Table of Contents

1. Installation

```
pip install opencv-python opencv-python-headless
```

Verify Installation:

```
import cv2
print(cv2.__version__)
```

---

2. Import OpenCV

```
import cv2
```

---

3. Reading and Displaying an Image

```
img = cv2.imread('image.jpg')
cv2.imshow('Image', img)
cv2.waitKey(0)  # Wait for a key press
cv2.destroyAllWindows()  # Close the window
```

---

4. Writing and Saving an Image

```
cv2.imwrite('output.jpg', img)
```

---

5. Resizing an Image

```
resized = cv2.resize(img, (300, 200))
cv2.imshow('Resized', resized)
cv2.waitKey(0)
```

---

6. Converting to Grayscale

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow('Grayscale', gray)
cv2.waitKey(0)
```

---

7. Drawing Shapes

Draw a Line:

```
cv2.line(img, (50, 50), (200, 50), (0, 255, 0), 3)
```

Draw a Rectangle:

```
cv2.rectangle(img, (60, 60), (200, 150), (255, 0, 0), 2)
```

Draw a Circle:

```
cv2.circle(img, (150, 150), 40, (0, 0, 255), -1)  # -1 fills the
circle
```

Put Text on Image:

```
cv2.putText(img, 'Hello OpenCV', (50, 250), cv2.FONT_HERSHEY_SIMPLEX,
1, (255, 255, 255), 2)
```

---

8. Image Filtering

Blurring an Image:

```
blur = cv2.GaussianBlur(img, (15, 15), 0)
```

Edge Detection (Canny):

```
edges = cv2.Canny(img, 100, 200)
cv2.imshow('Edges', edges)
cv2.waitKey(0)
```

---

9. Rotating and Flipping

```
rotated = cv2.rotate(img, cv2.ROTATE_90_CLOCKWISE)
flipped = cv2.flip(img, 1)  # 1 = Horizontal, 0 = Vertical
```

---

10. Cropping an Image

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

---

11. Video Capture from Webcam

```
cap = cv2.VideoCapture(0)  # 0 for webcam

while True:
    ret, frame = cap.read()
    cv2.imshow('Webcam', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):  # Press 'q' to exit
        break

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

---

12. Video Writing (Save to File)

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

while cap.isOpened():
```

```
    ret, frame = cap.read()
    if ret:
        out.write(frame)
        cv2.imshow('Recording', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break

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

---

13. Face Detection (Haar Cascade)

```
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray, 1.1, 4)

for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)

cv2.imshow('Face Detection', img)
cv2.waitKey(0)
```

---

14. Key OpenCV Functions

Function	Description
cv2.imread()	Read an image
cv2.imshow()	Display an image
cv2.imwrite()	Save an image
cv2.resize()	Resize an image

Function	Description
cv2.cvtColor()	Convert color space
cv2.GaussianBlur()	Apply Gaussian blur
cv2.Canny()	Edge detection
cv2.flip()	Flip image
cv2.VideoCapture()	Capture video from webcam
cv2.VideoWriter()	Write video to file

---

15. Morphological Operations

```
kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (5, 5))
erosion = cv2.erode(img, kernel, iterations=1)
dilation = cv2.dilate(img, kernel, iterations=1)
```

---

16. Drawing Contours

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
ret, thresh = cv2.threshold(gray, 127, 255, 0)
contours, _ = cv2.findContours(thresh, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
cv2.drawContours(img, contours, -1, (0, 255, 0), 3)
cv2.imshow('Contours', img)
cv2.waitKey(0)
```

---

17. Thresholding

```
_, threshold = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)
cv2.imshow('Threshold', threshold)
cv2.waitKey(0)
```

Example: Image Pipeline (Blurring & Edges)

```
img = cv2.imread('image.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (5, 5), 0)
edges = cv2.Canny(blur, 50, 150)
cv2.imshow('Pipeline Result', edges)
cv2.waitKey(0)
```

OpenCV provides powerful tools for image processing, computer vision, and video analysis.