Day 4: Images as NumPy Arrays

Outcomes:

- Understand images as NumPy arrays in OpenCV
- Read and display images using OpenCV
- Understand image shape, channels, and pixel values
- Differentiate between grayscale and color images

What is an image (for OpenCV)

(refer to day 1)

For OpenCV, image is just a NumPy array.

rows → height

columns → width

values → pixel information

Shape of the NumPy array (for color images):

(Machine view)

```
(height, width, 3)
```

Reading and Image

```
import cv2

img = cv2.imread(“image.jpg”)
```

- `imread()` takes a path to the image as the input
- `img` is now a NumPy array
- So, when you load an image, you are loading data, not a “picture”

Displaying an Image

```
cv2.imshow(“image”, img)
```

- `imshow()` displays the array as an image

Image Shape

```
img.shape
```

Output:

```
(height, width, channels)
```

- `img` is a NumPy array
- `.shape` is a property of a NumPy array
- Shape is a tuple which stores the dimensions of the array

Channels

- OpenCV uses BGR not RGB
- So, a pixel looks like:

```
[blue, green, red]
```

- Where each value varies from 0 – 255

Pixel values

Access a pixel

```
img[x, y]
```

Output:

```
[34, 123, 255]
```

Grayscale Images

Grayscale images only have one value.

```
gray = cv2.imread("image.jpg", cv2.IMREAD_GRAYSCALE)
```

Shape:

```
(height, width)
```

Pixel value:

```
0 → black

255 → white
```

Key connection

- Video frame = image
- Image = NumPy array
- Motion tracking = changes between arrays