Day79_Image_Processing_with_OpenCV_Basic

September 3, 2025

1 Image Processing with OpenCV

This notebook is a documentation of yesterday's learnig.
We will use **OpenCV (cv2)** and **NumPy** to perform simple image operations.

What we are going to do:

1. Import required libraries
2. Read an image from disk
3. Display the image in a new window
4. Check image properties (height, width, channels)
5. Save the image to disk
6. Print the current working directory

2 Simple Explanation — What We Did and Why?

1. **Read the Image** → We used `cv2.imread()` to load a picture (cat image).
   - Why? So that the computer can understand the image as numbers (pixels).
2. **Show the Image** → We used `cv2.imshow("Elephant", img)` to open the image in a new window.
   - Why? To actually see what we loaded.
   - We gave the window name **"Elephant"**, but the picture was actually a **cat** .
   - You can give *any name* to the window.
3. **Check Image Size** → We printed the height and width using `img.shape`.
   - Why? To know how many pixels the image has (its resolution).
4. **Save the Image** → We used `cv2.imwrite("output.jpg", img)` to save the image.
   - Why? To keep a copy of the processed image with a new name.
5. **Check Working Directory** → We used `os.getcwd()` to see where the file was saved.
   - Why? So we know the folder where our image is stored.

   In short: We learned how to **load, show, check, save, and locate images** using OpenCV.

3 Import libraries

[1]:
```
import cv2
import numpy as np
import os
```

3.1 Read an image

We use `cv2.imread(path)` to read an image.

- The function loads the image as a NumPy array.
- Each pixel has 3 values (B, G, R).
- If the path is wrong, it returns `None`.

[8]:
```
# Update path to your image
img = cv2.imread(r"C:\Users\Lenovo\Downloads\Test_Image.jpg")

# Check what we got
print(img)
```

```
[[[133  62 212]
  [133  62 212]
  [133  62 212]
  …
  [156  81 227]
  [156  81 227]
  [157  82 227]]

 [[133  62 212]
  [133  62 212]
  [133  62 213]
  …
  [156  81 227]
  [156  81 227]
  [156  81 226]]

 [[133  62 213]
  [133  62 213]
  [133  62 213]
  …
  [156  81 227]
  [156  81 227]
  [156  81 227]]

 …

 [[135  61 219]
```

```
 [135  62 218]
 [135  62 218]
 …
 [145  70 228]
 [146  71 229]
 [146  72 228]]

[[135  61 219]
 [135  61 219]
 [135  62 218]
 …
 [146  71 229]
 [147  72 230]
 [147  73 229]]

[[136  62 220]
 [136  62 220]
 [136  62 220]
 …
 [147  72 230]
 [147  73 229]
 [148  74 230]]]
```

The output is a **NumPy array** with pixel values ranging from 0–255.
Each element represents `[Blue, Green, Red]` intensity of a pixel.

4 Show image in a window

We use:

- `cv2.imshow("window_name", img)` → to open a new window and show image
- `cv2.waitKey()` → waits for a key press (otherwise window closes immediately)
- `cv2.destroyAllWindows()` → closes the window

[9]:
```
cv2.imshow('Elephant', img)
cv2.waitKey()
cv2.destroyAllWindows()
```

5 Image Properties

We can get image dimensions using `img.shape`:

- `img.shape[0]` → Height (rows, number of pixels vertically)
- `img.shape[1]` → Width (columns, number of pixels horizontally)

- `img.shape[2]` → Channels (3 for color: B, G, R)

[10]:
```
print('Height of Image:', int(img.shape[0]), 'pixels')
print('Width of Image:', int(img.shape[1]), 'pixels')
```

```
Height of Image: 240 pixels
Width of Image: 204 pixels
```

6 Save the image

We use `cv2.imwrite("output.jpg", img)` to save the image.
This creates a copy of the image in the current directory.

[11]:
```
cv2.imwrite('Op1.PNG', img)
```

[11]:
```
True
```

7 Check Current Working Directory

We use Python's `os` module.

- `os.getcwd()` → shows where the file will be saved.

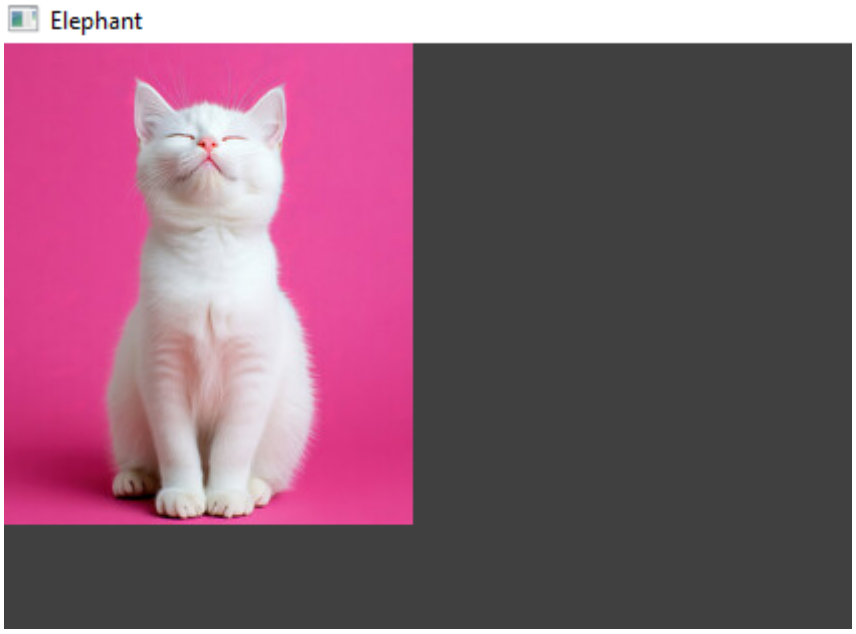
[12]:
```
current_dir = os.getcwd()
print("Current Working Directory:", current_dir)
```

```
Current Working Directory: C:\Users\Lenovo\OneDrive\Desktop\Python Everyday
work\Github work\DL
```

8 Displaying the Saved Output Image

We saved our image with a custom window name **"Elephant"**,
but in reality, the picture is of a **Cat** .

Here is the saved output image (`Op1.PNG`):



9 Summary

In this notebook, we learned how to:

- Import OpenCV and NumPy
- Read an image using `cv2.imread()`
- Display the image using `cv2.imshow()`
- Get image properties with `.shape`
- Save an image using `cv2.imwrite()`
- Check current working directory with `os.getcwd()`