Laboratory 2 Image Interpolation

1. Use nearest neighbor interpolation and bilinear interpolation to interpolate a grey scale image. Implement the interpolation as a function with the form:
bilinear_姓名(input_file, dim), or nearest_姓名(input_file, dim)
where input_file is the file name that to be interpolated, and dim is a 1 × 2 vector (data type 'list' maybe used) specifying the row and column numbers of the interpolated image. The dimension of the interpolated image may be larger or smaller than that of the original image.

2. Use Python function "interp2d" from packet "scipy" or your own written algorithm to interpolate a grey scale image by using bicubic interpolation. The requirements and naming rules for the function is the same as Question 1.

Scipy interp2d function reference:
https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp2d.html

Report requirements:
Report should include, but not limited to, the following parts:
1. Title
2. Objective of the laboratory and the basic principle of the algorithms.
3. Derivation of your formula, if any.
4. Pseudo code of the algorithm, if the algorithm is not a straightforward computing.
5. Python codes. (citing the most important codes in your report if they are necessary for your explanation, and submitting the entire code as a separated attachment)
6. Results, including images (included in the report, and meanwhile submitting a separated attachment), tables, data, figures, if any.
7. Other (comparison) algorithms if any.
8. Analysis and conclusions.

If a report has only a title, a Python code with resultant images, the report will have not more than 60 marks over 100 marks. Description, derivation, pseudo code for complex algorithms, comparison and analysis take heavy weight for high marks.

For each algorithm (the nearest neighbor, bilinear, and bicubic), submit your codes (with comments to important parts) and two interpolated images from 'rice.tif' with following two dimensions:

1. with dimension (256 * 1.x) × (256 * 1.x), where x is the last digit of your matriculation number; if the last digit of matriculation number is 0, set x to be 45. Round 256 * 1.x to the nearest integer.
For example, if you matriculation number is 11730234, the size of the interpolated image is round(256*1.4) by round(256*1.4) , i.e., 358 by 358.

2. with dimension (256 * 0.x) × (256 * 0.x), or (256 * (1– 0.x)) × (256 * (1– 0.x)), whichever is larger, where x is the same as the above. Round the dimension also to the nearest integer.

Submission documents:

Naming rules for files to be submitted:
interpolation_姓名.doc or pdf: The report for the interpolation algorithm.
nearest_姓名.py: The Python source code of the nearest interpolation with comments.
bilinear_姓名.py: The Python code of the bilinear interpolation with comments.
bicubic_姓名.py: The Python source code of the bicubic interpolation with comments.
enlarged_nearest_姓名.tif, enlarged_bilinear_姓名.tif, enlarged_bicubic_姓名.tif: The interpolated image file with an interpolation factor of 1.x
shrunk_nearest_姓名.tif, shrunk_bilinear_姓名.tif, shrunk_bicubic_姓名.tif: The interpolated image file with an interpolation factor of 0.x