AI6126: Advanced Computer Vision

Ong Jia Hui (G1903467L)
JONG119@e.ntu.edu.sg

Project 2: DIV2K Single Image Super-Resolution Challenge Report

1 Introduction

Single image super-resolution (SISR) aims to recover a high-resolution (HR) image from its low-resolution (LR) observation. The objective of this challenge is to increase the resolution of a single image by four times (x4). The mini dataset given is from DIV2K [1] which contains 500 training and 80 validation pairs of images. There are some constraints set for this challenge which include model training strictly on the provided train set and pre-trained models and model ensemble are not allowed. In addition, the model must contain fewer than 1,821,085 trainable parameters, which is 120% of those in SRResNet [7].

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(I_{HR}(i) - I_{SR}(i))^2 \tag{1}$$

$$PSNR = 10 \cdot \log_{10}\left(\frac{L^2}{MSE}\right), \quad \textit{where } L = \textit{max pixel value} \tag{2}$$

For this challenge, the evaluation metric used is the Peak Signal-to-Noise Ratio (PSNR) score. PSNR is inversely proportional to the logarithm of the Mean Squared Error (MSE) between the ground truth image and the generated image as shown in Equation 2.

2 Implementation

2.1 Data Pre-processing

As the DIV2K training dataset contains large 2K images, in order to improve the speed of disk IO during training, the 500 HR images are first cropped into 20,424 of 480x480 subimages before converting into a lmdb dataset (HR_sub.lmdb). Similarly, the 500 corresponding LR images are also cropped into 20,424 of 120x120 subimages before converting to a lmdb dataset (LR_x4_sub.lmdb).

2.2 Model Architecture

The code baseline used for this assignment is BasicSR[1]. Three different model architectures were trialed, namely SRResNet [7] (baseline), Enhanced Deep Super-Resolution (EDSR) [8] and Cascading residual network (CARN) [2]. As shown in Table 1, due to parameter constraints, the downsized EDSR model with the same number of parameters as the baseline model and did not perform better than SRResNet. However, it is important to note that the SRResNet architecture provided in BasicSR was modified to use residual blocks without Batch Normalization. This modification was originally suggested in EDSR paper [8]. Hence, this modified SRResNet model will be denoted as MSRResNet. The CARN model[2] was configured to have four residual groups with four residual blocks each instead of the original 3×3 to increase its learning capability. The original CARN model has 1,592K parameters instead of the 1,769K parameters of the deeper model. Hereon, we will simply refer the deeper CARN model as "CARN". The best performing model trained uses MSRResNet architecture with 20 residual blocks instead of the original 16 residual blocks.

[1] https://github.com/xinntao/BasicSR
[2] https://github.com/nmhkahn/CARN-pytorch

Architecture	Parameters	Val PSNR
MSRResNet-B16	1,517,571	28.9027
EDSR_Mx4	1,517,571	28.8421
CARN	1,768,643	28.9341
MSRResNet-B20	1,812,995	28.9413

Table 1: PSNR scores of all light-weight SISR networks after 1M iterations

2.2.1 CARN Architecture

The CARN model [2] is based on ResNet architecture [4] like the baseline SRResNet [7]. The main difference between CARN and ResNet is the presence of local and global cascading modules. As shown in Figure 1, in the CARN model, each residual block is changed to a "local" cascading block and the blue arrows indicate "global" cascading connections. The outputs of intermediary layers are cascaded into the higher layers, and finally converge on a single 1×1 convolution layer.

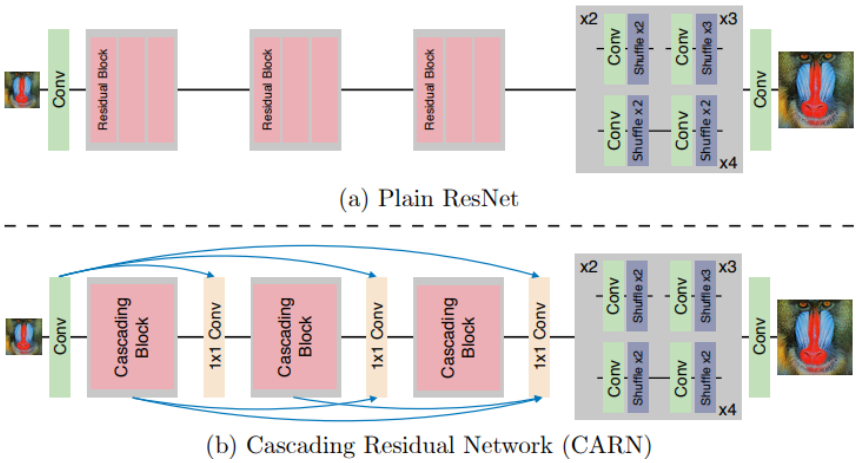
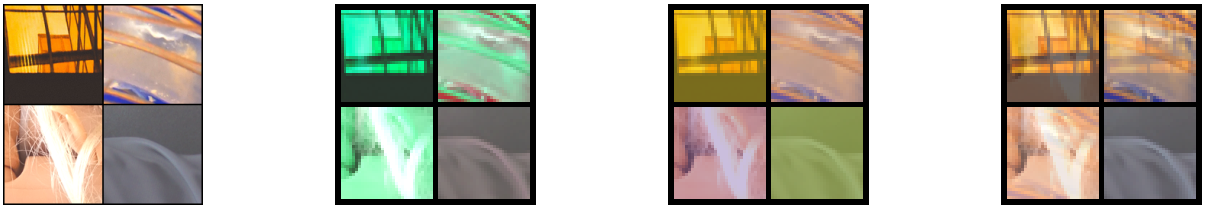


Figure 1: Comparison between Residual blocks in ResNet (top) versus Cascading blocks in CARN (bottom)

In the paper [2], it states that there are two advantages in using both local and global cascading blocks. Firstly it helps the model learns multi-level representations by incorporating features from multiple layers. Secondly, multi-level cascading connection behaves as multi-level shortcut connections that quickly propagate information from lower to higher layers and backwards in the case of back-propagation.

2.3 Data Augmentation



(a) HR (b) RGB Perm (c) Blend (d) MixUp

Figure 2: Examples of non-standard data augmentations applied to LR_x4 samples

Data augmentation is an important step in many computer vision tasks to "increase" the training set by creating variations of the original images. This expanded dataset alleviates the problem of overfitting

and allows the model to generalize better. The standard geometric manipulation techniques such as paired random crop, random horizontal and vertical flips and random rotations are applied in all model training.

Yoo et al. [10] recently released a paper introducing the alternative forms of augmentations that will be effective for Image Super Resolution. Among which, RGB permutation and Blending do not require any structural changes to the LR images. As experimented, with RGB permutation applied, it has indeed resulted in a 0.03 PSNR improvement on the validation set using MSRResNet model. As shown in Figure 2, RGB permutation is the random shuffling of the color channels of the HR and LR images (same permutations), while Blending adds a random constant value to the images. MixUp [11] is a data augmentation technique that blends two images to generate an unseen training sample. Previously for Project 1, I have also performed MixUp for the facial images, but for SISR, as the same pairs of HR and LR images are mixed together, there is no need to do a weighted linear interpolation to the loss function. The proposed method, CutBlur [10] was also trialed, but it does not seem to improve the PSNR for given task.

2.4 Loss Function

$$L1 = \frac{1}{N}\sum_{i=1}^{N}|(I_{HR}(i) - I_{SR}(i))| \tag{3}$$

$$CL = \sqrt{(I_{HR}(i) - I_{SR}(i))^2 + \epsilon^2} \tag{4}$$

As the PSNR metric is highly correlated with the pixel-wise difference, and minimizing the pixel loss directly maximizes the PSNR metric value. Three different pixel-wise loss functions were experimented to evaluate their effectiveness for this SISR task, namely L1 Loss (Equation 3), MSE Loss (Equation 1) and Charbonnier Loss (Equation 4) [6]. In my experiments, it was observed that L1 Loss performed better than Charbonnier Loss, which performed better than MSE Loss. The potential downsides of using MSE Loss for SISR were also extensively discussed in the studies by Anagun et al. [3]. Hence, for the training of the final model, L1 Loss is used.

2.5 Optimizer and Scheduler

The Adam optimizer [5] is used for optimization with an initial learning rate of 2e-4. All models are trained using the CosineAnnealingRestartLR [9] scheduler provided by BasicSR, which restarts the learning rate at every 250k iterations.

3 Results

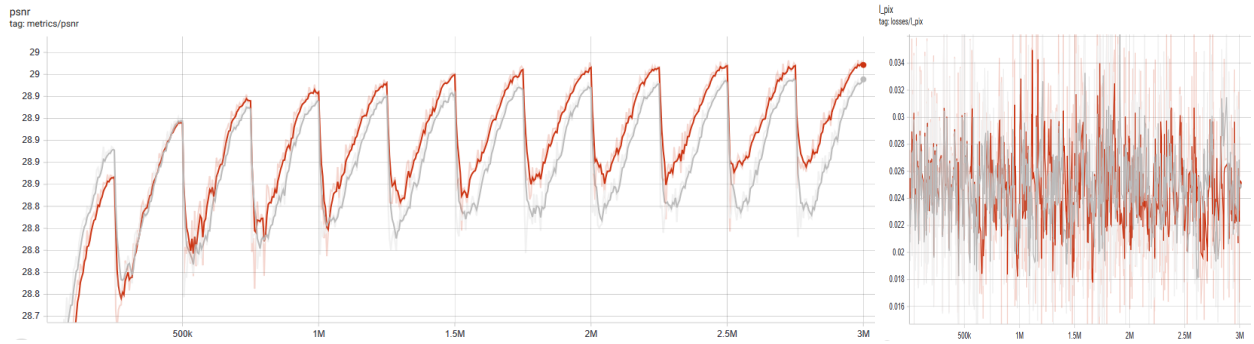
3.1 Baseline Comparison

Figure 3a shows the PSNR score comparison between baseline MSRResNet-B16 and CARN trained with standard geometric and RGB permutation augmentations after 3 million iterations. As shown in the Figure, CARN model (hereby denoted as CARN-RGB) outperformed MSRResNet-B16 model after 500k iterations.

Figure 4 shows the PSNR score comparison between the MSSResNet-B20 model trained from scratch with all data augmentations enabled and the CARN-RGB model. The model outperformed the CARN-RGB model and baseline MSRResNet-B16 from the 250k iterations.

3.2 Validation Results

The second best model (CARN-5M) trained using the deeper CARN network architecture with the first 3M iterations trained using standard and RGB permutation data augmentations and the next 2M iterations trained with standard, RGB permutation, blending and MixUp augmentations. The PSNR score of the CARN model on the Mini-DIV2K Validation set is 28.9902. The best performing model (MSRResNet-B20) was trained from scratch with all data augmentations turned on. Due to time constraint, it was only trained



(a) PSNR plot (b) Loss plot

Figure 3: Plots comparing baseline MSRResNet-B16 (Grey) and CARN-RGB (Orange) after 3M iterations

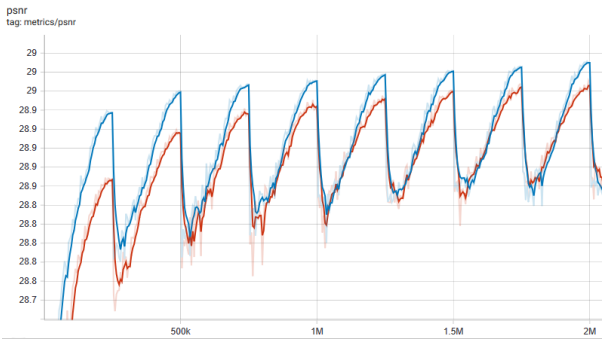


Figure 4: PSNR plot comparing CARN-RGB (Orange) and best model MSRResNet-B20 (Blue)

Architecture	**Augmentations**	**Parameters**	**Iterations Trained**	**PSNR**	
				DIV2KVal	**Set5**
MSRResNet-B16	RGB	1,517,571	3,000,000	28.9654	29.5005
CARN	RGB	1,768,643	3,000,000	28.9801	29.6343
CARN	RGB, Blend, MixUp	1,768,643	5,000,000	28.9902	29.6535
MSRResNet-B20	RGB, Blend, MixUp	1,812,995	2,000,000	**28.9915**	29.6094

Table 2: PSNR scores of baseline and final models on Mini-DIV2K validation set and Set5

for 2M iterations, but it has already outperformed the CARN-5M model on the Mini-DIV2K validation set with PSNR of **28.9915**.

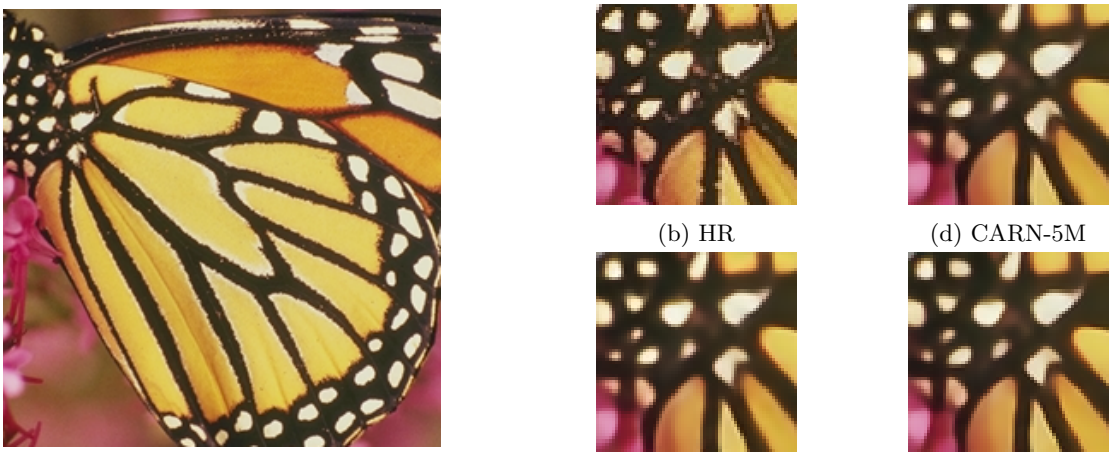
4 Output Visualization

4.1 Mini-DIV2K Validation and Set5 Restored Images

As seen in Figure 5, the restored images from the trained models were not able to reconstruct the spiral textures on the walls behind the stone dragon. Perceptually, the differences between the reconstructed images from different models are nearly unidentifiable for the DIV2K Validation set. However, slight optical differences can be observed on the butterfly wings in Figure 6 from Set5.

(a) Original 0018.png (b) HR (c) Baseline (d) CARN-5M (e) MSRResNet20

Figure 5: Original 0018.png (DIV2K) versus x4 reconstructed images from various models.



(a) Original butterfly.png (b) HR (c) Baseline (d) CARN-5M (e) MSRResNet20

Figure 6: Original butterfly.png (Set5) versus x4 reconstructed images from various models.

5 Conclusion

This report documented the model used for this Mini-DIV2K SISR challenge and the various data augmentations trialed. Four models were trialed namely the baseline MSRResNet-B16, EDSR_M, CARN and MSRResNet-B20. EDSR_M was trained for 1M iterations, but as it did not outperform the baseline, it was discontinued from the experiments. The CARN model was intensively trained with different augmentations enabled. It performed well with a PSNR of 28.9902 after 5M training iterations. However, the MSRResNet with increased residual blocks (MSRResNet-B20), has proved to outperform the CARN model when trained with standard geometric augmentations, RGB permutation, Blend and MixUp augmentations within 2M iterations. This model has **1,812,995** parameters (within constraints) and obtained a PSNR score of **28.9915** on Mini-DIV2K. It is likely that it can perform even better at 3M iterations, which shows the effectiveness of having a bigger model (more parameters) and better augmentations.

Credits

The computational work for this project was jointly performed on the resources of NTU's SCSE MSAI GPU cluster and the National Supercomputing Centre, Singapore (https://www.nscc.sg).

References

[1] Eirikur Agustsson and Radu Timofte. "Ntire 2017 challenge on single image super-resolution: Dataset and study". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 126–135.

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. "Fast, accurate, and lightweight super-resolution with cascading residual network". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 252–268.

[3] Yildiray Anagun, Sahin Isik, and Erol Seke. "SRLibrary: Comparing different loss functions for super-resolution over various convolutional architectures". In: *Journal of Visual Communication and Image Representation* 61 (2019), pp. 178–187.

[4] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[5] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[6] Wei-Sheng Lai et al. "Deep laplacian pyramid networks for fast and accurate super-resolution". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 624–632.

[7] Christian Ledig et al. "Photo-realistic single image super-resolution using a generative adversarial network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4681–4690.

[8] Bee Lim et al. "Enhanced deep residual networks for single image super-resolution". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 136–144.

[9] Ilya Loshchilov and Frank Hutter. "Sgdr: Stochastic gradient descent with warm restarts". In: *arXiv preprint arXiv:1608.03983* (2016).

[10] Jaejun Yoo, Namhyuk Ahn, and Kyung-Ah Sohn. "Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8375–8384.

[11] Hongyi Zhang et al. "mixup: Beyond empirical risk minimization". In: *arXiv preprint arXiv:1710.09412* (2017).