Day 12: Contour Detection

Outcomes:

- Understand what **contours** are
- Detect contours from a binary image
- Filter contours by area
- Draw bounding boxes around detected objects

Contour

What are Contours?

Contours are continuous boundaries of objects.

In OpenCV:

- They are detected from binary images.
- Usually from masks or threshold images.

Contour is simply the outline of detected objects.

Contours cannot be found directly from a color image.

Typical source: mask (binary image)

- Object of interest: White (255)
- Everything else: Black (0)

Finding Contours

```
contours, hierarchy = cv.findContours(
    mask,
    cv.RETR_EXTERNAL,
    cv.CHAIN_APPROX_SIMPLE
)
```

What it does:

- Scans the binary image
- Finds white regions
- Traces their boundaries

cv.RETR_EXTERNAL

- Detects only Outer Contours
- Ignores Contours inside Contours
- Perfect for Object Detection

cv.CHAIN_APPROX_SIMPLE

- Saves memory
- Stores only important points

Drawing Contours

Syntax:

```
cv.drawContours(image, contours, contourIdx, color, thickness)
```

`image`	Image to be drawn on
`contours`	List of contours returned by `cv.findContours()`
`contourIdx`	`-1` → draw all contours `0, 1, 2…` → draw only that index contour
`color`	Color of the contour line Format: `(B, G, R)`
`thickness`	Line thickness in `px`

Example:

```
cv.drawContours(frame, contours, -1, (0,255,0), 2)
```

- -1 → draw all contours
- Color → green
- Thickness → 2

Filtering Contours by Area

Small contour = noise

Syntax:

```
for cnt in contours:
    if cv.contourArea(cnt) > min_area:
        ...
```

Explanation:

- `for cnt in contours` Go through each detected contour one by one
- `cv.contourArea(cnt)` Get area of the current contour
- `min_area` Threshold to remove noise
    - small value → more noise
    - larger value → cleaner detection

Example:

```
for cnt in contours:
    area = cv.contourArea(cnt)
    if area > 500:
        . . .
```

Bounding Boxes around Objects

Syntax:

```
x, y, w, h = cv.boundingRect(contour)
```

x	x-coordinate of top-left corner
y	y-coordinate of top-left corner
w	Width of rectangle
h	Height of rectangle

Example:

```
x, y, w, h = cv.boundingRect(cnt)
cv.rectangle(frame, (x,y), (x+w, y+h), (255,0,0), 2)
```

Object Detection = Convert → Threshold → Mask → Morphology → Contours → Bounding Boxes → Apply

20-contourDetection.py

```python
import cv2 as cv
import numpy as np

cap = cv.VideoCapture(0)
if not cap.isOpened:
    print("Camera not accessible.")
    exit

cv.namedWindow("Trackbars")

def nothing(x):
    pass

cv.createTrackbar("LH", "Trackbars", 0, 179, nothing)
cv.createTrackbar("LS", "Trackbars", 0, 255, nothing)
cv.createTrackbar("LV", "Trackbars", 0, 255, nothing)
cv.createTrackbar("UH", "Trackbars", 179, 179, nothing)
cv.createTrackbar("US", "Trackbars", 255, 255, nothing)
cv.createTrackbar("UV", "Trackbars", 255, 255, nothing)

kernel = np.ones((5,5), np.uint8)

while True:
    ret, frame = cap.read()
    if not ret:
        print("An error occurred.")
        break
```

```
    hsv = cv.cvtColor(frame, cv.COLOR_BGR2HSV)

    lh = cv.getTrackbarPos("LH", "Trackbars")
    ls = cv.getTrackbarPos("LS", "Trackbars")
    lv = cv.getTrackbarPos("LV", "Trackbars")
    uh = cv.getTrackbarPos("UH", "Trackbars")
    us = cv.getTrackbarPos("US", "Trackbars")
    uv = cv.getTrackbarPos("UV", "Trackbars")

    lower_bound = np.array([lh, ls, lv])
    upper_bound = np.array([uh, us, uv])

    mask = cv.inRange(hsv, lower_bound, upper_bound)

    opened = cv.morphologyEx(mask, cv.MORPH_OPEN, kernel)

    contours, hierarchy = cv.findContours(opened,
cv.RETR_EXTERNAL, cv.CHAIN_APPROX_SIMPLE)

    cv.drawContours(frame, contours, -1, (0,255,0), 1)

    for cnt in contours:
        if cv.contourArea(cnt) > 500:
            x, y, w, h = cv.boundingRect(cnt)

    cv.rectangle(frame, (x,y), (x+w, y+h), (255, 0, 0), 1)
```

```
    cv.imshow("Result", frame)

    if cv.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv.destroyAllWindows()
```