CV2021 Exercises-1

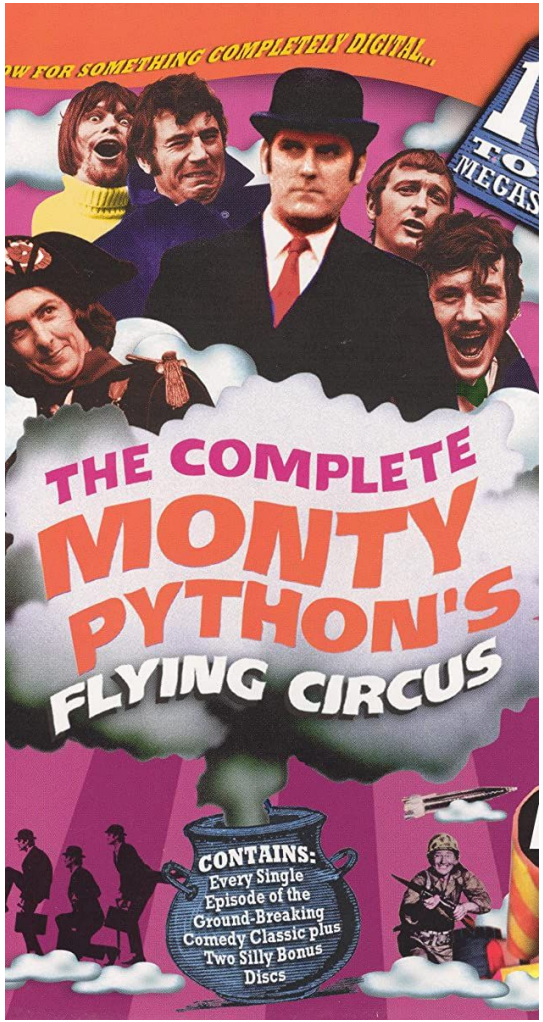
Florin-Alexandru Vasluianu
fvasluianu@student.ethz.ch

Outline

- Python elementary
- Numpy elementary
- Image manipulation in Python
- Image filtering in Python
- Q&A

Python

- Created by Guido van Rossum (‘91)
- Interpreted language
- Code readability and clear programming
- Efficient data structures
- Dynamic typing
- Effective approach to OOP
- Very useful in prototyping applications.



Install

- Windows installer can be downloaded from their official website
- Don't forget to append the python executable to $PATH on Windows
- **Pip** package manager can be used to add necessary packages to import from
- If working with multiple Python projects, **virtual environments** are recommended

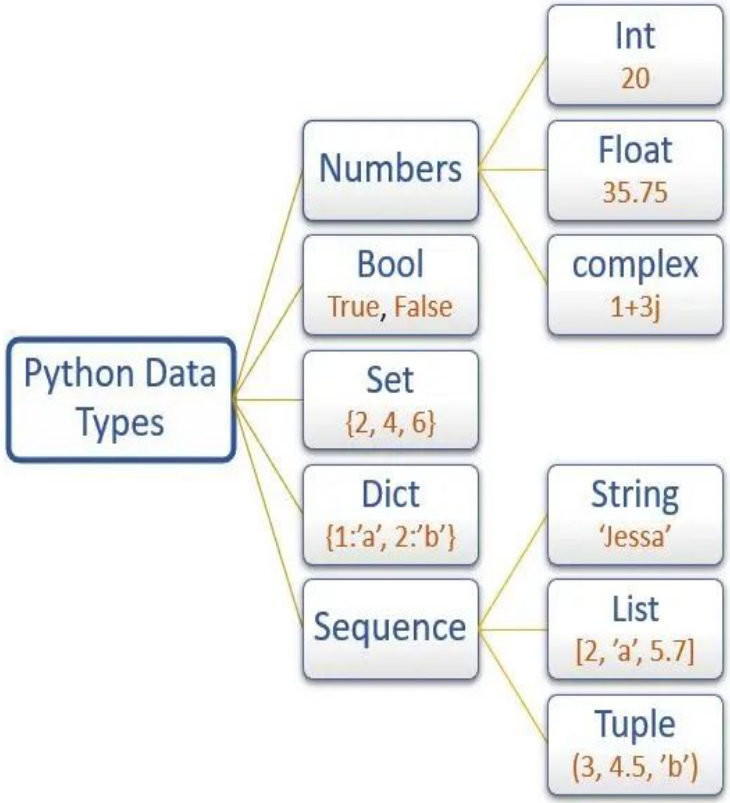

```
$ sudo apt-get update
$ sudo apt-get install python3.6
```

Data types

```
question = 'What is the answer?'
answer = 42
percent = 0.25
power_on = True
disabled = False
```

I/O operations

```
print(question, answer)
name = input('Your name:')
age = int(input('Your age:'))
```



Statements and iterables

- Python uses the **<tab>** character to interpret complex statements
- Python offers item based access to iterable data structures
- Index based iteration can be done using the **range** object
- The last example is called **list comprehension**, where the function **upper()** is called for each of the elements in the list

```
done = False
while not done:
    option = read_option()
    if option == 0:
        break
    elif option > 9:
        print('Valid options: 1..9')
        continue
    else:
        # do something
```

```
elements = ['some', 'items', 3.14, 42, True]
for item in elements:
    print(item)
```

```
fruits = ['Banana', 'Apple', 'Lime']
loud_fruits = [f.upper() for f in fruits]
print(loud_fruits)
```

Data structures

- Lists can contain different types of elements
- Tuples are immutable
- Sets consist of unique elements
- Dictionaries are sets of (key, value) pairs. Methods are provided to get the list of keys or the list of values.

```
list = ['this', 'is', 'the', 1]
set = {1, 3, 5, 7, 11}
dict = {'John': 24, 'Jane': 22}
tuple = (13.5, 7, 255)
```

```
d = {'John': 24, 'Jane': 22}
for key, value in d.items():
    print(key, value)
```

File I/O

```
f = open('data.txt', 'r')
data = f.read()
f.close()

with open('data.txt') as f:
    data = f.read()
# here f is automatically closed
```

Functions

- Are defined when the code is parsed;
- Can use default values arguments
- When default values are set, the parameters can be omitted at function call
- Can use named arguments, but when calling them , the named arguments have to be used after the directly specified parameters;

```
def add(a=42, b=0):
    return a + b

r = [add(1, 1), add(a=1, b=2),
add(b=3), add()]

print(r)
```

Exceptions

- Exceptions are raised during runtime
- Can be handled using try/except/finally blocks.

```
import sys

try:
    f = open('myfile.txt')
    s = f.readline()
    i = int(s.strip())
except OSError as err:
    print("OS error: {0}".format(err))
except ValueError:
    print("Could not convert data to an integer.")
except:
    print("Unexpected error:", sys.exc_info()[0])
    raise
```

numpy

- Library used to operate with arrays using Python
- Offers support for slicing/dicing operations
- Offers fast operations with SIMD support
- Very useful when manipulating objects represented in multiple dimensions (like an RGB image)
- Offers a pseudo-random numbers generator
- Data types are platform-dependent

```
import numpy as np
print(np.zeros((3, 4)))
print(np.ones((2, 3 , 4)))

numbers = [1, 2, 3, 4, 5, 6]
print(np.array(numbers))
arr_numbers = np.array(numbers)
print(arr_numbers[2:])
print(arr_numbers[:4])
print(arr_numbers[1:4])
```

```
A = np.array( [[1,1], [0,1]] )
B = np.array( [[2,0], [3,4]] )
print(A * B)
print(A @ B)
```

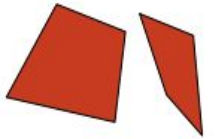
Unitary transforms

2D transformation hierarchy

A square transforms to:

Projective 8dof

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

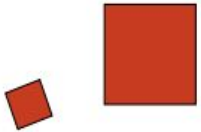


Affine 6dof

$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Similarity 4dof

$$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Euclidean 3dof

$$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

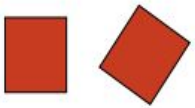
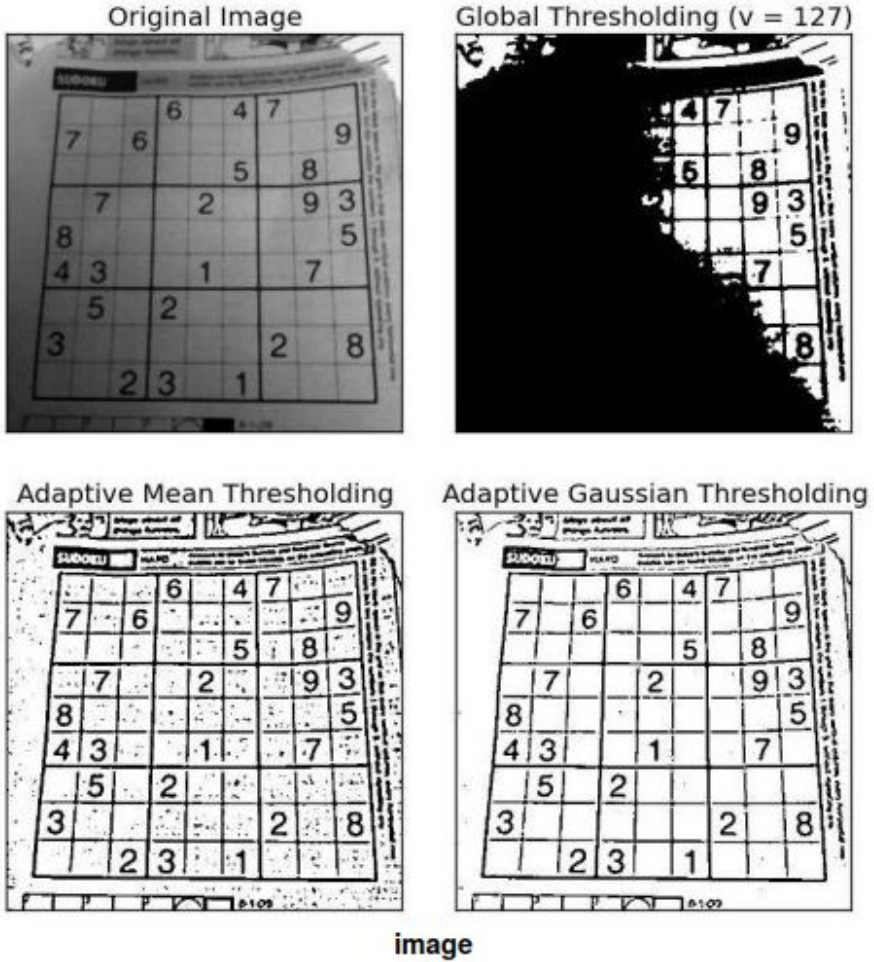


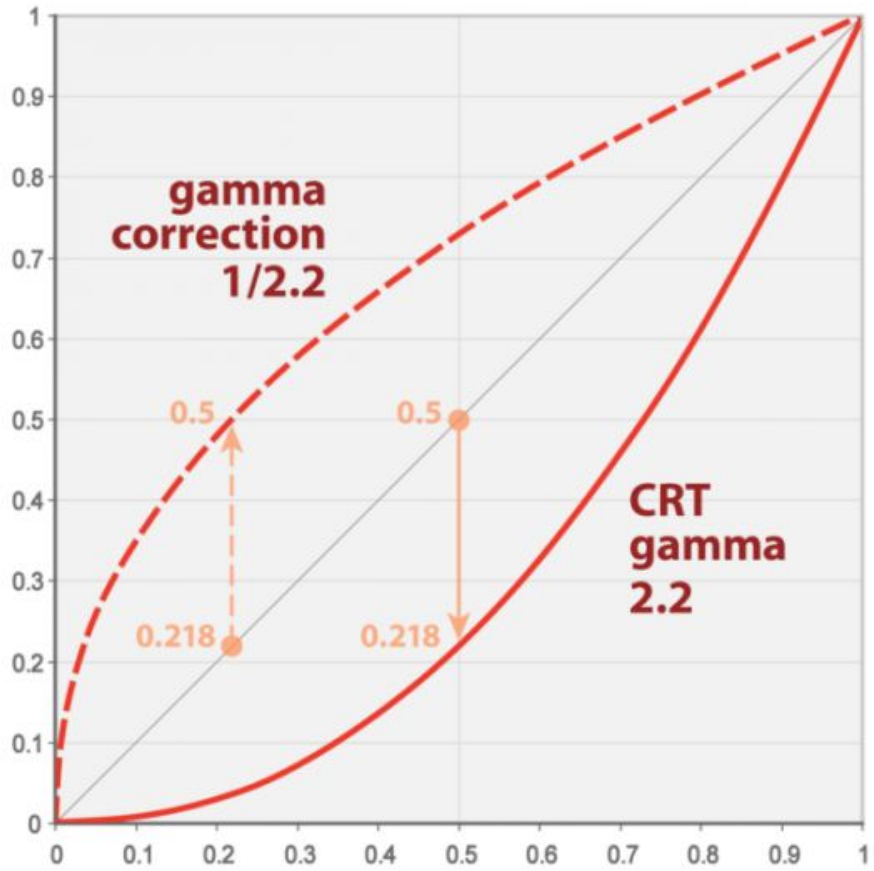
Image binarization

- Fixed threshold binarization;
- Adaptive mean binarization;
- Gaussian binarization



Gamma correction

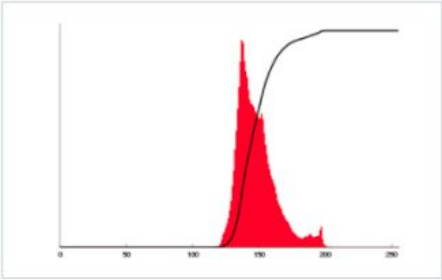
$$R_{corrected} = \left(\frac{R_{uncorrected}}{R_{max}}\right)^{\gamma} \times R_{max}$$



Histogram equalization

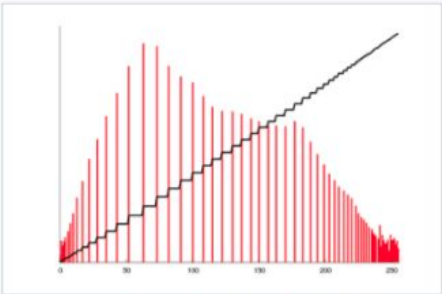
- A method of contrast enhancement;
- Useful with images having a “sharp” histogram
- Spreads the gray values proportionally to their probability of appearance

Before Histogram Equalization



Corresponding histogram (red) and cumulative histogram (black)

After Histogram Equalization



Corresponding histogram (red) and cumulative histogram (black)

Q&A

Have a look on the exercises left to you as homework!
For any questions, use the email on the first slide, until we set another communication channel.