AI6126 Project 2
DIV2K Single Image Super-Resolution Challenge

Project 2 Specification

Important Dates

Issued: 20 October 2020

Test data released: **13 November 2020 12:00AM** (Revised)

Due: **20 November 2020 11:59PM** (Revised)

Group Policy

This is an individual project

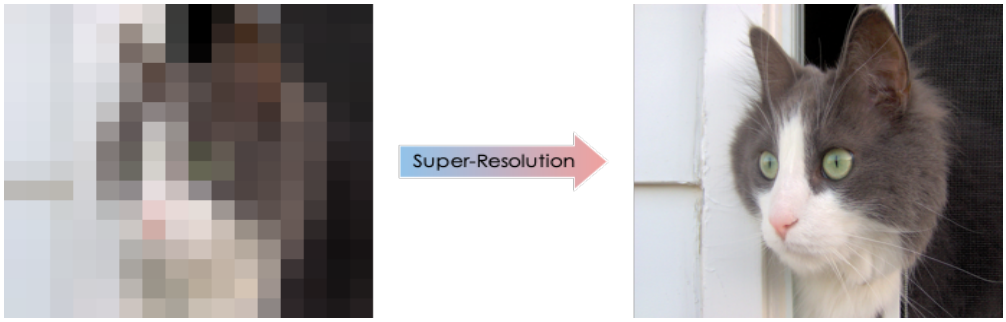
Late Submission Policy

Late submissions will be penalized (each day at 5% up to 3 days)

Challenge Description

The goal of this mini challenge is to increase the resolution of a single image (by four times). The data for this task comes from the DIV2K dataset [1]. For this challenge, we prepared a mini-dataset, which consists of 500 training and 80 validation pairs of images, where the HR images have 2K resolution and the LR images are downsampled four times.

For each LR image, algorithms will increase the resolution of the images. The quality of the output will be evaluated based on the PSNR between the output and HR images. The idea is to allow an algorithm to reveal more details imperceptible in the LR image.



Assessment Criteria

We will evaluate and rank the performance of your network model on our given **80** testing LR images based on the PSNR.

The higher the rank of your solution, the higher the score you will receive. In general, scores will be awarded based on the Table below.

Percentile in ranking	≤ 5%	≤ 15%	≤ 30%	≤ 50%	≤ 75%	≤ 100%	*
Scores	20	18	16	14	12	10	0

Notes:

- We will award bonus marks (up to 2 marks) if the solution is interesting or novel.
- Marks will be deducted if the submitted files are not complete, e.g., important parts of your core codes are missing or you do not submit a short report.

Submission Guideline

Students should improve the PSNR of their network model outputs.

- Download dataset: https://www.dropbox.com/sh/e7opsbgu5ww1qe3/AAAvrcVCykCR2-G--e2H1WCxa?dl=0
- Train your network using our provided mini training set.
- Tune the hyper-parameters using our provided mini validation set.
- To maintain fairness, your model should contain fewer than 1,821,085 trainable parameters, which is 120% of the trainable parameters in SRResNet [2] (your baseline network). You can use

  ```
  sum(p.numel() for p in model.parameters())
  ```

  to compute the number of parameters in your network.
- Submit output images of the test set for evaluations. Note that the test set is not extracted from DIV2K. The test set will be available one week before the deadline (this is a common practice of major computer vision challenges).
- No external data and pretrained models are allowed in this mini challenge. You are only allowed to train your models from scratch using the 500 image pairs in our given training dataset.
- You should not use an ensemble of models.

Each student can only turn in one submission. Resubmission is allowed. But only the latest one will be counted.

Submit the following files (all in a single **zip** file named with your matric number, e.g., **A12345678B.zip**) to NTU Learn before the deadline:

- A **short report in pdf format** of not more than five A4 pages (Arial 10 font) to describe the model that you use, the loss functions and any processing or operations that you have used to obtain your results. Report the **PSNR** of your model on the validation dataset, and also the **number of parameters** of your model.
- The results (i.e., the upscaled images) from your model on the 80 test images. Put them in a subfolder and use the same file name as the input image. (e.g. If your input image is named as 0001.png, your result should also be named as 0001.png)
- All necessary **codes** you used in this project.
- The **model checkpoint (weights)** of your submitted model.
- A **Readme.txt** containing the following info:
    - Description of the files you have submitted.
    - References to the third-party libraries you are using in your solution (leave blank if you are not using any of them).
    - Any details you want the person who tests your solution to know when he/she tests your solution, e.g. which **script to run**, so that we can check your results, if necessary.

Tips

- You can use BasicSR [3] or MMEditing [4] for this project. Both codebases have provided SRResNet [2] for you. You may need to make necessary adjustment (e.g. modifying data path, reducing batch size). For this project, BasicSR may be a more intuitive choice for you to start with. But it is up to you to choose the one you are comfortable with. Please specify in your report the codebase you used.
- The following techniques may help you to boost the performance:
    - data augmentation, e.g. random flip [5]
    - deeper model (but be careful of the parameter constraint)

Computational Resource

You can use the computational resources assigned by the MSAI course. Alternatively, you can use Amazon's EC2 or Google CoLab for computation. As a student, you can sign up to receive free $100 credit through the AWS Educate program. We encourage students to use *g2.2xlarge* instances running Ubuntu for maximal ease of installing. Note that $100 of Amazon credit allows you to run a *g2.2xlarge* GPU instance for approximately 6 days without interruption (you should keep it on only while using it).

References

[1] E. Agustsson and R. Timofte, NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study, CVPRW 2017
[2] C. Ledig et al., Photo-realistic single image super-resolution using a generative adversarial network, CVPR17
[3] BasicSR: https://github.com/xinntao/BasicSR
[4] MMEditing: https://github.com/open-mmlab/mmediting
[5] He et al. Bag of Tricks for Image Classification with Convolutional Neural Networks, ArXiv 2018