

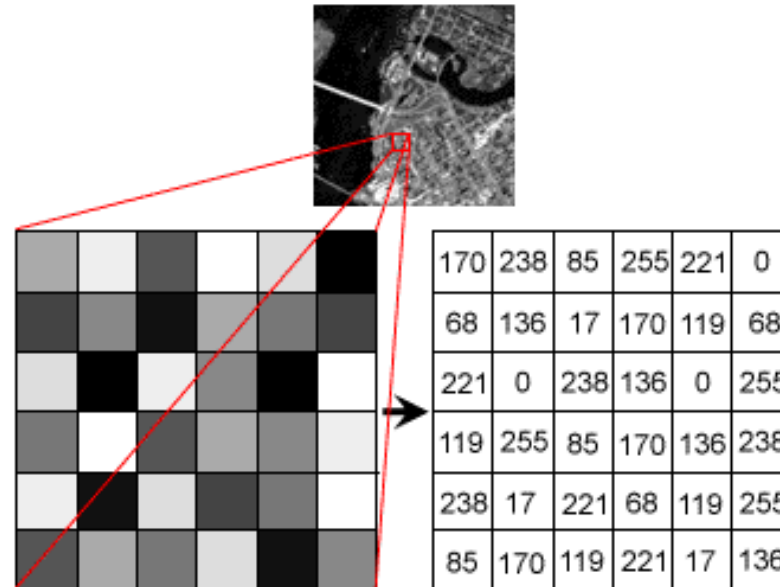


IMAGES 101

—◆—
A crash course in image processing

Images can be represented as functions $f(x, y)$

For a grayscale image, each pixel would have an intensity between 0 and 255, with 0 being black and 255 being white.



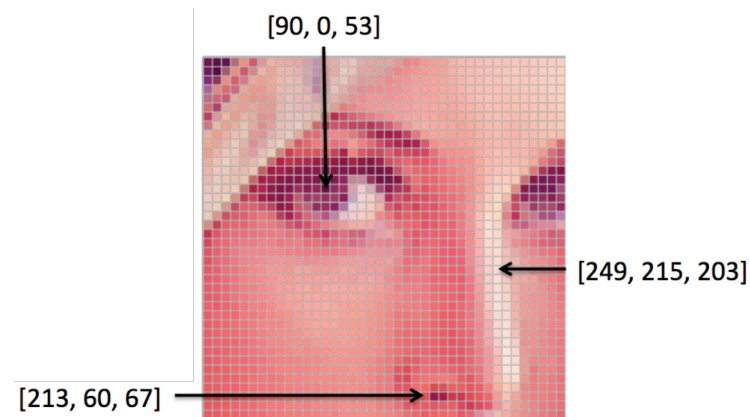
Images can be represented as functions $f(x, y)$

A color image is just a simple extension of this.

Using an RGB image as an example, the colors are constructed from a combination of Red, Green, and Blue (RGB).

Therefore, each pixel of the image has three channels and is represented as a 1x3 vector.

$$f(x, y) = \begin{bmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{bmatrix}$$





A Python Imaging Library that adds image processing capabilities to your Python interpreter.

<https://pillow.readthedocs.io/en/stable/index.html>

Other (more advanced) alternatives for those who are interested:

- OpenCV: <https://opencv.org>
- scikit-image: <https://scikit-image.org>

Pro tip: start with small images!

Debugging images can be hard, so using a smaller image can be helpful if you need to debug.

So here is my go-to image (you can find it in the repo): 🐶

cleffa.png

dimensions: 28×29



```
cleffa_grayscale = cleffa.convert("L")
```



(29, 28, 3)



(29, 28)