical/horizontal growth. Outputs match closely.

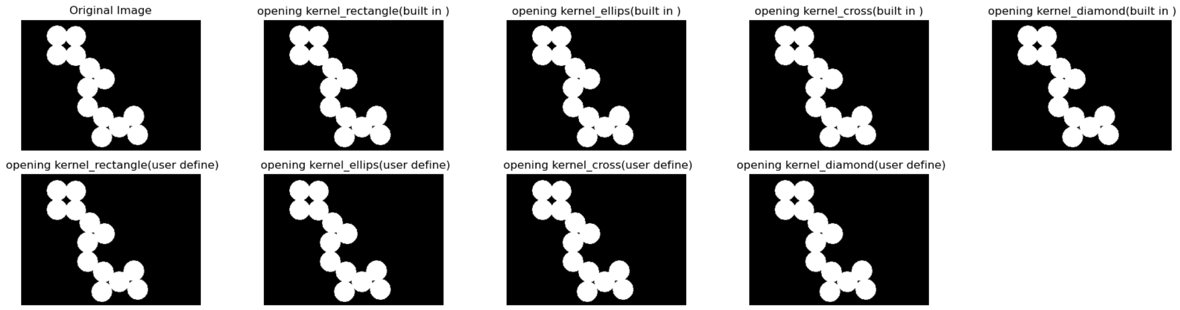

Figure 3: Opening: Noise removal with shape preservation is consistent across methods. Ellipse/diamond (identical here) preserve rounded corners slightly better than rectangle; cross may erode diagonals more.

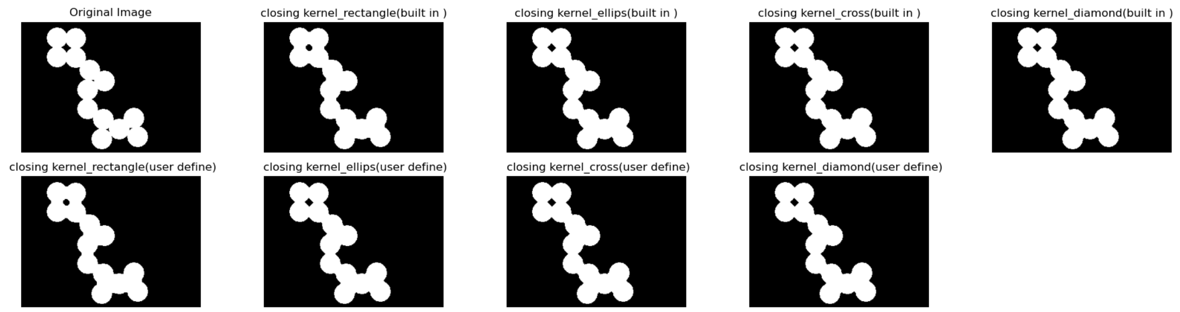

Figure 4: Closing: Small holes and narrow gaps are filled similarly in both pipelines. Rectangle connects regions most; cross may leave diagonal gaps. Visual parity between built-in and manual.

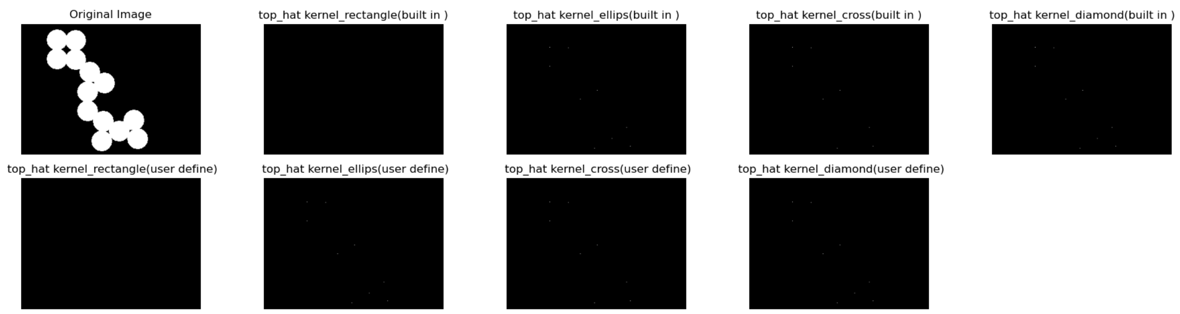

Figure 5: Top-Hat: Highlights small bright residues; both implementations isolate similar features. Differences, if any, stem from display scaling, not operation logic.

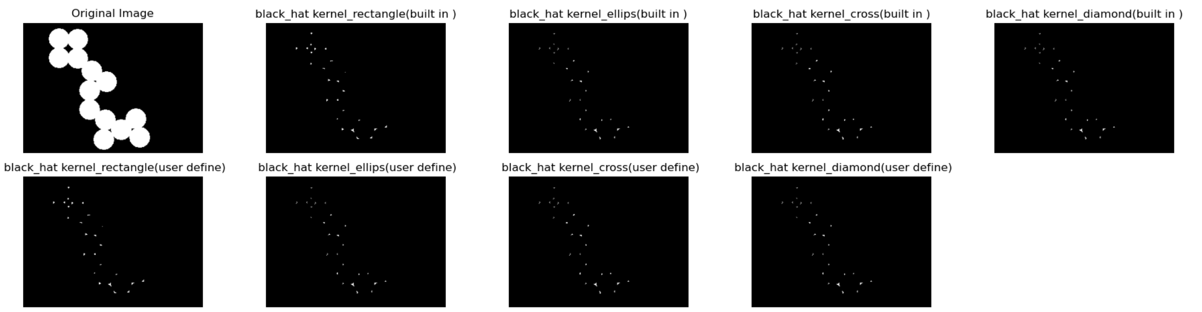

Figure 6: Black-Hat: Emphasizes small dark details on bright background. Built-in and user-defined outputs align strongly; rectangle produces the strongest emphasis.