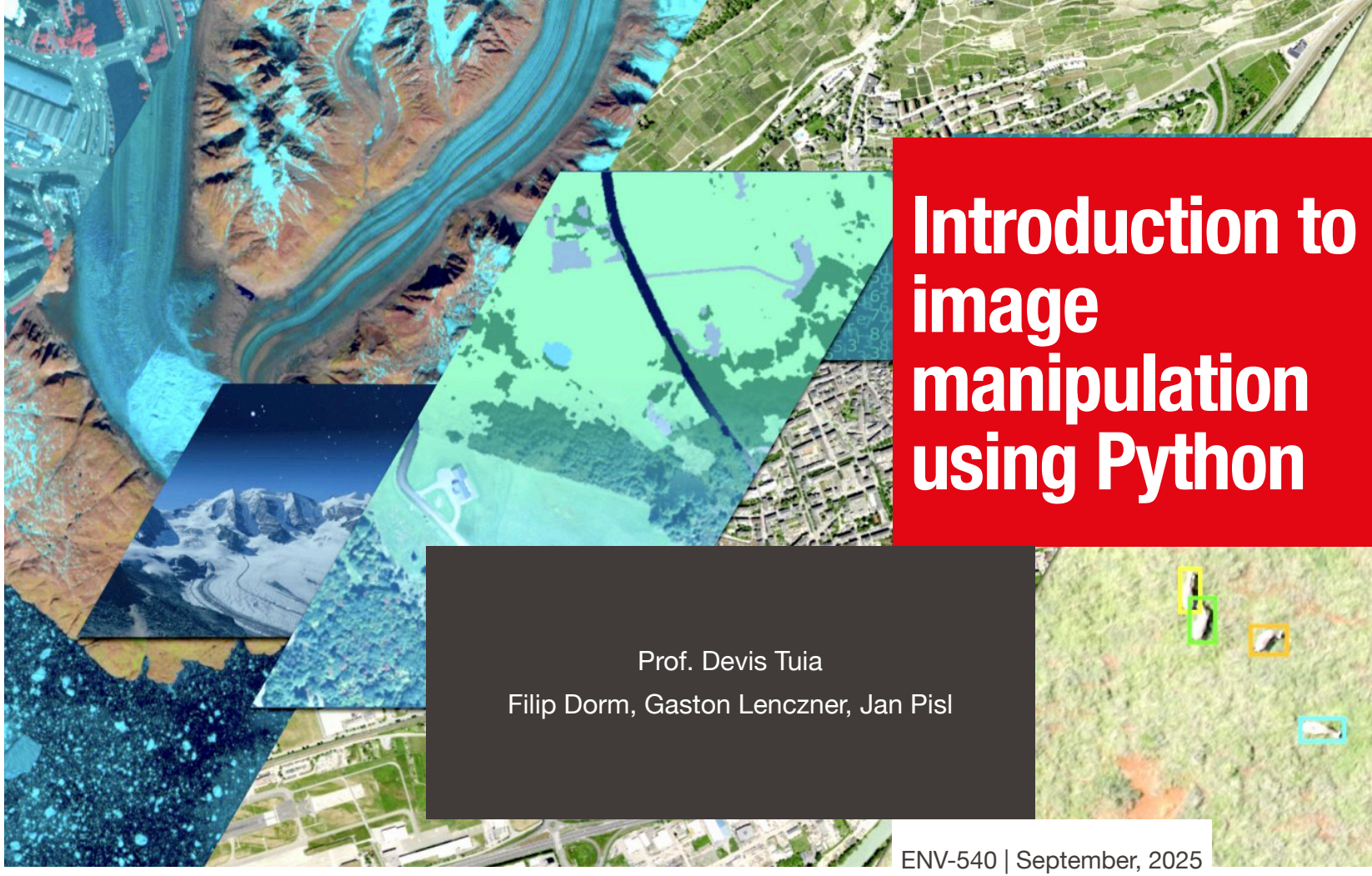


Introduction to image manipulation using Python

Prof. Devis Tuia

Filip Dorm, Gaston Lenczner, Jan Pisl

ENV-540 | September, 2025

Python

Python is a general-purpose programming language that is widely used for scientific computing. See this tutorial to learn more about Python. We will use Jupyter notebooks for the exercises.

```
####### Data types #######
x = 2
print(x, type(x))              # Integer
y = 2.5
print(y, type(y))              # Float
s = "hello"
print(s, type(s))              # String
t, f = True, False
print(t, type(t), f, type(f))# Boolean
```

```
2 <class 'int'>
2.5 <class 'float'>
hello <class 'str'>
True <class 'bool'> False <class 'bool'>
```

```
####### Basic operations #######
print(x + 2)         # Addition
print(x - 2)         # Subtraction
print(x * 2)         # Multiplication
print(x / 2)         # Division
print(s + " world") # String concatenation
print(t and f)       # Logical AND;
print(t or f)        # Logical OR;
```

```
4
0
4
1.0
hello world
False
True
```

Python

Python has several container types. Two of the most widely used are Lists and Dictionaries

```
####### Lists #######
# In python indices start in 0
numbers = [10, 30, 20, 60]
print(numbers[1])       # Second element
numbers[0] = 50         # Change a value
print(numbers)
numbers.append(40)      # Add an element
print(numbers)
print(numbers[1:4])     # Select 2nd to 4th
```

```
30
[50, 30, 20, 60]
[50, 30, 20, 60, 40]
[30, 20, 60]
```

```
####### Dictionaries #######
data = {"red" : 0, "green": 1, "blue": 2}
print(data['green'])   # Get entry value
print("green" in data) # Verify if exist
data["violet"] = 5     # Add new entry
print(data)
data["blue"] = 3       # Change value
print(data)
```

```
1
True
{'red': 0, 'green': 1, 'blue': 2, 'violet': 5}
{'red': 0, 'green': 1, 'blue': 3, 'violet': 5}
```

Python functions


```
####### Functions #######
def select_positive_numbers(numbers):
    selected_numbers = []
    for number in numbers:
        if number > 0:
            selected_numbers.append(number)
    return selected_numbers

# Define input list
input_numbers = [2, 3, -4, 6, -10, 8]
# Calling a function
positives = select_positive_numbers(input_numbers)
print(positives)
```

```
[2, 3, 6, 8]
```

Numpy

Numpy is a scientific computing library for high-performance operations of multidimensional arrays. A Numpy array is a list/array of values of the same type.

```
####### Numpy arrays #######
array_1d = np.array([5, 6, 7, 8])          # 1D array
print(array_1d[0])                         # Access first elements

array_2d = np.array([[1, 2, 3],[4, 5, 6]])# 2D array
print(array_2d.shape)                      # 2D array Shape (rows, columns)
print(array_2d[1, 0])                      # Access elem in the 2dn row, 1st column

print(array_2d[1, 1:3])                    # Select 2nd and 3rd element of the 2nd row
# or -> print(array_2d[1, 1:])
```

```
5
(2, 3)
4
[5 6]
```

Note: You can use the basic operations +, -, *, / for element-wise operations

Images are basically arrays of numbers

We can visualize images using the ***matplotlib*** library

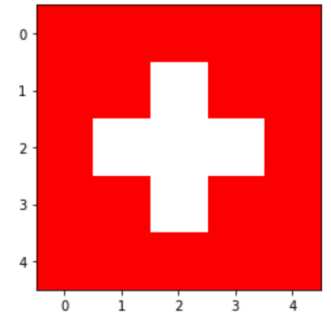
```
import matplotlib.pyplot as plt

red_band = np.array([[255, 255, 255, 255, 255],
                     [255, 255, 255, 255, 255],
                     [255, 255, 255, 255, 255],
                     [255, 255, 255, 255, 255],
                     [255, 255, 255, 255, 255]])

green_band=np.array([[  0,   0,   0,   0,   0],
                     [  0,   0, 255,   0,   0],
                     [  0, 255, 255, 255,   0],
                     [  0,   0, 255,   0,   0],
                     [  0,   0,   0,   0,   0]])

blue_band =np.array([[  0,   0,   0,   0,   0],
                     [  0,   0, 255,   0,   0],
                     [  0, 255, 255, 255,   0],
                     [  0,   0, 255,   0,   0],
                     [  0,   0,   0,   0,   0]])

image = np.stack([red_band, green_band, blue_band])# Shape: (3, 5, 5)
image = image.transpose(1, 2, 0)                   # Shape: (5, 5, 3)
image = image.astype(np.uint8)                     # Transform data type
plt.imshow(image)                                  # Show image
```



Reading Sentinel-2 images using skimage

For the exercises we will use Sentinel-2 imagery (available for download in https://scihub.copernicus.eu/dhus)

```
from skimage.io import imread
band2 = imread("image_directory_path/band2.jp2") # Blue
band3 = imread("image_directory_path/band3.jp2") # Green
band4 = imread("image_directory_path/band4.jp2") # Red
```

```
def get_normalized_image(image, percentiles=(2, 98)):
    output = np.zeros_like(image)
    for k in range(image.shape[2]): # for each band
        p_min, p_max = np.percentile(image[:, :, k], percentiles)
        output[:, :, k] = exposure.rescale_intensity(image[:, :, k],
                                in_range=(p_min, p_max), out_range=(0, 255))
    return output.astype(np.uint8)
```

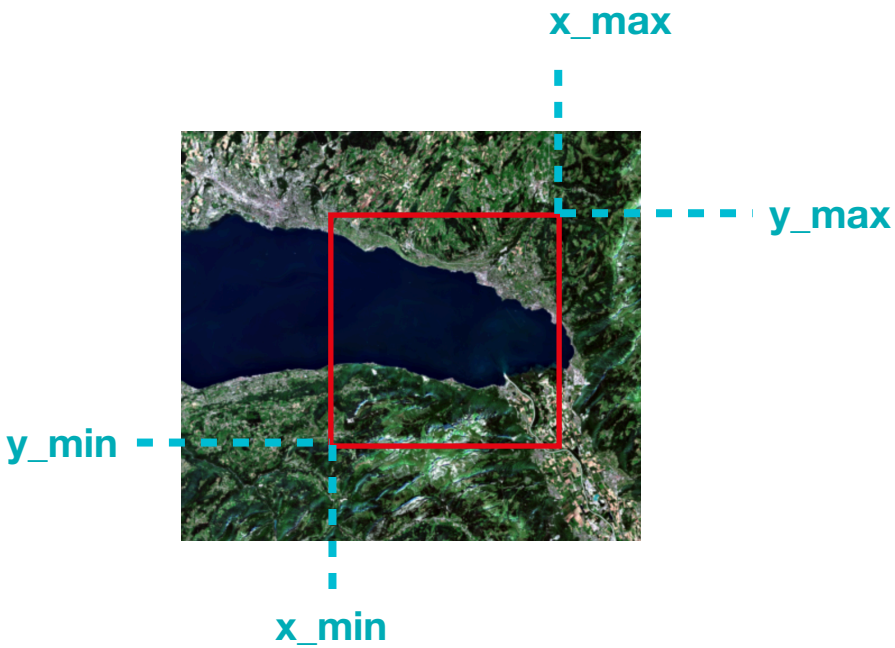
```
image = np.array([band4, band3, band2]) # Shape: (3, height, width)
image = image.transpose(1, 2, 0)        # Shape: (height, width, 3)
# Normalize image for visualization
normalized_image = get_normalized_image(image)
plt.imshow(normalized_image)            # show image
```

Unnormalized image

Normalized image

Cropping an image

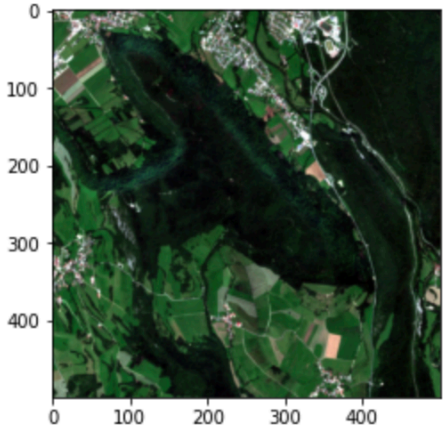
To crop an image we need to define the area of interest using a bounding box



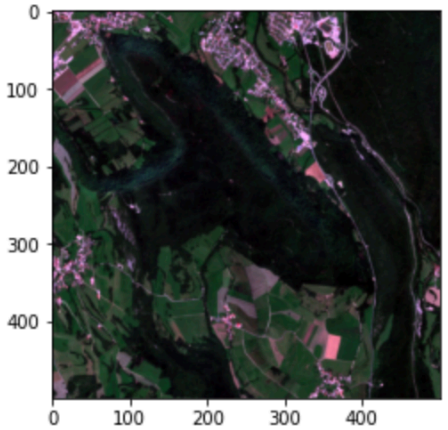
Examples: Manipulating images

After reading the images we can perform operations with the array representations

```
# Cropping part of the large image
# (using pixel coordinates)
image1 = normalized_image[2000:2500, 2000:2500]
plt.imshow(image1)
```



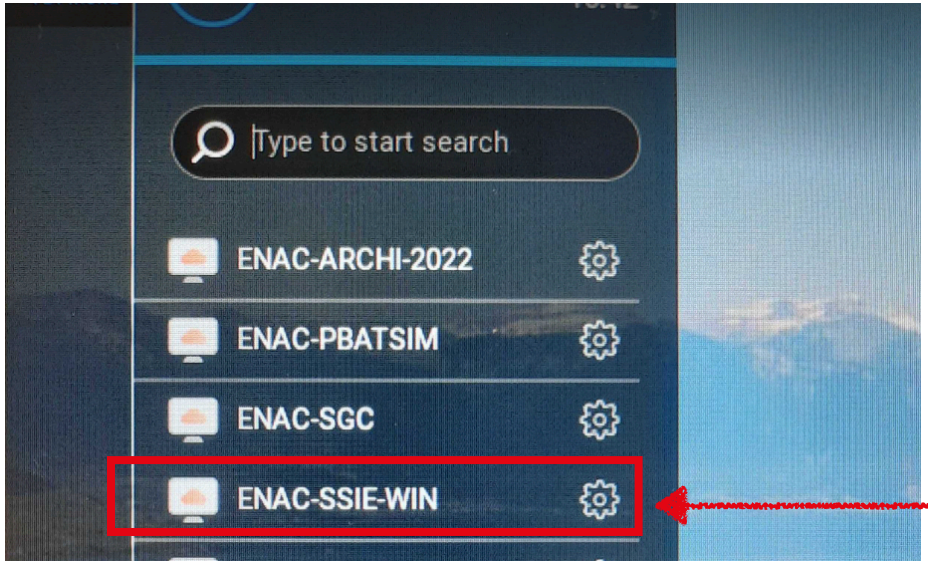

```
# Decreasing the green values of the image
image1[:, :, 1] = (image1[:, :, 1] * 0.7).astype(np.uint8)
plt.imshow(image1)
```



Exercise

- Read the provided PDF file and Jupyter Notebook with detailed instructions
- Tasks:
  1. Read the provided Sentinel-2 imagery
     (Band 2 : Blue, Band 3 : Green, Band 4 : Red, Band 8 : Near Infrared)
  2. Save in ***tif*** files the following geo-located multi-band images:
     - Natural color image
     - False color composite image
       - Near infrared → Red
       - Red → Green
       - Green → Blue
  3. Crop the images (natural colour and false colour composite) using the provided bounding boxes, and answer the questions about each cropped region

Working on the computers of the classroom



Access to “ENAC-SSIE-WIN”