NumPy and Image Basics

PIERIAN DATA

Computer Vision with Python

- Section Goals
  - Understand how to work with the basics of NumPy
  - Understand how to create arrays
  - Slice and index elements from arrays
  - Open and display images with NumPy

Let’s get started!

What is an image?

Computer Vision with Python

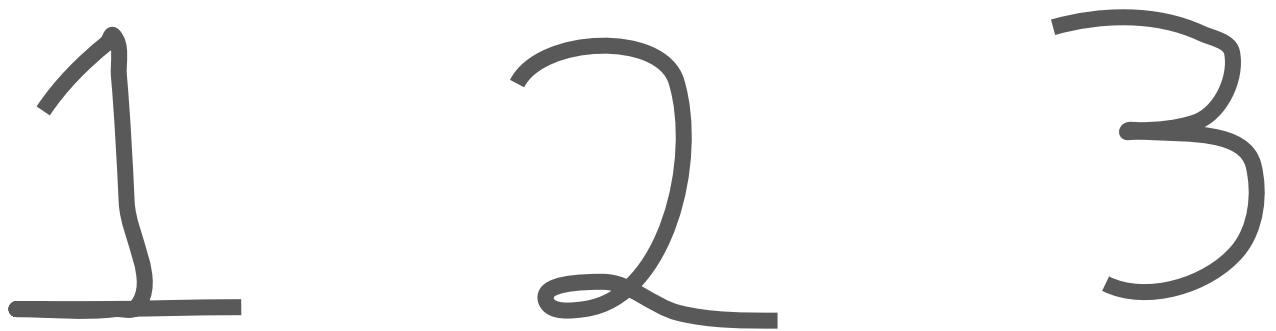
- Before we talk about how to use numpy with image files, let’s first discuss how computers handle images.
- Let’s imagine we wanted to build software that could sort out mail based on the zip code of the address.

Computer Vision with Python

- Often mail is handwritten, which means we would need to begin to understand how to use computer vision to actual read in image data of these handwritten numbers.
- How does a computer represent image data?

Computer Vision with Python

- Let’s imagine we have a simple image of a handwritten number:



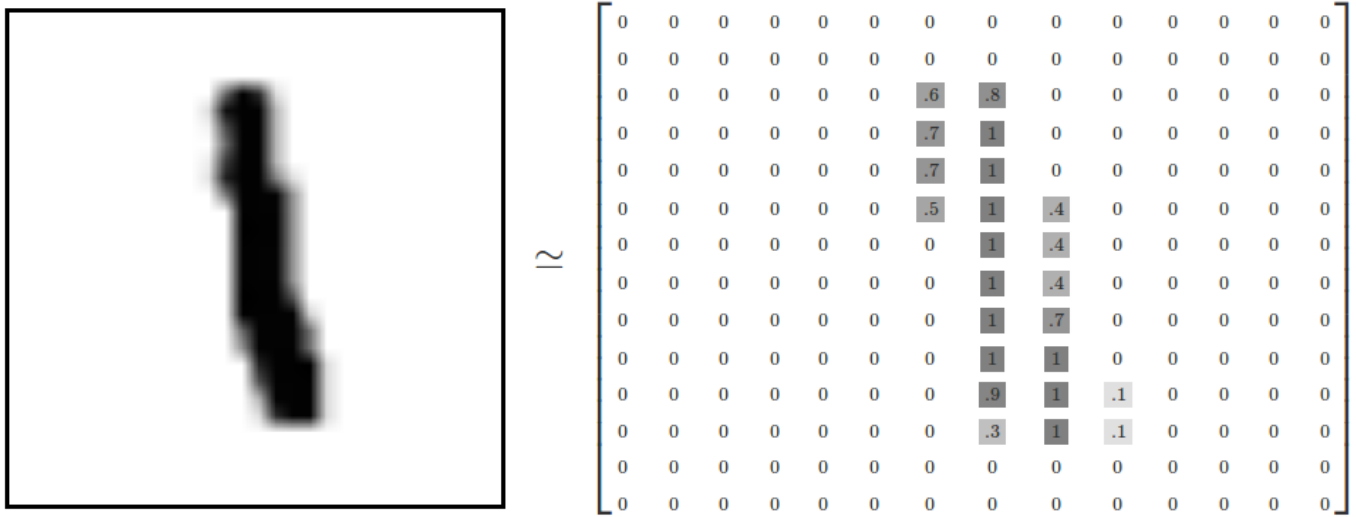
Computer Vision with Python

- Each single digit image can be represented as an array

PIERIAN DATA

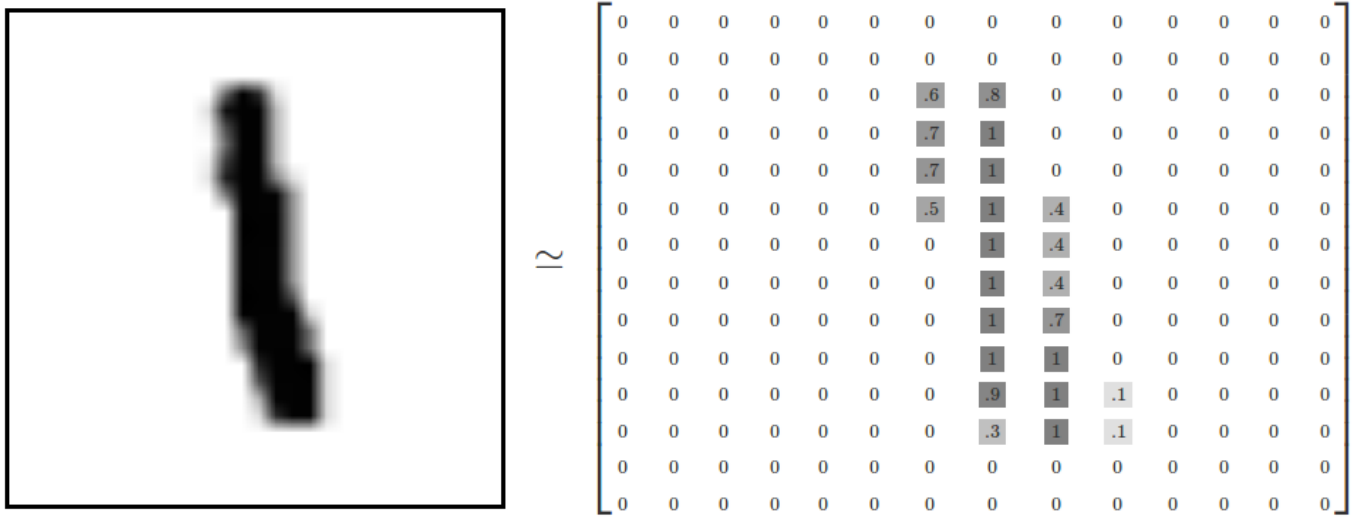
Computer Vision with Python

- For example, here a number is 28 by 28 pixels



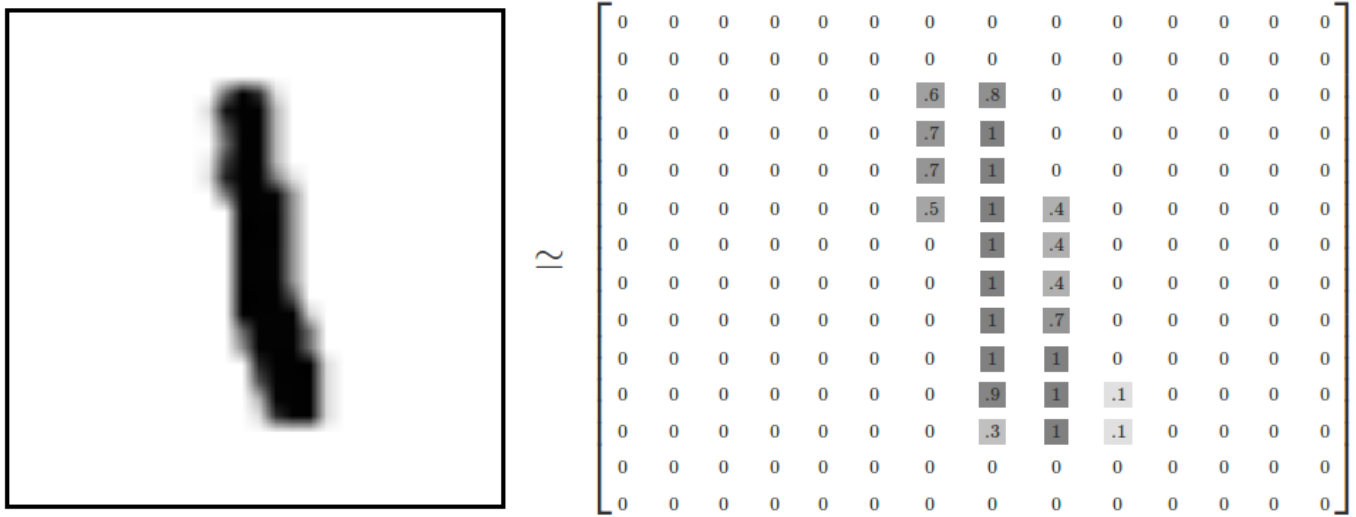
Computer Vision with Python

- Then how dark a pixel should be can be represented as a value between 0 and 1.



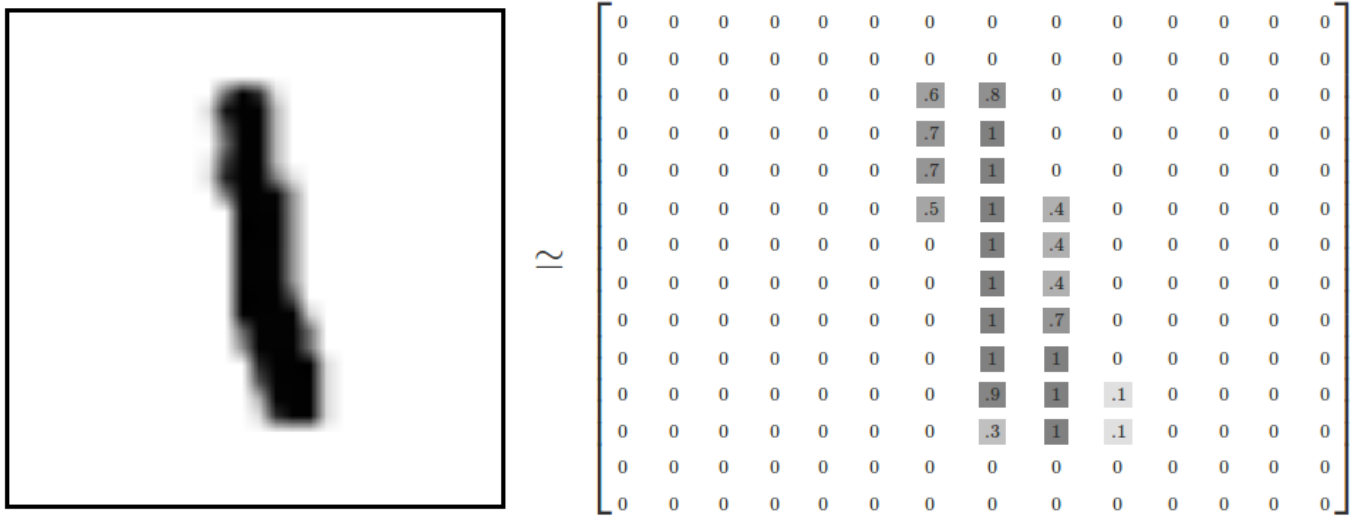
Computer Vision with Python

- Often the default images have values between 0 and 255



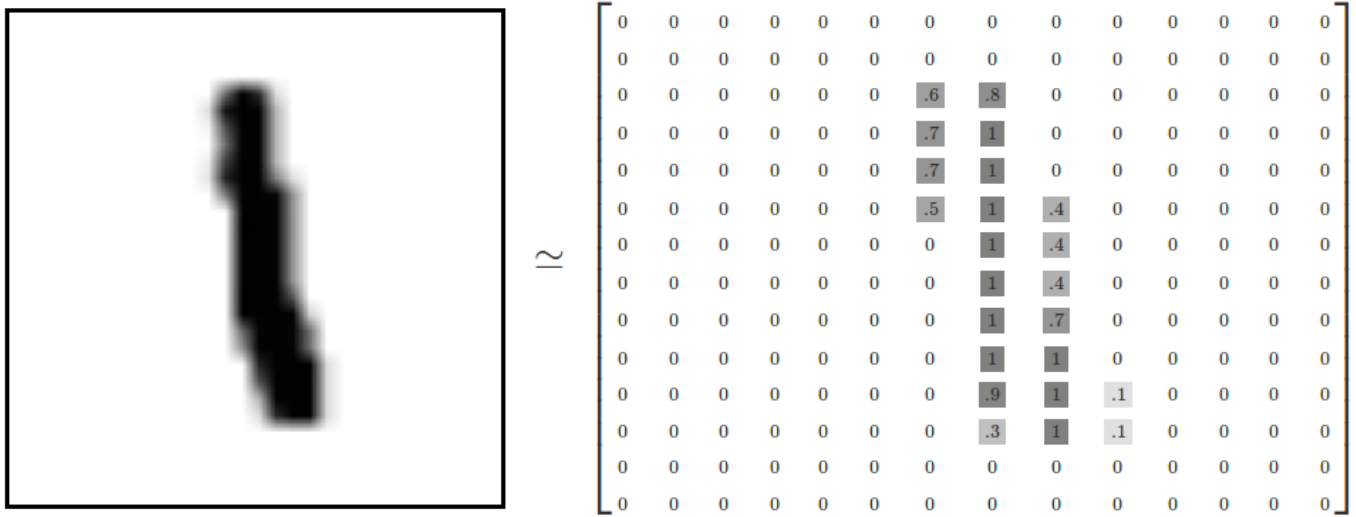
Computer Vision with Python

- The range 0 to 255 has to do with how computers store 8-bit numbers.



Computer Vision with Python

- But you can always divide all the values by 255 to normalize to between 0 and 1

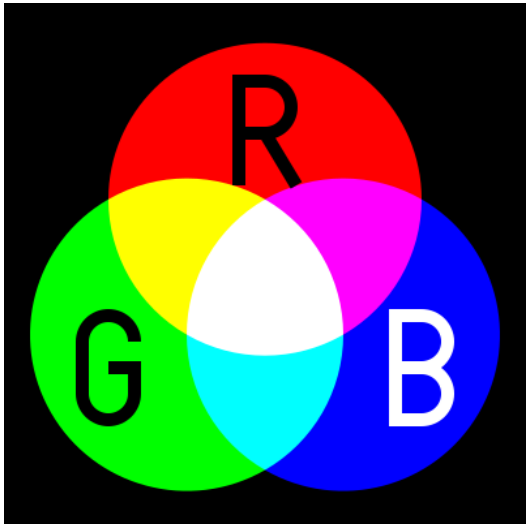


Computer Vision with Python

- Now that we've understood how grayscale images can be represented as arrays, what about color images?
- Color images can be represented as a combination of Red, Green, and Blue.

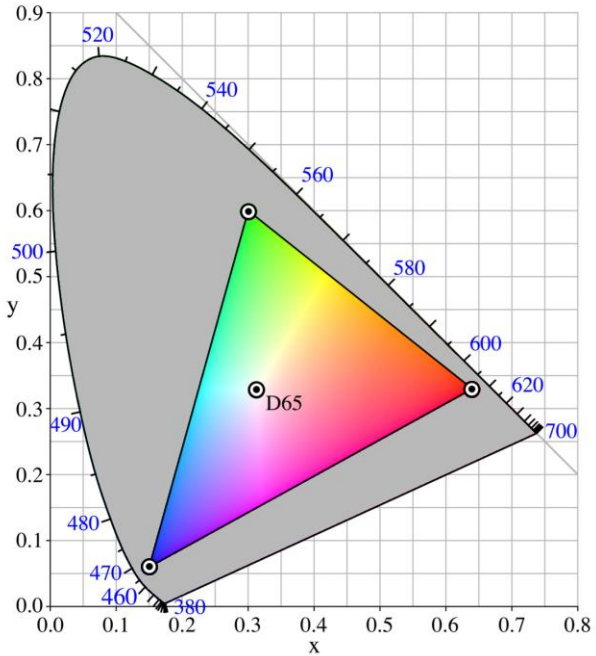
Computer Vision with Python

- Additive color mixing allows us to represent a wide variety of colors by simply combining different amounts or R, G, B.



Computer Vision with Python

- RGB allows to produce a range of colors



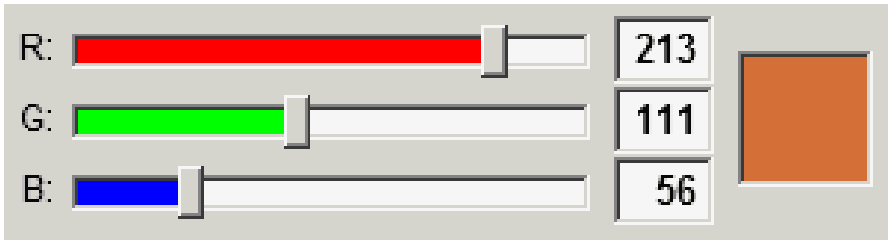
PIERIAN DATA

Computer Vision with Python

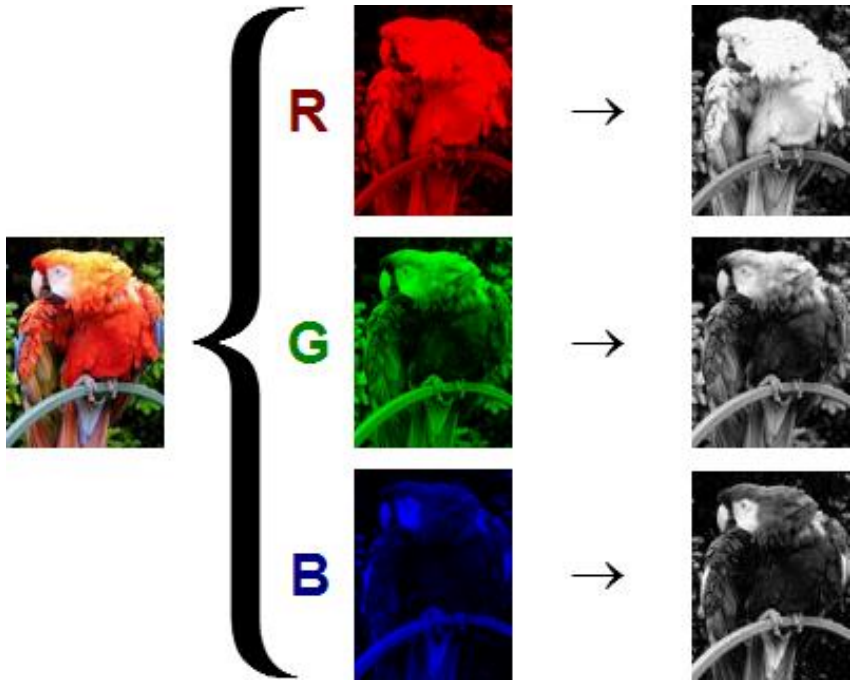
- Later on in the course we will learn about alternative color mappings that can be applied to images.

Computer Vision with Python

- Each color channel will have intensity values.
- You may have already seen this sort of representation in other software with RGB sliders.



Computer Vision with Python



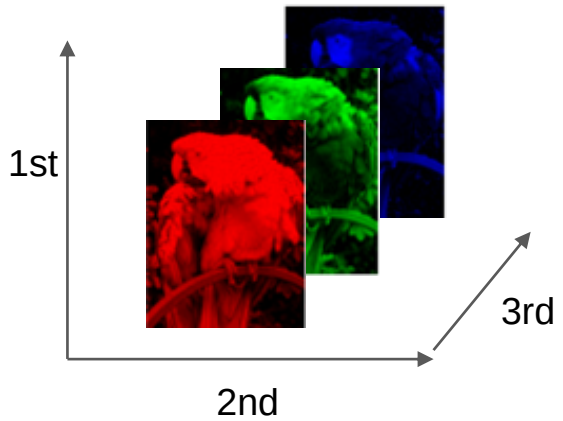
- The shape of the color array then has 3 dimensions.
- Height
- Width
- Color Channels

Computer Vision with Python

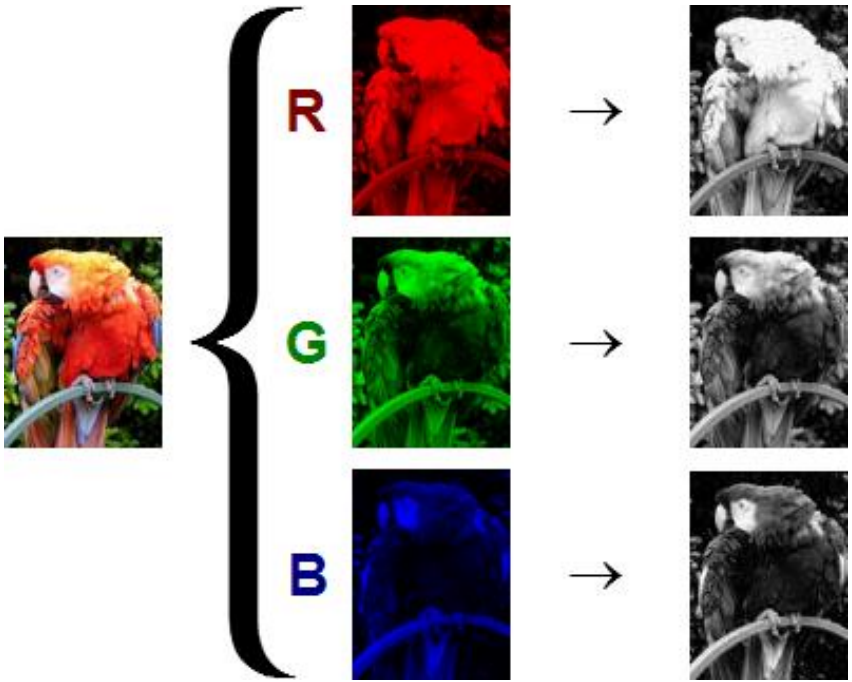
- This means when you read in an image and check its shape, it will look something like:
  - **(1280,720,3)**
  - **1280** pixel width
  - **720** pixel height
  - **3** color channels

Computer Vision with Python

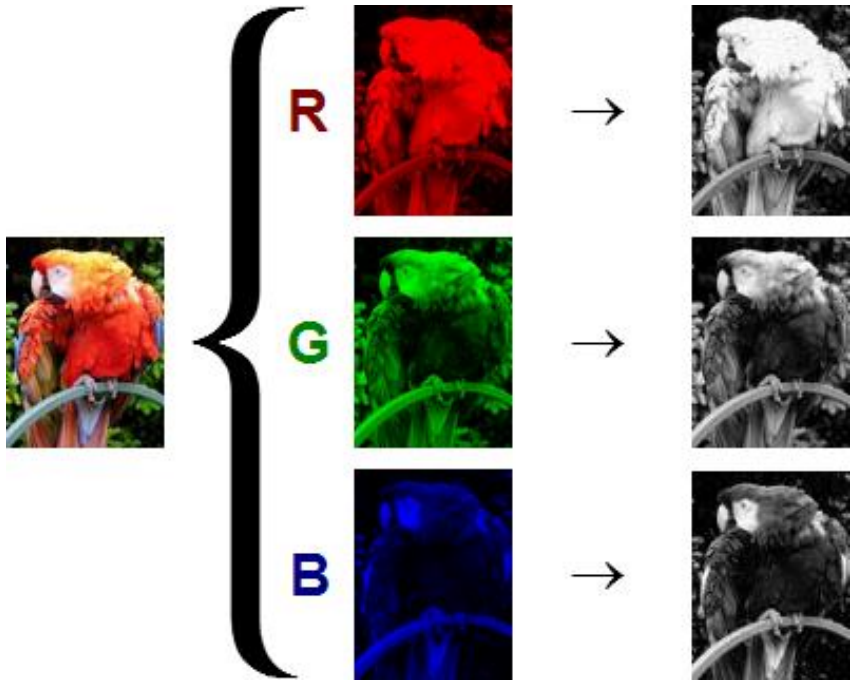
- This means when you read in an image and check its shape, it will look something like:
  - **(720,1280,3)**
    - **720** pixel height
    - **1280** pixel width
    - **3** color channels



Computer Vision with Python



- Keep in mind the computer won't "know" a channel is Red, it just knows that there are now 3 intensity channels.



Computer Vision with Python

- The user needs to dictate which channel is for which color.
- Each channel alone is essentially the same as a grayscale image.

Computer Vision with Python

- Let’s explore this further with Numpy and Python!
- I encourage you to also check out the Wikipedia article on RGB color channeling for more interesting details!

NumPy Arrays

Images and NumPy

Numpy and Images Assessment Overview

PIERIAN DATA

Numpy and Images Assessment Solutions

PIERIAN DATA