Introduction to Images

Wednesday, May 31, 2023 6:19 PM

[Code Demonstration before PIL]

£ If the images are stored as numpy ndarrays, then why do we care about numerical feature Extraction?

£ Given multiple images how do we put them in dataframe?

↳ Sol: Convert the numpy nd array to 1 d array using flatten.

[ ]$_{3\times2}$ → flatten → [ ] 6 elements

So for an image of 1400 × 1400 × 3 = 5 8,80,000 elements

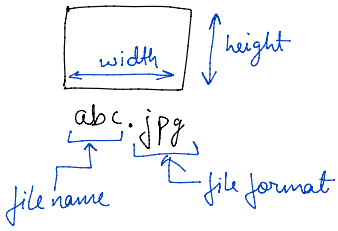
* CNN will focus on reducing dimensionality → this no. So that we dont run into curse of

---

$ Working with Images

{ nltk → Text
PIL → Images }

Image Storage format
- PNG
- JPG
- JPEG etc

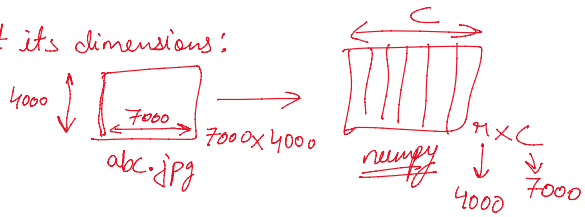


Dimensions / Shape of an image → width × height

{ Generally if
W > H ⇒ Landscape image
H > W ⇒ Portrait Image }

$ In general when we have an image when we talk about its dimensions:

Eg 7000 (width) × 4000 (height)



but when representing in numpy array → 4000 (rows) × 7000 (columns)

[Show img properties + code]
PIL

PIL → Image object → from PIL import Image
img = Image.open(path)
- Size → W×h
- format → JPEG/
- mode → RGB
- filename →

Ways to display img.
- img.show() #PIL
  ↳ To open image in new window
- plt.imshow(img) #matplotlib
  ↳ To render img in Jupyter

↓ wxh

↓ JPEG/PNG

↓ RGB

filename ↓ Path

↳ To render img in Jupyter

Aspect ratio = $\frac{img.size[0]}{img.size[1]}$

[Show code]

ways to segregate channels

- numpy
  - `img[:,:,0]`
  - `img[:,:,1]`
  - `img[:,:,2]`
- PIL `r, g, b = img.split()`