OPEN CV

1. What is an image

- In OpenCV, an image is represented as a **NumPy array** (matrix of pixel values).
- Each pixel has values:
    - **Grayscale:** single intensity value (0–255).
    - **Color:** 3 values (B, G, R in OpenCV by default).

```
import cv2
import numpy as np

# Creating a blank black image (300×300, 3 channels for BGR)
img = np.zeros((300, 300, 3), dtype=np.uint8)
cv2.imshow("Blank Image", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

2. Reading an image

```
img = cv2.imread("image.jpg")   # Read in color
gray = cv2.imread("image.jpg", cv2.IMREAD_GRAYSCALE)  # Read in grayscale
cv2.imshow("Color Image", img)
cv2.imshow("Gray Image", gray)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

3. Converting it to grayscale

```
img = cv2.imread("image.jpg")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
cv2.imshow("Grayscale", gray)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

4. Playing with RGB color channels

```
b, g, r = cv2.split(img)
cv2.imshow("Blue Channel", b)
cv2.imshow("Green Channel", g)
cv2.imshow("Red Channel", r)

merged = cv2.merge([b, g, r])
cv2.imshow("Merged", merged)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

5. Resize

```
resized = cv2.resize(img, (200, 200))  # Resize to 200×200
cv2.imshow("Resized", resized)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

---

6. Flipping

```
flip1 = cv2.flip(img, 0)   # Vertical flip
flip2 = cv2.flip(img, 1)   # Horizontal flip
flip3 = cv2.flip(img, -1)  # Both directions
cv2.imshow("Flip Vertical", flip1)
cv2.imshow("Flip Horizontal", flip2)
cv2.imshow("Flip Both", flip3)
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

7. Cropping

```
cropped = img[50:200, 100:300]   # [y1:y2, x1:x2]
cv2.imshow("Cropped", cropped)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

8. Saving using `imwrite`

```
cv2.imwrite("saved_image.jpg", img)
```

9. Stacking multiple images

```
hstack = np.hstack((img, img))   # Horizontal stack
vstack = np.vstack((img, img))   # Vertical stack
cv2.imshow("Horizontal", hstack)
cv2.imshow("Vertical", vstack)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

10. Drawing shapes (rectangles, circles, lines) and texts

```
canvas = np.zeros((400, 400, 3), dtype="uint8")

cv2.rectangle(canvas, (50, 50), (200, 200), (0, 255, 0), 2)
cv2.circle(canvas, (300, 100), 50, (255, 0, 0), -1)
cv2.line(canvas, (0, 0), (400, 400), (0, 0, 255), 3)
```

```
cv2.putText(canvas, "OpenCV", (100, 350), cv2.FONT_HERSHEY_SIMPLEX,
1, (255, 255, 255), 2)

cv2.imshow("Shapes", canvas)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

11. Live Direct Drawing

```
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret: break
    cv2.circle(frame, (100,100), 40, (0,255,0), 2)
    cv2.imshow("Live Drawing", frame)
    if cv2.waitKey(1) & 0xFF == 27:  # ESC to quit
        break
cap.release()
cv2.destroyAllWindows()
```

12. Fetching webcam stream (in black and white)

```
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret: break
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow("Webcam Gray", gray)
    if cv2.waitKey(1) & 0xFF == 27:
        break
cap.release()
cv2.destroyAllWindows()
```

13. Recording live stream

```
cap = cv2.VideoCapture(0)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi', fourcc, 20.0, (640,480))

while True:
    ret, frame = cap.read()
    if not ret: break
    out.write(frame)
    cv2.imshow('Recording', frame)
    if cv2.waitKey(1) & 0xFF == 27:
        break

cap.release()
out.release()
cv2.destroyAllWindows()
```

14. Working with videos (shapes)

```
cap = cv2.VideoCapture("video.mp4")
while True:
    ret, frame = cap.read()
    if not ret: break
    cv2.rectangle(frame, (50, 50), (200, 200), (0, 255, 0), 2)
    cv2.imshow("Video with Shapes", frame)
    if cv2.waitKey(25) & 0xFF == 27:
        break
cap.release()
cv2.destroyAllWindows()
```

15. Writing on live video

```
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret: break
    cv2.putText(frame, "Live Video", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255,255), 2)
    cv2.imshow("Text on Live", frame)
    if cv2.waitKey(1) & 0xFF == 27:
        break
cap.release()
cv2.destroyAllWindows()
```

16. Image Processing Operations

Examples: blur, edge detection.

```
blur = cv2.GaussianBlur(img, (7,7), 0)
edges = cv2.Canny(img, 100, 200)

cv2.imshow("Blurred", blur)
cv2.imshow("Edges", edges)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

17. Object Detection and Tracking

Example using Haar Cascade (face detection).

```
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade_frontalface_default.xml")
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    if not ret: break
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,0), 2)
    cv2.imshow("Face Detection", frame)
    if cv2.waitKey(1) & 0xFF == 27:
        break

cap.release()
cv2.destroyAllWindows()
```