# Assignment 3

COL-783

March 2022

The goal of this assignment is to perform Image super-resolution using single image. You will be required to implement a paper named "Super-Resolution from a Single Image"[1] for image super-resolution, further you need to implement extensions (as mentioned) over it.

## 1 Image Super-Resolution using Single Image

In the first part, you will implement Image super-resolution algorithm as proposed in [1].

Image super-resolution reconstruction refers to a technique of recovering a high-resolution (HR) image (or multiple images) from a low-resolution (LR) degraded image (or multiple images). Single image super-resolution aims at reconstructing a high-resolution (HR) image by restoring the high frequency details from a single low-resolution (LR) image. Single image super-resolution is heavily ill-posed since multiple HR patches could correspond to the same LR image patch. To address this problem, the Single image super-resolution literature proposes interpolation-based method[2], reconstruction-based methods[1], and learning-based methods[3].
Methods for Super-resolution (SR) can be broadly classified into two families of methods: (i) The classical multi-image super-resolution, and (ii) Example-Based super-resolution. Recurrence of patches within the same image scale (at subpixel misalignments) forms the basis for applying the classical SR constraints to information from a single image. Recurrence of patches across different (coarser) image scales implicitly provides examples of low-res/high-res pairs of patches, thus giving rise to example-based super-resolution. from a single image (without any external database or any prior examples). The main contribution of the [1] work was to combine the above two techniques and form a single unified computational framework for image super-resolution using single image.

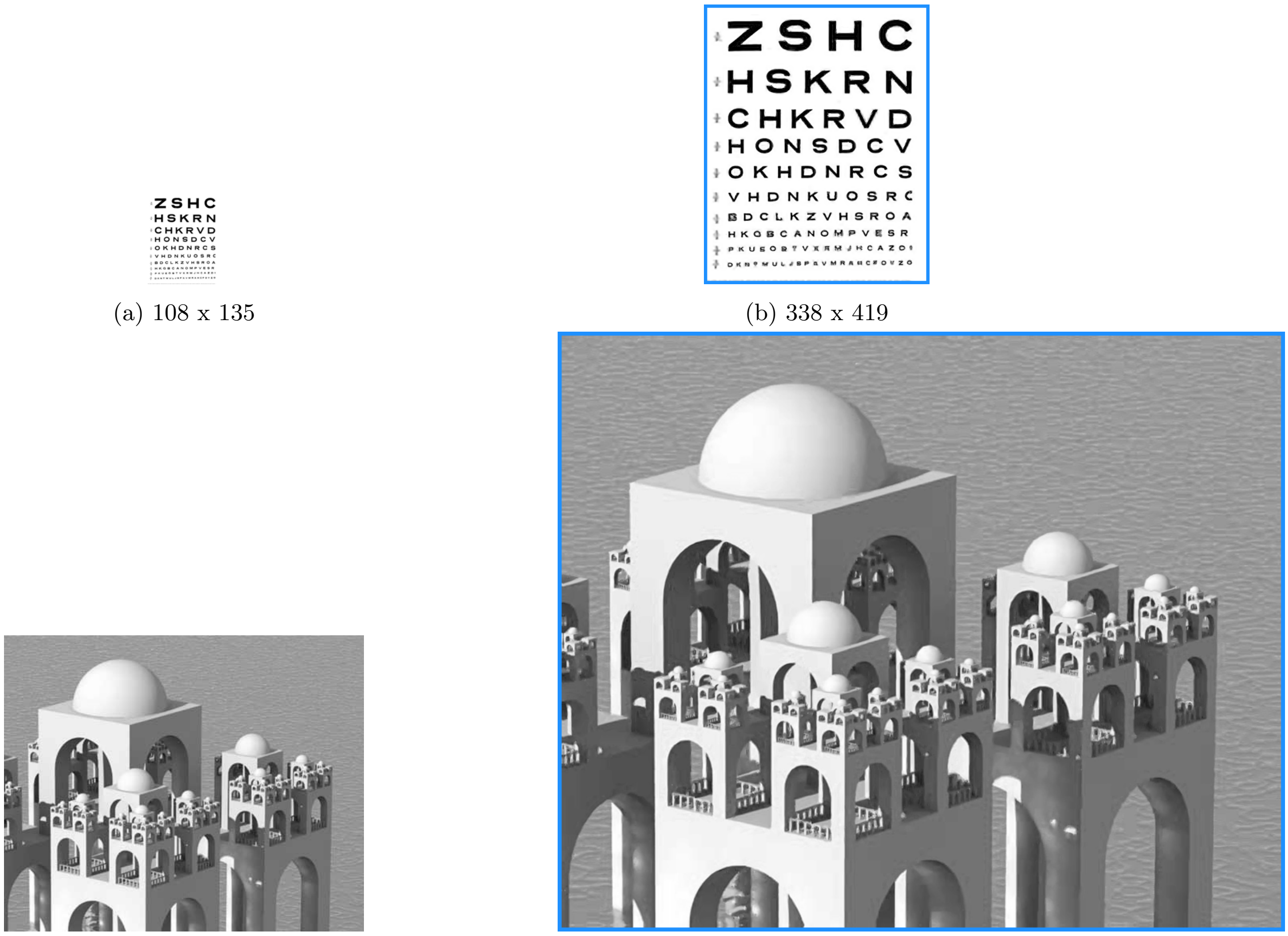
You should implement the following methods for Image super-resolution and compare the results:

1. Nearest-neighbor interpolation method
2. Bi-Cubic Interpolation method
3. [1] Super-resolution method

## 2 Extensions

In the second part, you will implement the following extensions over [1] technique:

1. compare the results of [1] technique using different Patch similarity Measures:
    - Gaussian Weighted SSD (Original as used by [1] )
    - Dot Product: Length of the projection of patch $p_1$ on patch $p_2$. Mathematically, $p_1.p_2 = \sum_i p_1[i]p_2[i]$. The dot product is large when $p_1$ and $p_2$ are similar.
    - Cosine distance between the patches: The cosine of the angle between patch $p_1$ and patch $p_2$. Mathematically, it can be computed using the formula $\cos\theta_{p_1,p_2} = \frac{p_1 \cdot p_2}{|p_1||p_2|}$ The images are similar if the cosine of the angle is close to 1 (so the angle is close to 0), and different if the cosine of the angle is close 0 (so the angle is close to $\pi/2$).




(a) 108 x 135

(b) 338 x 419

(c) 450x370

(d) 910x750

Figure 1: Few Examples: Original RGB images and their super-resolution form using [1] technique

2. Compare [1] results with the **Enhanced Prediction** technique as proposed by [4]. In image classification often the prediction for an input image is enhanced by averaging the predictions on a set of transformed images derived from it. In SR image rotations and flips should lead to the same HR results at pixel level. [4] apply rotations and flips on the LR image and then apply the SR method on each, reverse the transformation on the HR outputs and average for the final HR result.

3. Compare [1] results with the **Cascade of core SR method** technique as proposed by [4], using [1] approach in cascading. Instead of superresolving the LR image in small steps, [4] go in one step (stage) and then refine the prediction using the same SR method again adapted to this input. We consider the output of the previous stage as LR image input and as target the HR image for each stage. Thus, we build a cascade of trained models, where each stage brings the prediction closer to the target HR image.

# 3 Datasets

For both the parts, you will be provided with Image dataset, which consists of 3 images and your objective is to convert the original width and height in each image (wxh) by the mentioned amount.

### 3.1 Description of Dataset and the Requirement

1. Forest Image
    - Input size: 276 x 149
    - Output size (req): 552 x 296
2. World War 2 Image
    - Input size: 305 x 193
    - Output size (req): 610 x 385
3. Building Image
    - Input size: 242 x 131
    - Output size (req): 487 x 261

## Rules

1. Please use Python for implementation using only standard python libraries (e.g., numpy, opencv, matplotlib etc.). Please do not use any third party libraries/implementations for algorithms.
2. You can do the Project in groups of two, or individually.
3. You are required to write all the tasks (i.e. Image super-resolution paper implementation and its extensions) in different folders using '.py' format files.
4. Your code should have appropriate documentation for readability.
5. You can take inspiration from open-source code(s) available on the internet, **do not forget to cite the source** in your report as well as code, (but you can't copy the whole thing) the relative similarity between the code and referenced code should be less (if referred). You have to write code on your own.
6. You are advised to make your own functions and classes for different tasks.
7. For each part of the project, we have provided the dataset on which you should run your method. Include your resulting images in the report as well your submission folder.
8. Additional Reading: Refer resources section
9. You are not allowed to discuss or borrow code from other groups.
10. You will be graded based on what you have submitted as well as your ability to explain your code.

## Submission Instructions

1. Submit your source code, along with a readme.
2. Submit output files for all input images provided. The output file for a given image should be stored in the same directory as the input image.
3. Please prepare a report to accompany your implementation. The report should contain the methodology, design choices, results and analysis.Outputs (along with corresponding input images) should also be clearly shown in your report.
4. Zip the code and report in a single file, rename the zip as <Member1-entry-number> _ <Member2-entry-number>.zip and submit on moodle.
5. Only one submission per team.

## Evaluation Rubric

Your evaluation will be based on:

- Part 1 Paper Implementation of [1] - 4 points
- Part 2 Extension
  - Comparison with Various Patch similarity measures - 1 point
  - Comparison with **Enhanced Prediction** of [4] - 1.5 points
  - Comparison with **Cascade of core SR method** of [4] - 2.5 points
- Report + demo - 1 point

  **There are no quantitative metric on which you will be evaluated. Evaluation will be based on qualitative Analysis only.**

## Resources

Resources which could be used for references:

- SR by example Github Matlab code
- Matching Image Patches: Notes
- Seven ways to improve example-based single image super resolution [4]

## References


[1] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 349–356.

[2] P. Thévenaz, T. Blu, and M. Unser, "Image interpolation and resampling," *Handbook of medical imaging, processing and analysis*, vol. 1, no. 1, pp. 393–420, 2000.

[3] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 6, pp. 1127–1133, 2010.

[4] R. Timofte, R. Rothe, and L. Van Gool, "Seven ways to improve example-based single image super resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1865–1873.