
1) Project Overview

This repo demonstrates four traditional segmentation pipelines:

Otsu Thresholding – global, histogram-based binary segmentation.

K-Means Clustering – color clustering for multi-region segmentation.

Contour Detection – edge detection followed by contour extraction/visualization.

Watershed – marker-based region growing for touching-object separation.

Each pipeline is a standalone Python script for clarity and quick experimentation.

2) Requirements & Installation

Supported on Python 3.9–3.12.

(Optional) create and activate a virtual environment

```
python -m venv venv
```

Windows

```
venv\Scripts\activate
```

macOS / Linux

```
source venv/bin/activate
```

Upgrade installer

```
python -m pip install -U pip
```

Install dependencies

```
pip install -r requirements.txt
```

Minimal requirements.txt:

```
numpy
```

```
matplotlib
```

```
opencv-contrib-python
```

If you installed opencv-python and miss functions like selectROIs, switch to opencv-contrib-python.

3) Quickstart

Place a small test image in data/ (e.g., data/sample.jpg) and run any script from src/:

```
python "src/Otsu Thresholding.py" --image data/sample.jpg
```

```
python "src/K-Means Algorithm.py" --image data/sample.jpg --k 3
```

```
python "src/Contour Detection.py" --image data/sample.jpg
```

```
python "src/Watershed Algorithm.py" --image data/sample.jpg
```

If your scripts don't yet accept CLI arguments, open them and set the input path near the top, then run without flags. GUI windows (`cv2.imshow`, ROI selector, or `plt.show`) require a desktop environment.

4) Troubleshooting

- **ModuleNotFoundError: cv2** → `pip install opencv-contrib-python`
- **No GUI/Windows open** (headless server) → run locally with a desktop environment, or modify code to save figures instead of showing them.
- **selectROIs missing** → uninstall `opencv-python`, install `opencv-contrib-python`.
- **Unicode path issues on Windows** → avoid non-ASCII paths; wrap paths in quotes.