National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Lab 3: Interpolation.

Objective:

The objective of this lab is:

- To understand and implement the rescale function from skimage library.
- To understand and implement the resize function from skimage library.
- To understand and implement 1d and 2d interpolation.

Tools:

- Python (jupyter notebook)
- Skimage, numpy, matplotlib

Lab Tasks and Description:

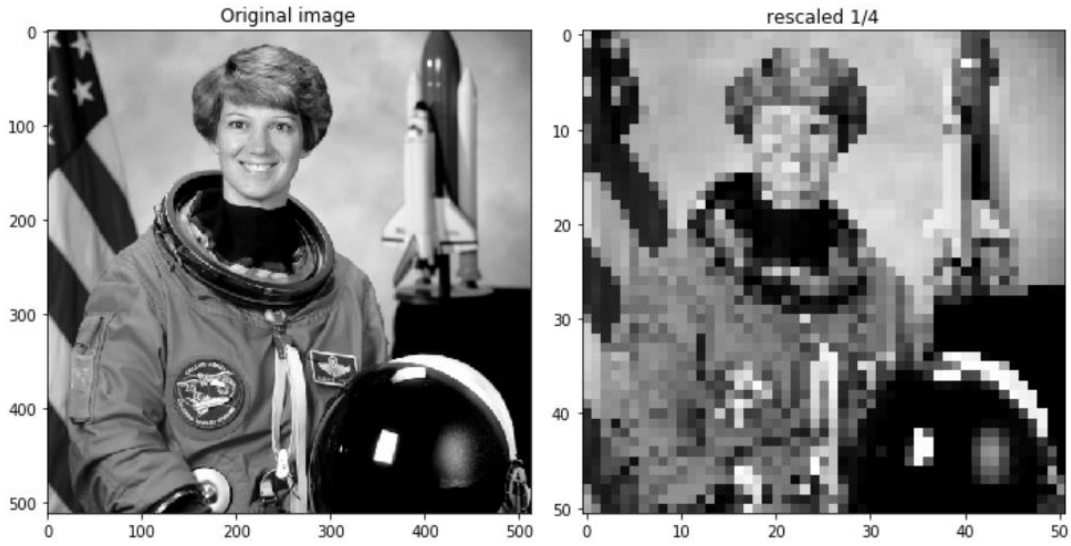
Task 1: Rescale an image

Rescale operation resizes an image by a given scaling factor. The scaling factor can either be a single floating point value, or multiple values - one along each axis.

- Firstly, you have to load an image using the following command in skimage library.

  image = color.rgb2gray(data.astronaut())

- In the next step, you have to rescale the actual image to one fourth the size of actual image
  - HINT: you have to play around with the parameters of rescale function from skimage library
- Summarize your findings with images in the following way.

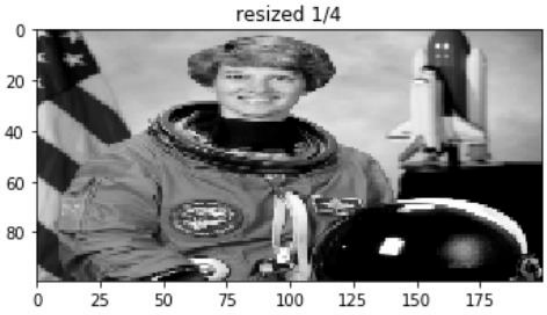


Task 2: Rescale an image using Resize function

Resize serves the same purpose, but allows to specify an output image shape instead of a scaling factor. Only integers are allowed as output image size

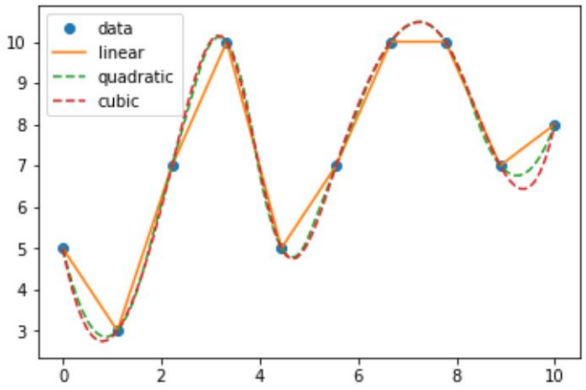
- Firstly, you have to load an image using the following command in skimage library.

  image = color.rgb2gray(data.astronaut())
- In the next step, you have to resize the actual image to the size (100,200) using resize function of skimage library.
- You are supposed to print the size of actual image and the rescaled image.
- Summarize your findings with images in the following way.



Task 3: 1D interpolation

- Given, the following data, you have to fit a line using the following three types of interpolation.
  1):- Linear
  2):- Quadratic
  3):- Cubic
  x = np.linspace(0, 10, num=10, endpoint=True)
  y = [5,3,7,10,5,7,10,10,7,8]
  Hint:- You may seek help from scipy.interpolate
- Once you have fitted the line, the next step is to generate new type of “x_new” , but it should comprise 100 points instead of 10 points.
- In the next step, you have to predict the “y” corresponding to the “x_new” using the lines fitted in the step 1.
- Summarize your findings with images in the following way.



Task 4: 2D interpolation

Two dimensional interpolation takes a series of (x,y,z) points and generates estimated values for z's at new (x,y) points. Interpolation is used when the function that generated the original (x,y,z) points is unknown.

- In the first step, your task is to generate any of the data in the form of (x,y,z)
- Then you have to create a rectangular grid out of an array of x values and an array of y values.
  Hint:- You may seek help from np.meshgrid() function.
- Now you have to compute the interpolation function "f"
  Hint:- You may seek help from scipy.interpolate.
- In the next step you have to generate the test data like this:-
  xnew = np.arange(-5.01, 5.01, 1e-2)
  ynew = np.arange(-5.01, 5.01, 1e-2)
- Once the test data has been generated, you have to pass it to the interpolation function "f" to generate "z".
- In the last step you have to plot your results.

Deliverables and submission guidelines:

- **You have to submit either a python notebook (with complete code of all the tasks and screenshots of output.**
- **Or, you may submit a lab report containing the code of all the tasks and screenshots of the output.**
- **Please don't submit the zipped folders or .rar files etc.**
- **Lastly, you are strictly supposed to submit within the deadline (on LMS, not via emails).**