YAŞAR
UNIVERSITY

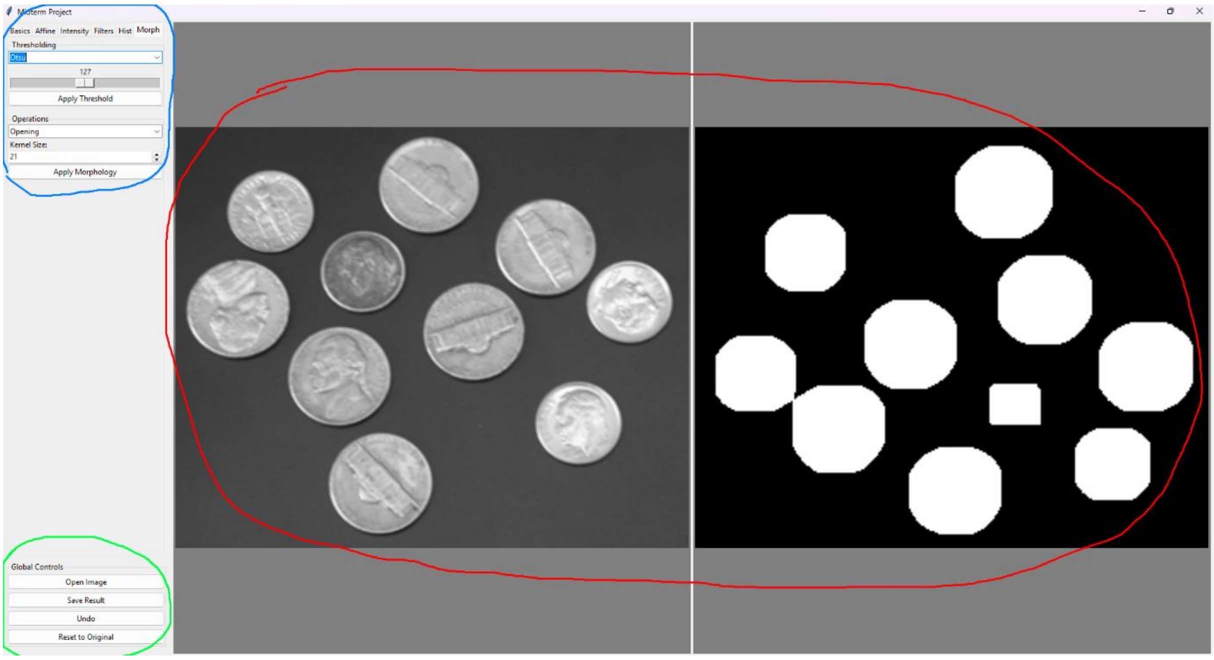
COMP 4360
IMAGE PROCESSING
COMPUTER ENGINEERING DEPARTMENT
MIDTERM PROJECT

30.11.2025

MUSTAFA GÖKSU ERDOĞAN
22070001052

1. App Overview and UI

I built this desktop application to demonstrate the image processing techniques we learned in class. I used Python, Tkinter (for the interface), and OpenCV. It allows the user to upload their image, use image processing techniques and save the result.



Thresholding: Otsu

127

Operations: Opening

Kernel Size: 21

Interface has three main sections, highlighted with colours blue, red and green.

1. The Toolbox (Top Left): This is where I put all the buttons. I grouped them into categories like "Basics", "Filters", and "Morph" so it is easy to find what you need.

2. Controls (Bottom Left): Here, I put the important buttons. You can open an image, save your work, or use Undo if you make a mistake or Reset to the original image.

3. Display Area (Right Side): This part shows two images. On the left, you see the Original Image, and on the right, you see the Processed Image. This helps to compare the changes instantly.

2. Implemented Features

Basics Tab

- **Grayscale:** Converts the image to black and white.
- **Flip Horizontal:** Mirrors the image left-to-right.
- **Flip Vertical:** Mirrors the image up-to-down.

Affine Tab (Transformations)

- **Rotation:** I used an input box so the user can enter any angle (like 45 or 90 degrees).
- **Scale:** Resizes the image. You can set X and Y values separately.
- **Translate:** Moves the image by a specific X and Y distance.
- **Shear:** Distorts the image along the axes.

Intensity Tab

- **Contrast Stretching:** Automatically fixes the contrast range.
- **Negative:** Inverts the colors.
- **Gamma Correction:** I used a slider for this. It goes from 1 to 50, but in the code, I divide it by 10. So, you can choose values like 0.5 or 2.5 easily.

Filters Tab

- **Blurring:** I added Mean, Gaussian, and Median filters to remove noise.
- **Edge Detection:** I implemented Laplacian, Sobel X, and Sobel Y.

Hist Tab (Histogram)

- **Show Histogram:** Opens a separate popup window showing the color or grayscale graph.
- **Equalization:** Enhances the contrast using histogram equalization.

Morph Tab (Morphology)

- **Thresholding:**

  Otsu: Automatically finds the best threshold.

  Manual: I added a slider to choose a value between 0 and 255.

- **Operations:** Erosion, Dilation, Opening, and Closing. Rectangle kernel was used.

Reseting to the original was implemented.

3. Known Limitations

1. **Speed:** If you load a very large image and try a heavy operation like Median Filter, the app might freeze for a second. This is because Tkinter runs on one thread.
2. **Memory:** The Undo feature saves a copy of the image every time. If you do 50 operations on a big image, it might use a lot of RAM.
3. **Kernel Shape:** For morphological operations, I used a Rectangular kernel. I didn't add an option to change it to a Cross or Ellipse shape in the UI to keep it simple.