TECH BOOTCAMP

DAY 06

INTRODUCTION TO IMAGE PROCESSING AND TOOLS

01

INTRODUCTION TO OPENCV

OPEN SOURCE COMPUTER VISION LIBRARY

IMAGE PROCESSING LIBRARY

CRUCIAL IN COMPUTER VISION

MULTILINGUAL, CROSS-PLATFORM, EVOLVING

DETECTION, RECOGNITION, CALIBRATION

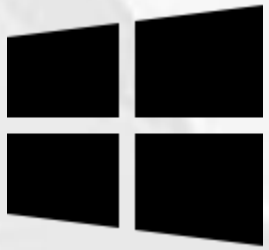
ACTIVE, RESOURCE-RICH COMMUNITY

02 INSTALLATION

OFFICIAL WEBSITE

https://opencv.org/

AVAILABLE FOR



03

BASIC IMAGE OPERATIONS

LOAD IMAGE

```
cv2.imread('path_to_image.jpg')
```

DISPLAY IMAGE

```
cv2.imshow('Loaded Image', image)
```

SAVE IMAGE

```
cv2.imwrite('saved_image.jpg', image)
```

IMAGE TO GREYSCALE

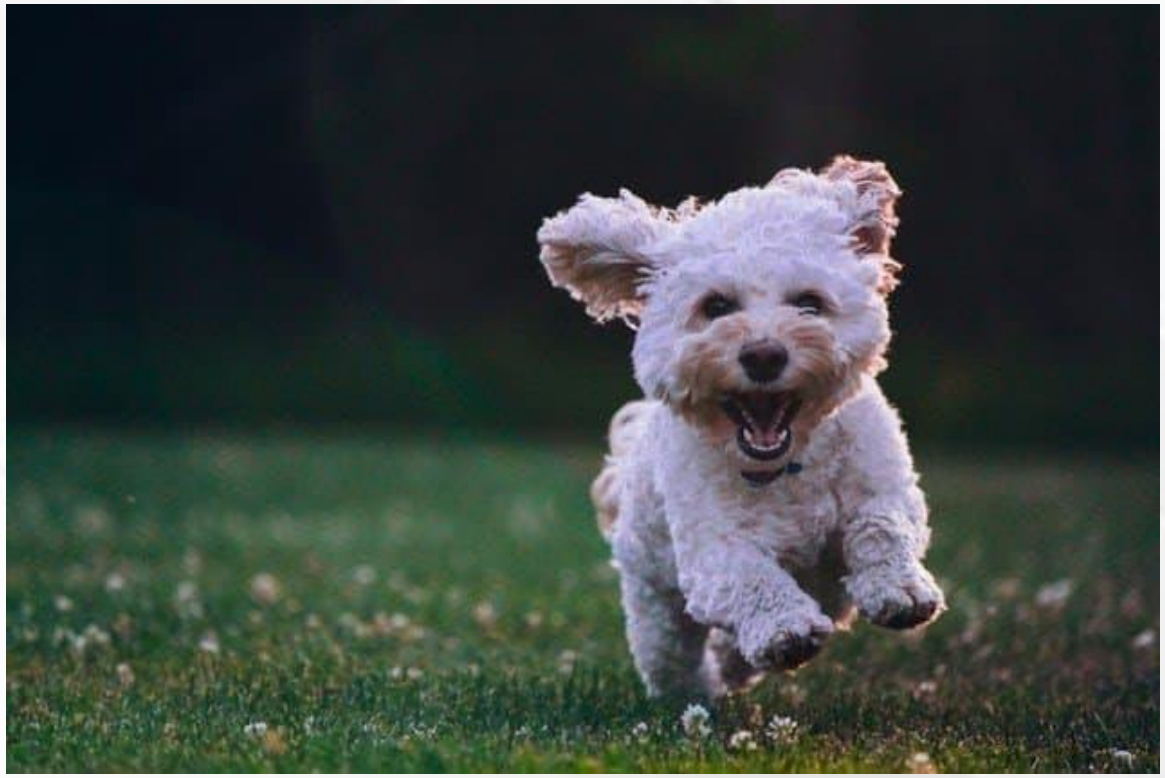
cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

RESIZE IMAGE

cv2.resize(image, (500, 500))

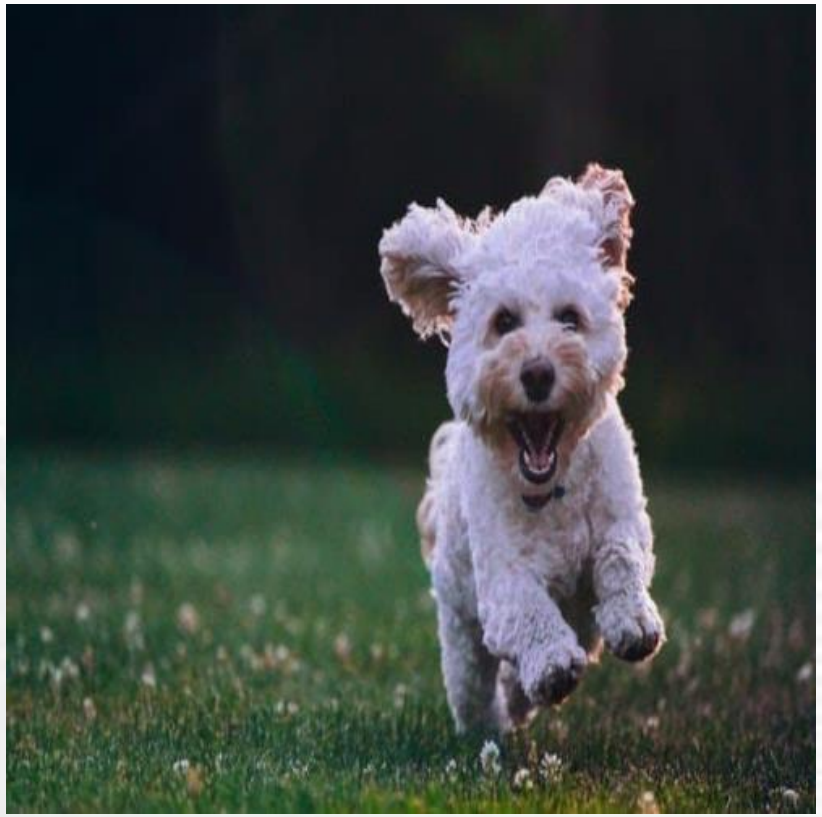
640 PX

427 PX



500 PX

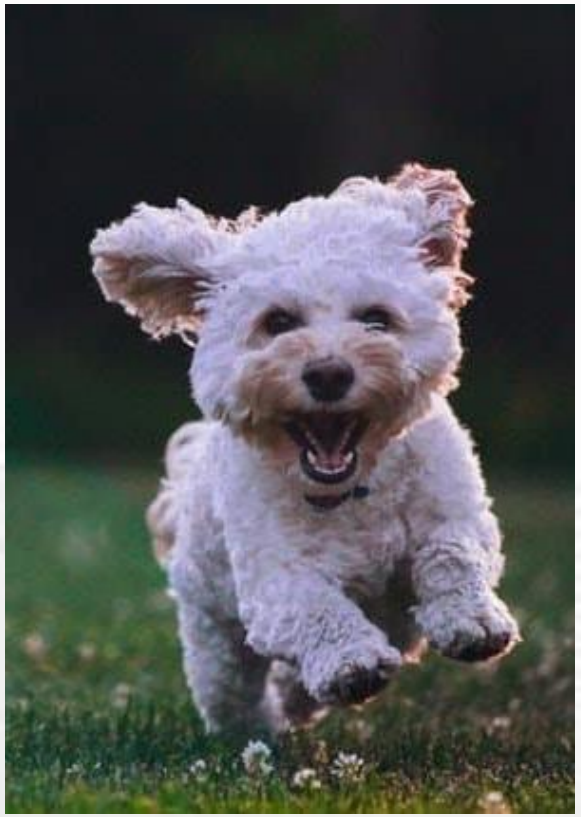
500 PX



CROP IMAGE

ORIGIN

image[y:y+h, x:x+w]



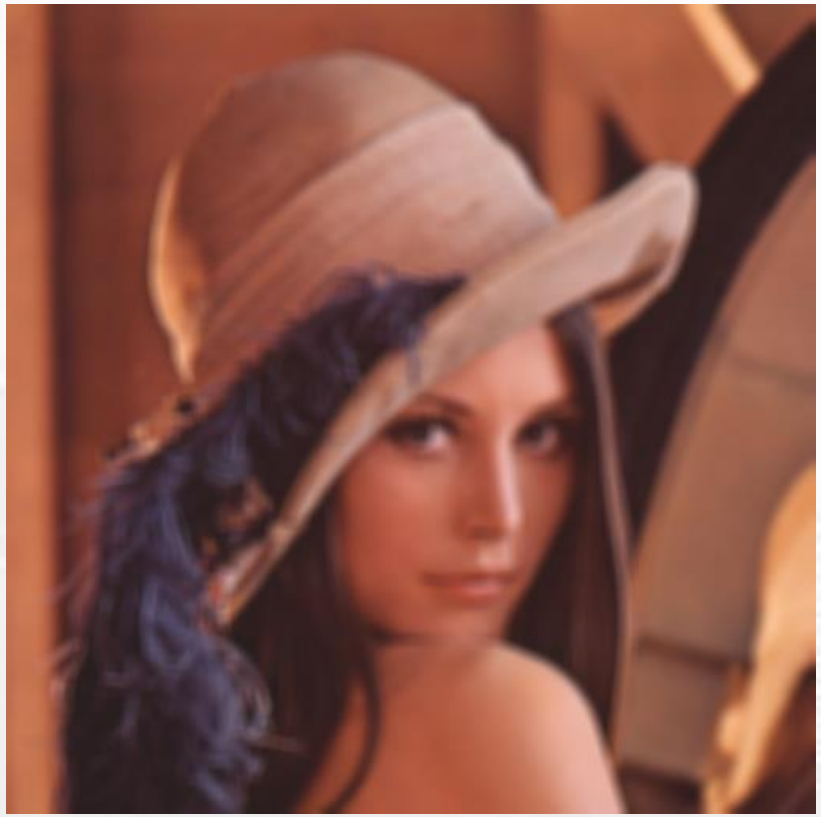
04 IMAGE FILTERING

BLURRING

AVERAGE BLURRING
GAUSSIAN BLURRING
MEDIAN BLURRING

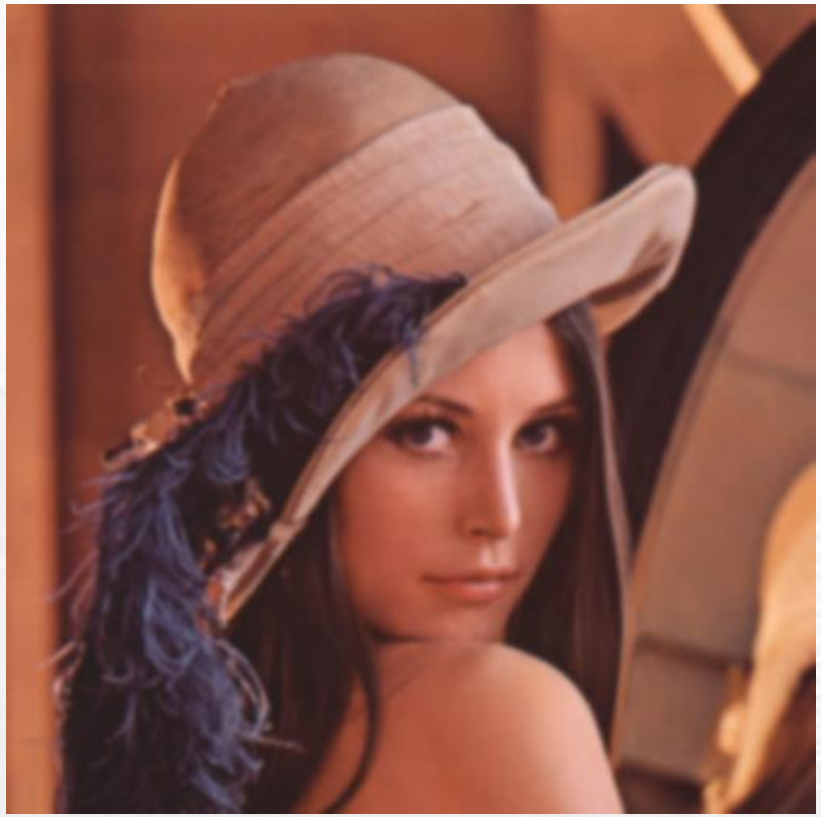
AVERAGE BLURRING

```
blurred_image = cv2.blur(image, (9, 9))
```



GAUSSIAN BLURRING

gaussian_blurred_image = cv2.GaussianBlur(image, (9, 9), 0)

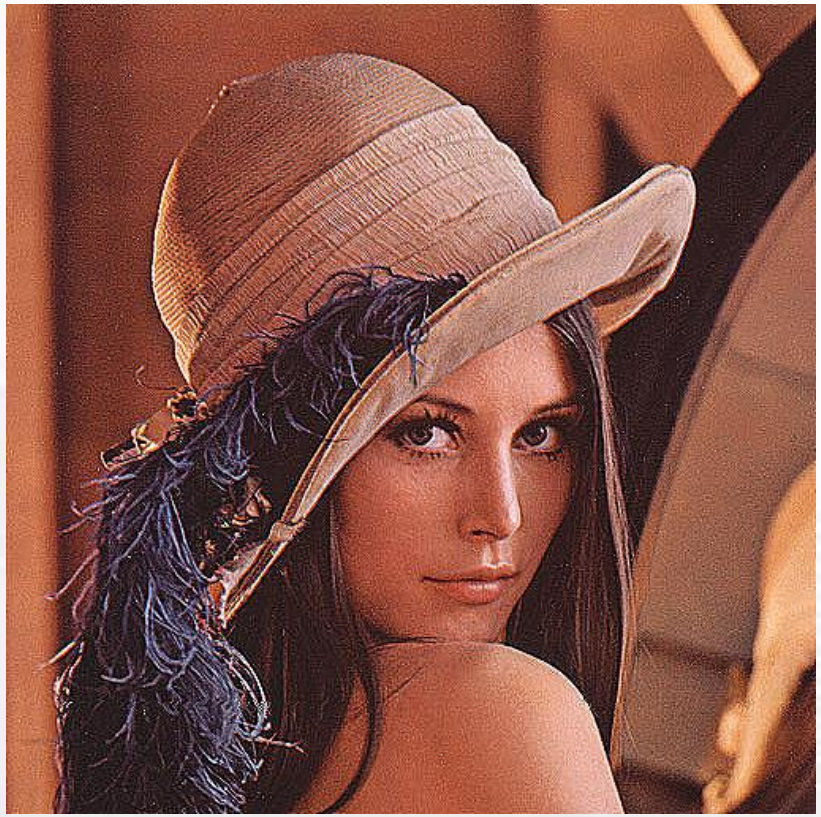


MEDIAN BLURRING

median_blurred_image = cv2.medianBlur(image, 9)

SHARPENING

cv2.filter2D(image, -1, kernel)



EDGE DETECTION

SOBEL EDGE DETECTION

LAPLACIAN EDGE DETECTION

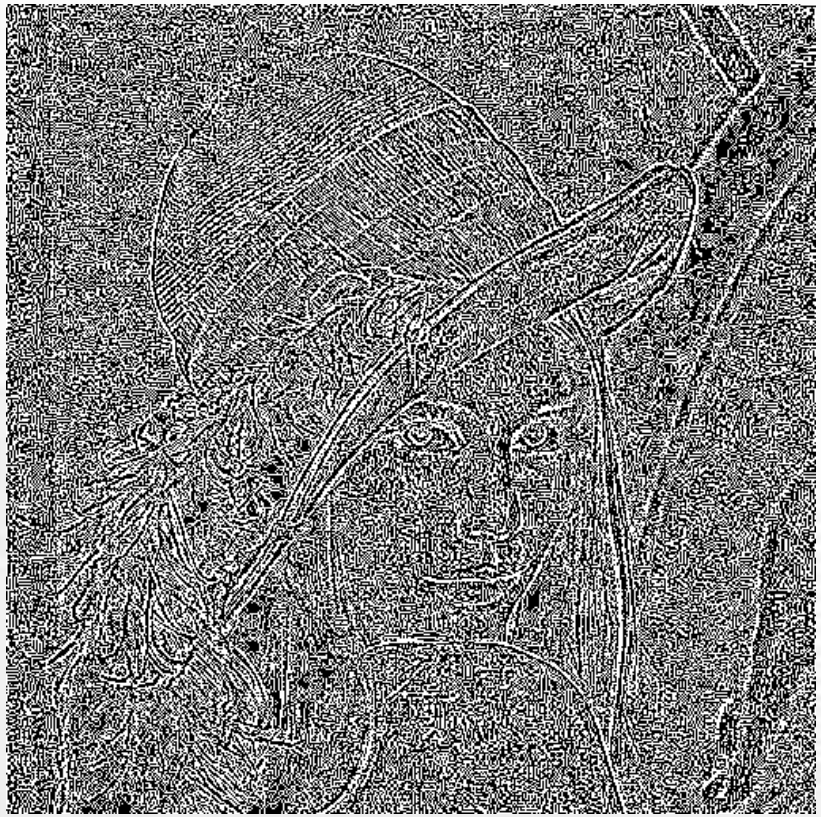
CANNY EDGE DETECTION

SOBEL EDGE DETECTION

cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=5)

LAPLACIAN EDGE DETECTION

cv2.Laplacian(image, cv2.CV_64F)



CANNY EDGE DETECTION

cv2.Canny(image, 100, 200)

05

IMAGE TRANSFORMATIONS

RESIZING

cv2.resize(image, (image.shape[1] // 2, image.shape[0] // 2))

ROTATING

cv2.getRotationMatrix2D(center, angle, scale)
cv2.warpAffine(image, rotation_matrix, (w, h))

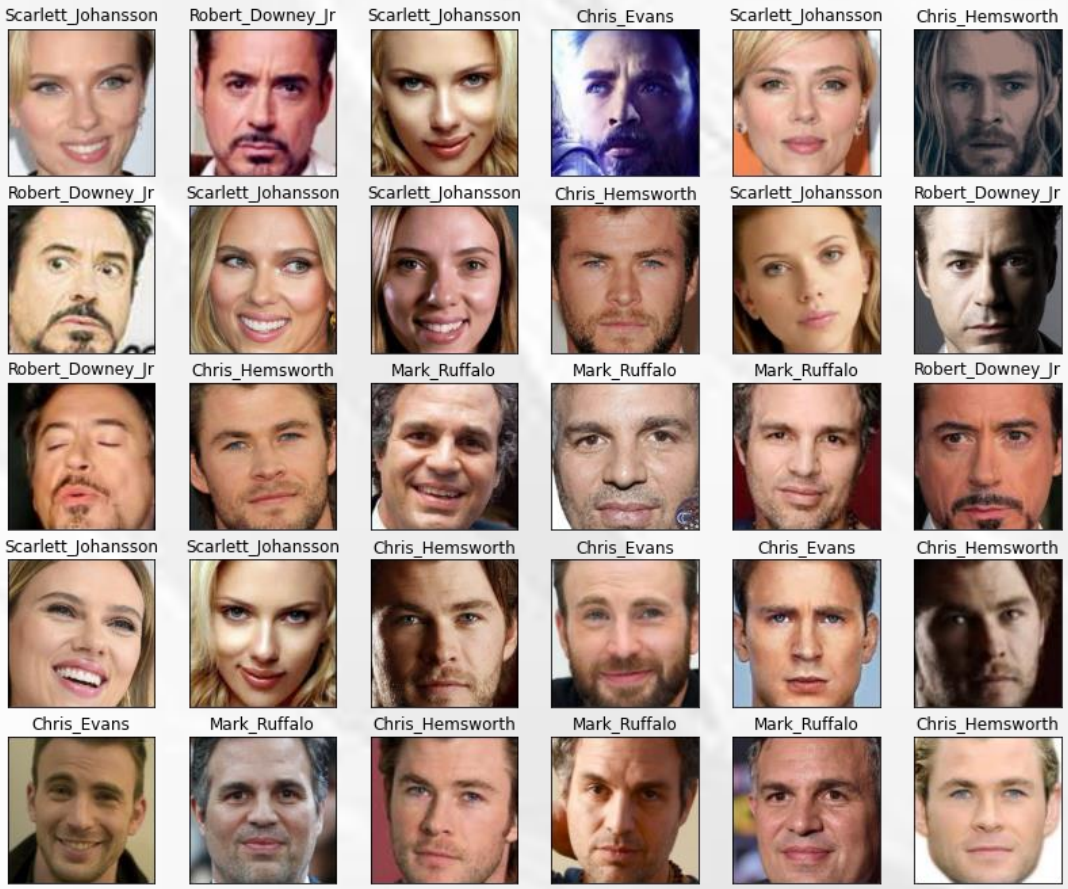
CROPPING

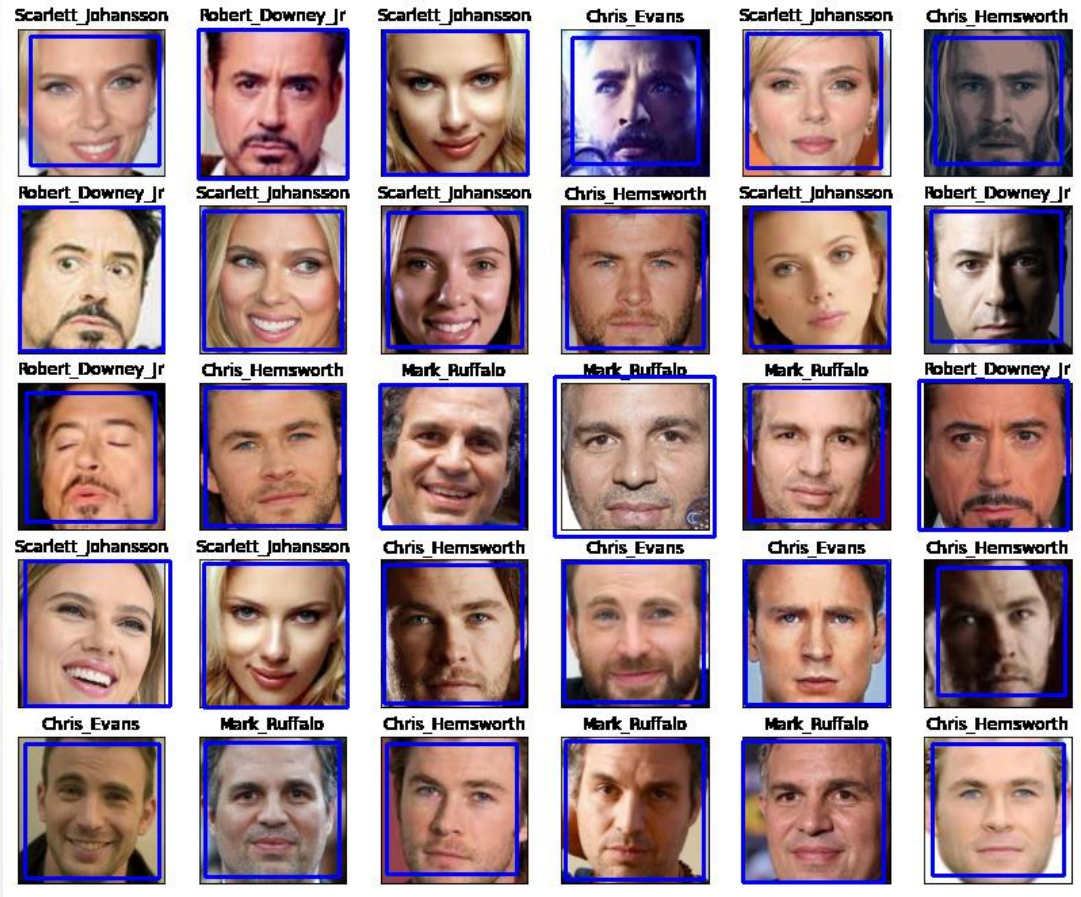
image[y:y+h, x:x+w]

06 OBJECT DETECTION

HAAR CASCADE CLASSIFIER

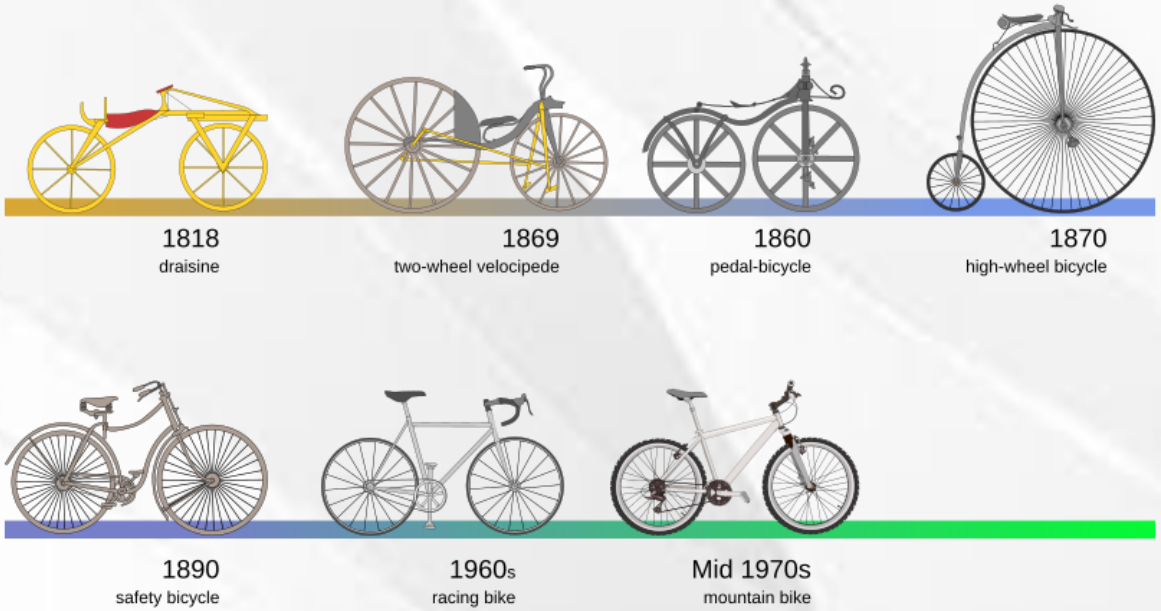
cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

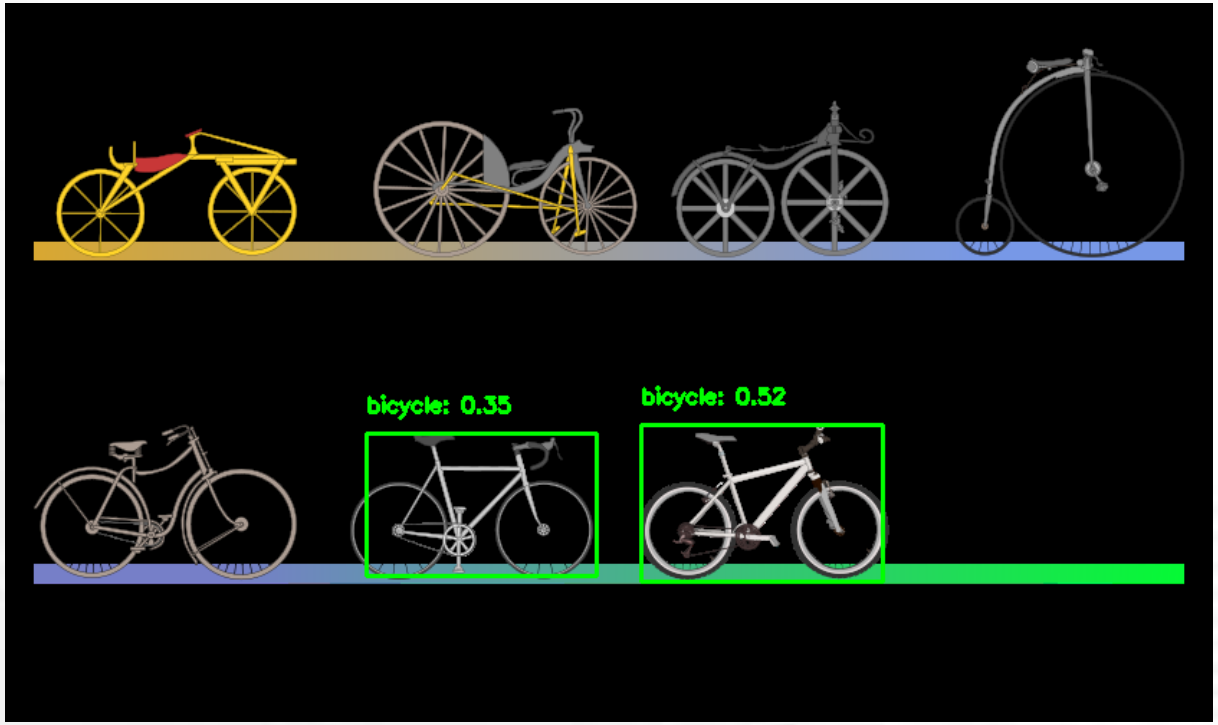




PRE-TRAINED DEEP LEARNING MODEL

cv2.dnn.readNetFromCaffe('path_to_deploy.prototxt', 'path_to_model.caffemodel')





07

IMAGE PROCESSING TECHNIQUES

THRESHOLDING

cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)

MORPHOLOGICAL OPERATIONS

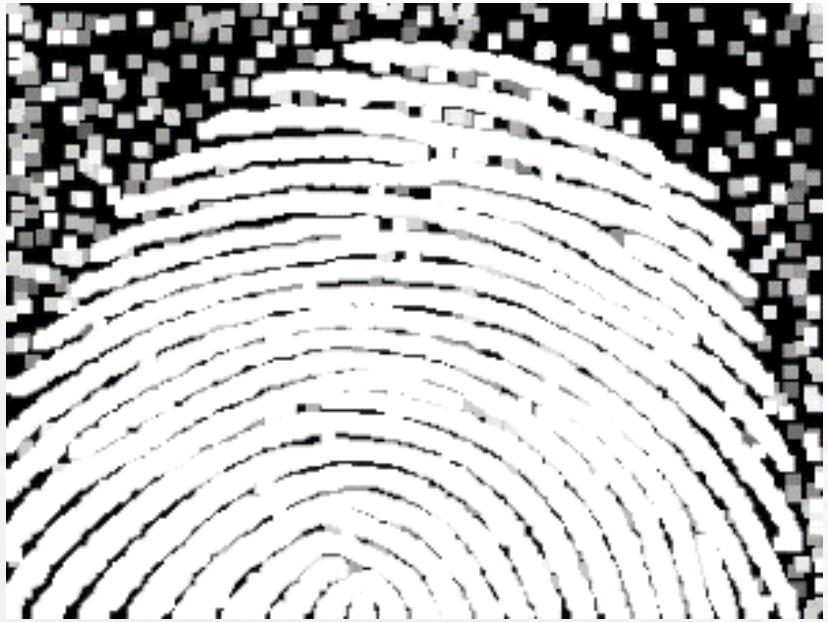
DILATION

EROSION

OPENING AND CLOSING

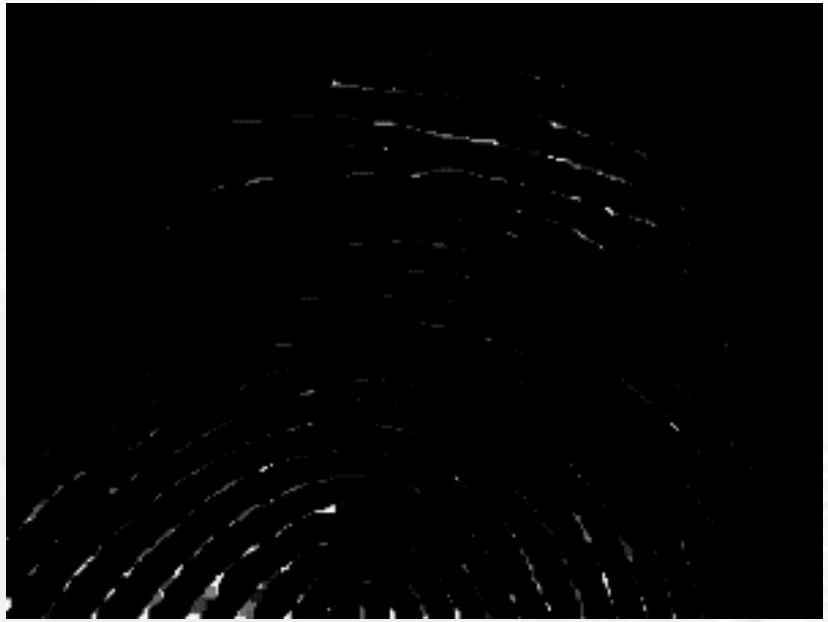
DILATION

cv2.dilate(image, kernel, iterations=1)



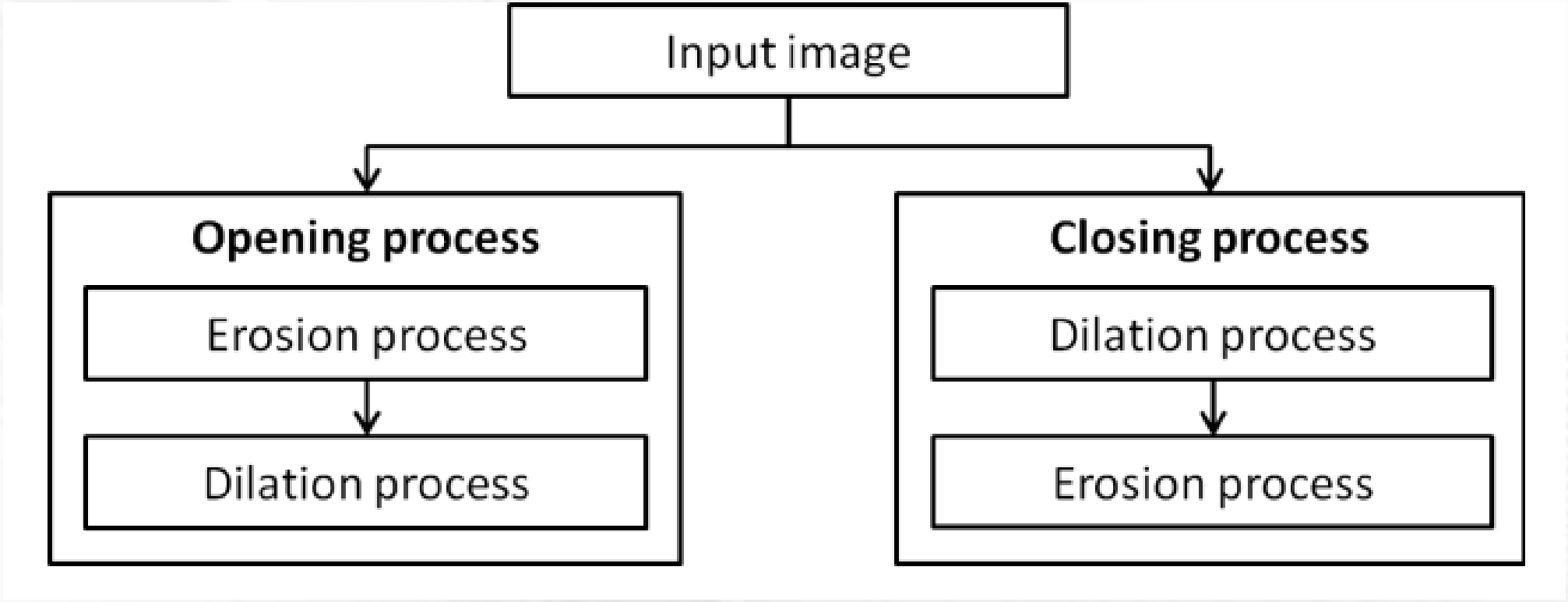
EROSION

cv2.erode(image, kernel, iterations=1)



OPENING AND CLOSING

cv2.morphologyEx(image, cv2.MORPH_OPEN, kernel)
cv2.morphologyEx(image, cv2.MORPH_CLOSE, kernel)



CONTOUR DETECTION

cv2.findContours(binary_image, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

08 PRACTICAL EXAMPLES

How do you load an image in grayscale and display it using OpenCV?

HINT

How can you resize an image to 200x200 pixels?

HINT

How do you convert a colored image to a binary image using a threshold value of 127?

HINT

How can you detect edges in an image using the Canny edge detector?

HINT

How do you detect and draw contours on a binary image?

HINT

TECH BOOTCAMP

DAY 06

INTRODUCTION TO IMAGE PROCESSING AND TOOLS

THANK YOU!