EE838B

Special Topic on Image Engineering: Advanced Image Restoration and Quality Enhancement

Homework Assignment 3

Implementation and Verification of deep learning-based video frame interpolation

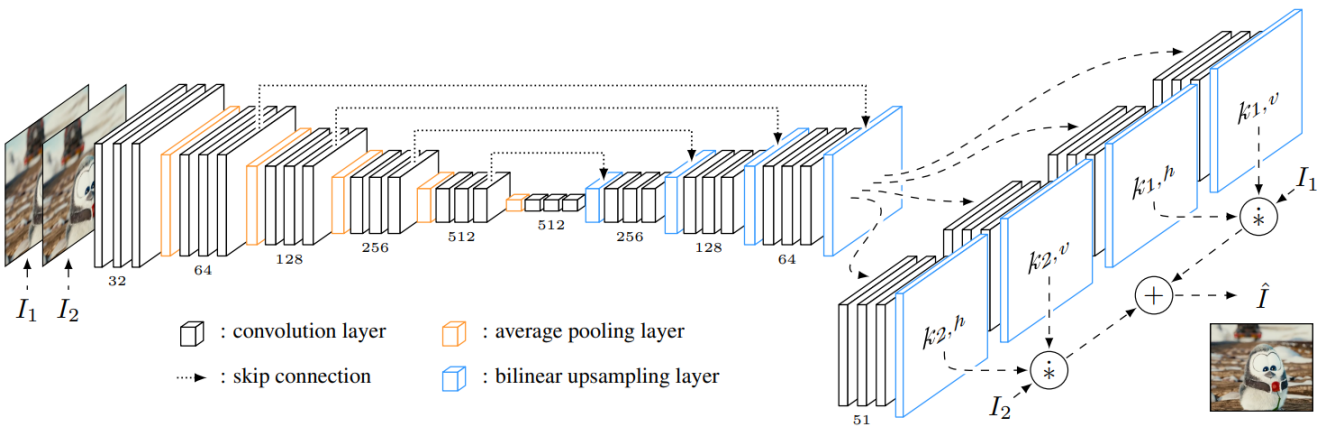
The third home assignment is to implement a "Video Frame Interpolation via Adaptive Separable Convolution (pdf)" to train it properly on Vimeo90K. After training, the performance is measured on UCF101 in terms of PSNR and SSIM. The data set for experiments is provided via the below Dropbox link. The evaluation of the submitted trained networks for Homework Assignment 3 will be performed based on the correct implementation and performance of the given network.

Dataset (1.68GB)

- Training samples: Vimeo90K (3783 triplets) (Input: im1.png, im3.png, GT: im2.png)
- Test samples: UCF101 (21 scenes) (Input: frame0.png, frame2.png, GT: frame1.png)
  - Refer this link (class ucf, DAVIS etc.) : how to compose input triplets.
- Download links:
  - Training/Test data: https://www.dropbox.com/s/o2d60jpbppugoco/Dataet_VFI_HW3.zip?dl=0
  - You can modify the previous skeleton code because input dataset loader will be different. (composing triplet of Vimeo90K, UCF101)

Implementation: Please **do not** copy GiHub codes. TA will check your codes.

- Adam Optimizer
- Only use L1 loss function, do not use perceptual loss of Eq. (3) in the paper.
- Training patch size: two 128×128×3 RGB images. (example of data format in PyTorch: [batch_size, C, T, H, W])
- Batch size: 8, Epoch: 200, estimated 1D kernel size set to be 25.
- You can use several data augmentation techniques such as randomly flipping horizontally and vertically, and rotating by 90 degrees etc.
- Other hyper-parameters for training can be freely selectable.
- The network architecture that you have to implement is shown in Fig. 2 of the paper.
- Important thing: generated kernels depends on input images so you have to consider **batch** dimension.



The following deliverables must be submitted:

- Both training and test codes (Pytorch)
- Readme.txt describes how to run your code, information of your code structure.
- Report
  - Experimental conditions.
  - Average PSNR and SSIM on UCF101.
  - The interpolated results with analysis produced by your test code for the three images with estimated kernels (visualization).
    - ucf101_HW3/15/frame1.png
    - ucf101_HW3/17/frame1.png
    - ucf101_HW3/20/frame1.png
  - One example result for occlusion case with visualized kernels with your analysis.
  - Simple code description for each component of the neural network.
  - Quantitative and qualitative comparisons with a network estimating 7x7-sized 2D kernels (AdaConv. (link))

Submission

- Due date: **2020-11-30 23:59 (Monday)**
- Submission should go to the class TAs at: jhoh94@kaist.ac.kr
- Submission format
  - Your report must include your name, student ID and e-mail
  - Your report must be in ZIP format with following directories:
    - ✓ **Do not** include the train and test **data** in your ZIP file.
    - ✓ `source` where readme.txt, training code and test code must reside
    - ✓ `report` where your report is put
  - The file name of your submission should be "HW3_studentID_YourName.zip".
  - **<NOTE: If your train and test code are not working, your implementation score is zero!!!>**