Fast CNN Enhancement using Channel Attention and Residual Networks for Image Super-resolution

Haoning Qu1†, Ming Zhu2†, Yifeng Ruan3†, Zerui Wan4†

1College of Arts and Science, New York University, New York, NY, 10012, USA

2New College, University of Toronto, Toronto, ON, M5S 1A1, Canada

3Alfred Lerner College of Business & Economics,University of Delaware, Newark, DE, 19711, USA

4Faculty of Applied Science & Engineering, University of Toronto, Toronto, ON, M5S 1A4, Canada

[ming.zhu@mail.utoronto.ca](mailto:ming.zhu@mail.utoronto.ca)其他人邮箱呢

†These authors contributed equally.

**Abstract**. Single image super resolution (SISR) refers to the process of reconstructing a high-resolution (HR) image from a low-resolution (LR) input image. Deep learning super-resolution algorithms have widely been used to solve SISR tasks. However, the high computational cost and memory storage for training the deep learning models has been hindering its real-world application. In this paper, we rebuild FSRCNN and apply it to solve SISR tasks. Firstly, we change the original training dataset to RealSR, a larger dataset consists of real-world images. Secondly, we add channel attention and shortcut connections into the mapping layers and reset important parameters including learning rate and optimizer. Thirdly, we change the loss function from Mean Squared Error (MSE) to Mean Absolute Error and replace the activation function from PReLU to ELU, to verify the discrepancies between different loss functions and activation functions. Finally, we compare the proposed model with official FSRCNN based on Peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM) on three commonly used test datasets. The experimental results show that the performance of rebuild model is slightly better. The proposed model achieves higher PSNR and SSIM on all the test datasets across different scale factors. Our analyses illustrate that different loss function and activation function do not generate large impact on the rebuild model.

**Keywords:** super-resolution, super-resolution, deep convolutional neural networks, residual block, residual block, channel attention

1. Introduction

Single image super-resolution (SISR) aims to recover a single HR image from a corresponding low-resolution (LR) image, an archetypal problem of computer vision research field. Since the inception of digital imagery, the demand of high-resolution (HR) images has become increasing higher within application fields including intelligent surveillance, medical imaging, and remote sensing. This is because higher image resolution contains rich details of the image and has a better visual quality. Due to the high demand of HR images, image super-resolution (SR) has been attracting substantial research interests for a long time. Since a same LR image could be originated from down-sampling infinitive HR images, SISR is still an ill-posed problem [1].

To address the above issue, many deep learning technologies has been proposed, among which Convolutional Neural Network (CNN) models has become the preferred method to be applied to solve SISR tasks. Dong et al. [2] introduced the first deep CNN, a three-layer Super-Resolution Convolutional Neural Network (SRCNN) to acquire the SR image in an end-to-end mapping manner. Though the simple and shallow three-layer network achieved great performance, deeper network structures have also been proved to enhance performance of SISR tasks [3].

The desire of pursuing superior SR performance has motivated researchers to design deeper and more complex networks such as DRCN [4], SRGAN [5] and SwinIR [6] to solve SR tasks. DRCN is a deep CNN based models contains a very deep recursive layer which is consist of 16 recursions. Ledig et al. proposed SRGAN, which is the first model to solve the SR task through the utilization of Generative Adversarial network. SwinIR is designed mainly based on Swin Transformer. Although these deep and sophisticated models have pursued great performance on SISR tasks, the high computation cost for memory storage due to its architecture complexity has become the obstacle to its practical application. To solve this problem, multiple lightweight networks have been proposed, such as FSRCNN [7] and ESRT [8]. FSRCNN is a re-designed SRCNN that can solve SR task faster and better. ESRT is a combination of lightweight CNN Backbone and a lightweight Transformer backbone. Though ESRT reduces the network parameters, its computation cost and training difficulty is still very high, while FSRCNN could be trained and achieve good performance on a generic CPU.

This paper proposes a re-build FSRCNN to solve SISR task. Firstly, we train the model on real-world super-resolution (RealSR) [11], a large dataset consists of realistic images to obtain a well-trained model. Secondly, we rebuild the model by adding Squeeze and Excitation (SE) [9] blocks, residual blocks [10] in the mapping layers and resetting parameters in convolutional layers and activation function after each convolutional layer. Thirdly, we changed the loss function from Mean Squared Error (MSE) [16] to Mean Absolute Error (MAE) [17]. To verify the effectiveness of the proposed method, we compare our model to the official FSRCNN with two common objective methods: Peak signal-to-noise ratio (PSNR) [13] and the structural similarity index measure (SSIM) [14]. Finally, we set up several controlling experiments to investigate effectiveness of different model structures, different loss function and different activation function. The rebuild model outperforms the original FSRCNN model on the test datasets used by the official FSRCNN model across all the scale factors.

1. The Proposed Method

This part introduces the proposed method with the re-implemented FSRCNN model. Firstly, we use a larger dataset RealSR instead of using the original training dataset General 100 [7] and 91-image dataset [12] to get a self-trained model for RSISR tasks. Then we rebuild FSRCNN by adding channel attention through SE blocks [9] and set up residual blocks [10] in the convolutional layers. Finally, we reset the parameters of the rebuild model to increase the SR performance and make comparisons with official model, including optimizer, learning rate, activation function and loss function.

## Data Pre-processing

Since we aim to increase the generalization capability of the proposed model, the RealSR [11] version 3 train dataset is utilized to get a well-trained model. The volume of the RealSR is relatively large in comparison to the original training dataset, for the sake of reducing training time, we only apply images captured through Canon 5D3 which is only half of the RealSR training dataset.

## Model Structure

We add SE blocks shown in figure.1 and residual blocks in the mapping layers of the rebuild FSRCNN model. The official FSRCNN [7] is redesigned based on SRCNN [2]. There are three main structural differences between them. Firstly, FSRCNN directly learns from the original low-resolution image. Secondly, FSRCNN applies a deconvolution layer to perform up-sampling. Thirdly, the Non-linear mapping step is replaced by shrinking, mapping, and expanding. Besides SE blocks and Residual blocks, we rebuild the FSRCNN by resetting the activation function after each convolution layers from parametric rectified linear unit (PReLU) to exponential linear unit (ELU) [15] and changing the loss function from MSE to Mean Absolute Error (MAE).

|  |
| --- |
| Diagram  Description automatically generated |
| **Figure 1.** Structure of a SE block [9] |

* + 1. SE blocks: A SE block contains two operations: Squeeze and Excitation. Squeeze operation aims to obtain the feature of the input through Global Average Pooling. Excitation operation is achieved by two fully connected layers which enables the convolution layers make full use of information gathered in the squeeze operation. SE blocks introduces channel attention to the network which enables the model to recalibrate channel wise feature information. The recalibration can improve the model performance by emphasizing important features and mitigate unnecessary features. Moreover, SE blocks are computational efficient which enhance the complexity of the model with slight computational burden.
    2. Residual blocks: By setting up residual blocks, we aim to ameliorate the degradation and vanishing gradients problem caused by model complexity. The residual blocks enable the output from the previous convolution layer feed forward to layers from more than 1 hops which make it easier for the model to learn the expected feature mapping.
    3. Convolution layers: Convolution layer is the cardinal part of CNN based models, which incur the most computation tasks in the model. The computation volume of output O is determined by the size of input data W, Stride S, filter size F, and padding P, which is defined in (1):

(1)

* + 1. Activation function: The original FSRCNN model adopts PReLU instead of Rectified Linear Unit (ReLU) as the activation function after each convolution layers, which is designed to mitigate the zero gradients caused by ReLU. Since ELU enables faster learning, we also applied ELU as the activation after each convolution layers.

## Model parameter settings

We reset several parameters in the rebuild model, including optimizer and learning rate. A detailed description is as below.

* + 1. Optimizer: Different from official FSRCNN, we use Adam optimizer [18] instead of using stochastic gradient descent (SGD). This is because Adam optimizer has a minimal memory usage. Besides, the convergence speed of Adam optimizer is relatively fast and it can dynamically adjust the hyper-parameters. By applying a higher learning rate for low-frequency parameters the network can learn more information.
    2. Learning rate: The learning rate is a crucial hyper-parameter that determines the extent of weight assigned to newly acquired information and the convergence time. Different from the official FSRCNN [4], we reset the learning rate of convolution layers to 0.0002 and decay the learning rate to half of the previous rate every 200 epochs. Moreover, we remain the original learning rate of 0.0001 in deconvolution layer.

## Loss function

One type of learning strategy in machine learning is loss function, which is used to measure prediction error or reconstruction error. During the training, it provides a guide for the model optimization. In our proposed model, we investigated two frequently used loss functions.

* + 1. MSE loss function: The loss, also known as Mean Squared Error [16], is the squared difference between a prediction and the actual value for each sample in the given dataset. The aggregation of all these loss values is calculated as below in (2):

In this equation, is the height of the image, is the width of the image, and is the number of channels of the image.   is the constructed individual pixels value at row , column and channel , is the original individual pixel value. Using MSE as cost function enables the model to get a high PSNR, which is one of the assessment metrics to evaluate model performance. Many CNN based networks such as SRCNN and FSRCNN use MSE as cost function.

* + 1. MAE loss function: The loss, which is the sum of the all the absolute differences between the actual value and the predicted value, also known as Mean Absolute Error [17]. Compared to loss, loss provides commendable accuracy and convergence ability to the model, although it may not promote the model to achieve a better PSNR. The equation of loss is defined as in (3):

1. Experiment Settings

This section illustrates the experimental setting in this paper. Firstly, we introduce the dataset used for training, validation and test (Sec. 3.1). Secondly, we describe the performance measurement indicators including PSNR and SSIM (Sec. 3.2). Then we illustrate our test on different loss function. Finally, we elaborate our baseline model and controlling experiments on different settings (Sec. 3.3).

## Dataset Description

This part gives a detail description of our training dataset, test and validation dataset. In addition, we also introduce our training strategy in the training dataset section.

* + 1. Training dataset: We use RealSR version 3 [11] as training dataset instead of using The 91-image [12] and General-100 dataset [7]. RealSR dataset consist of realistic HR-LR images pairs captured by Canon 5D3 and Nikon D810 through focal length adjusting with the scales of 2,3,4 respectively. In terms of assuring the convergence of our model, we trained our model with a dynamic learning rate for 1000 epochs. We trained our model on a generic device, a RTX 2060, the whole training process takes about 12 hours.
    2. Test and Validation dataset: In terms of comparing the performance of our rebuild model with the original FSRCNN model, we also adopt Set5[19], Set14 [20] and BSD200 [21] as testing dataset. Set5 and Set 14 are commonly used testing dataset which contains 5 and 14 different images respectively. The BSD200 dataset contains 200 different images with high noise level. The dataset contains images of animals, natural scenes and artificial landscapes. Among the pictures, some subjects have similar shape or colour as the background of the images.

## Performance measurement metrics

We use two commonly used quality evaluation method to assess the performance of our model, PSNR [13] and SSIM [14]. A detailed description for each of them is below.

* + 1. Peak signal-to-noise ratio: The peak signal-to-noise ratio (PSNR) is one of the most commonly used objective quality evaluation metric for image reconstruction. With a given maximum, and X, the PSNR is defined as in (4):

where R stands for the maximum pixel value and MSE represents the Mean Squared Error between and X. Generally, higher PSNR indicates a higher restoration quality. However, in some cases a higher PSNR may result in a poor correlation in visual quality based on human perception.

* + 1. structural similarity index measure: The structural similarity index measure is also a widely used objective quality evaluation metric which measures the structure similarity between the reconstructed image and original image. The evaluation process is collectively performed in three aspects: luminance, contrast, and structure. The SSIM is defined as in (5):

(5)

## Baseline model and different settings

We rebuild FSRCNN [7] by adding SE blocks [9] and residual blocks [10] as our baseline, then we design several controlling experiments to test the effectives of our proposed method. We give detailed description of them below.

* + 1. Baseline model: FSRCNN is a relative shallow network which consist of five parts: feature extraction layer, shrinking layer, Non-linear mapping layer, expanding layer and a deconvolution layer. Since the Non-linear mapping layer exerts significant impact on the SR performance of the model, we add SE blocks and residual blocks through all the mapping layers to enhance the efficiency of the mapping layer.
    2. Investigation on different settings: We implement controlling experiments to verify the discrepancies of baseline model based on different settings, including model structure, loss functions and activation functions. Firstly, we test the influence of residual blocks by removing all the residual blocks of baseline model. Secondly, we explore the impact of SE blocks by only keep the first SE block in the mapping layer and remove other SE blocks. Then we examine the usefulness of different loss functions, we apply MAE loss as loss function for our baseline model as well as model without residual blocks. Finally, we measure the effect of different activation function, we adopt ELU as activation function for our baseline model and models without residual blocks.

1. Result and Discussion

This section illustrates the detailed performance of the proposed model with different settings, including PSNR, SSIM and visual quality. Firstly, we test the effectiveness of different structure - residual blocks. Then we compare the model performance between loss and loss. Finally, we investigate the influence of different activation functions based on PReLU and ELU.

## Comparison on Residual blocks

From table.1 we can see that all the models with residual blocks reach a slightly higher PSNR and SSIM compared to model without residual blocks except for the factor 4 test on BSD200. Though Residual block is originally designed to mitigate the vanishment of gradients and degradation problems for models with deeper structure and more complexity, the experiments result indicate that it can also slightly improve the performance of models with a relative shallow structure. We apply all the residual blocks in our mapping layers. If we add more layers in our mapping layers, the residual block will further improve the performance of the model.

## Comparison on SE blocks

We also test the effectiveness of SE blocks. We reset the proposed baseline model by reserve the SE block in the first convolution layer of mapping layers remove the other SE blocks in the mapping layers while keep the residual blocks in the mapping layers and obtains our best model. From table.2, the results show that by removing other SE blocks while maintaining the residual blocks unchanged, our best model outperforms the official FSRCNN based on the test of Set5, Set14 and BSD200 across all the upscaling factors. Moreover, the experiment result shows that the proposed model gets higher PSNR on BSD200 test dataset than Set14 for scale factor 3 and 4, which indicates the proposed model has great generalization capability. The reconstructed “butterfly” image from Set5 of upscale factor of 3 and the “lenna” image from Set14 of upscale factor of 3 are shown in figure.2 and figure.3 respectively.

## Comparison on different Loss Function

Based on our experiments, MSE and MAE loss function do not result in a salient difference. Specifically, for the models without residue blocks, the results of using MAE as loss function are in analogy to models using MSE loss function. The PSNR values for the scale 2 test sets using MAE are slightly higher than those using MSE. For the scale 3 test sets, the PSNR values for two different methods are almost the same. However, the performance for the model using MAE loss function goes down by 0.3 dB specifically for the scale 4 BSD200 testing set compared to the baseline model. Moreover, for the experiments utilizing residue block, the overall test results for the model using MAE as loss function appear to be lower than the results using MSE. The differences for scale 2 and scale 3 testing sets are not prominent, at the maximum of 0.13 dB. Nevertheless, the scale 4 BSDS200 testing result for the model using MAE is 0.33 dB lower than the one using MSE. Generally, MSE loss function will enable the model to reach a higher PSNR, the experiment results are consistent. The quantitative results across all the upscaling factors are shown in table.1.

## Comparison on different Activation Functions

Original FSRCNN adopts PReLU as the activation function instead of commonly used activation function ReLU, primarily aims to avoid the “dead features” caused by zero gradients in ReLU. In our experiments, we investigate the effect of ELU activation function and compared the results with the original activation function. The average PSNR values on all the test dataset of models with different residual block setting and different activation function are listed in table 3. In the experiments with no residue block, the PSNR values corresponding to the model using ELU are similar to the baseline PReLU methods. For scale 2 and scale 4, the results are almost equal. The result of scale 3 testing sets for model using ELU is slightly better. However, the differences are trivial, which are at most 0.02 dB. In addition, for the experiments using residue block, the result is consistent. The PSNR values goes down while choosing ELU activation function instead of PReLU. Only for the scale 3 experiments, we can see that model with ELU performs better. Thus, overall, the performance of PReLU and ELU activation functions is analogical. Specifically, ELU is a better choice in scale 3 experiments. In addition, while using ELU as activation function the parameters in the model dropped 1xxx which slightly release the computational cost of the model.

1. Conclusion

In this paper, we rebuild the FSRCNN by adding SE blocks and residual blocks in the mapping layers to achieve a better SISR performance. Meanwhile, we also substitute the original synthetic training dataset to a training dataset consist of realistic images. Then, we implement several controlling experiments to explore the impact of SE blocks and residual blocks, MAE and MSE loss function, PReLU and ELU activation function. Our experimental results shows that the rebuild FSRCNN with only one SE block in the first mapping layer and with residual blocks placed in all the mapping layers can get higher PSNR and SSIM on the proposed test datasets than the official model. Our analysis illustrates that the residual blocks can slightly improve the performance of shallow networks. In addition, MAE and MSE loss function, PReLU and ELU activation function do not exert a significant impact on our rebuild model.

In the future, our study will focus on further improvement of the rebuild models. Meanwhile, we will try to apply our models to solve real-world image super-resolution tasks and strengthen the possibility of its real-world application.

**Table 1.** The results of PSNR (dB) on three test datasets for baseline model without residual blocks and apply MAE as loss function, baseline model and apply MAE as loss function.

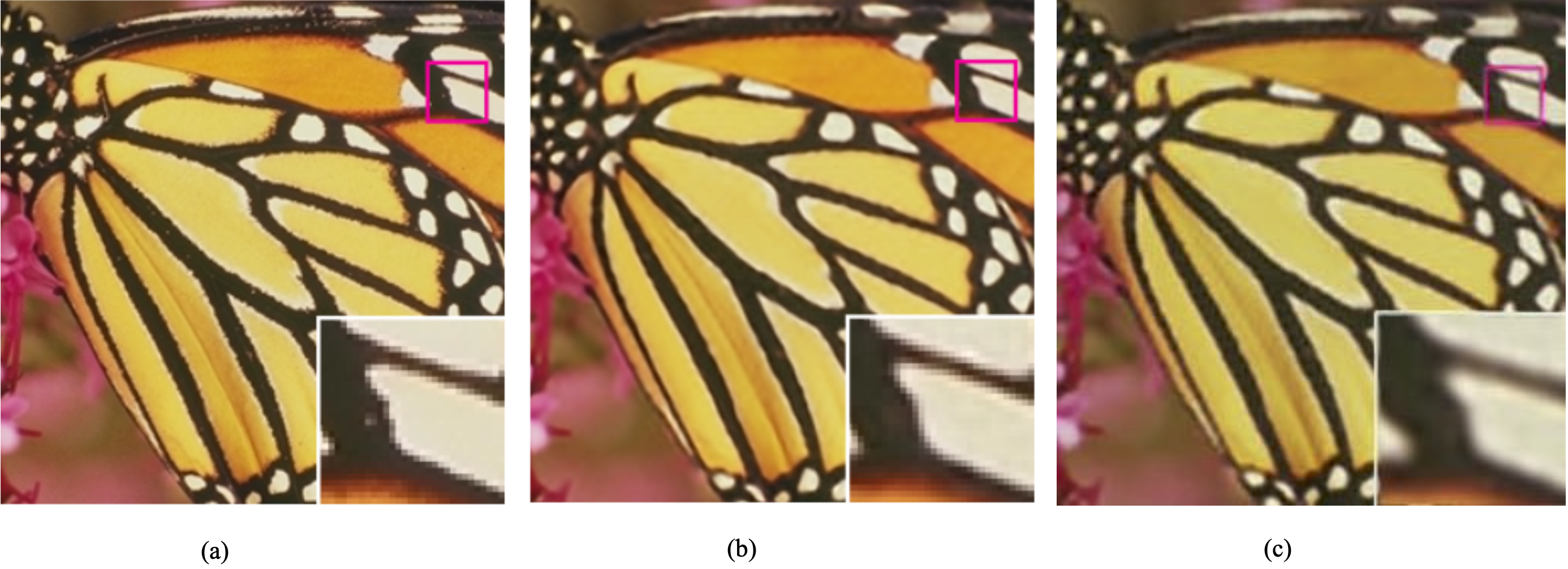
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Dataset | Upscaling Factor | w/o res | w/o res+ | baseline | base+ |
| PSNR | PSNR | PSNR | PSNR |
| Set5 |  | 31.51 | 31.58 | 31.68 | 31.52 |
| Set14 |  | 27.99 | 28.05 | 28.06 | 27.97 |
| BSD200 |  | 28.62 | 28.72 | 28.71 | 28.71 |
| Set5 | 3 | 27.21 | 27.13 | 27.27 | 27.20 |
| Set14 | 3 | 25.06 | 25.00 | 25.08 | 25.06 |
| BSD200 | 3 | 25.10 | 24.91 | 25.12 | 24.99 |
| Set5 | 4 | 26.19 | 26.16 | 26.31 | 26.19 |
| Set14 | 4 | 24.28 | 24.28 | 24.33 | 24.29 |
| BSD200 | 4 | 23.38 | 23.07 | 23.37 | 23.09 |

**Table 2.** Comparison of FSRCNN and our best rebuild model.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Dataset | Upscaling Factor | FSRCNN | Ours |
| PSNR/SSIM | PSNR/SSIM |
| Set5 |  | 37.00/0.9558 | 37.97/0.9629 |
| Set14 |  | 32.63/0.9088 | 33.18/0.9426 |
| BSD200 |  | 31.80/0.9074 | 32.21/0.9277 |
| Set5 | 3 | 33.16/0.9140 | 34.29/0.9372 |
| Set14 | 3 | 29.43/0.8242 | 31.05/0.8810 |
| BSD200 | 3 | 28.60/0.8137 | 31.68/0.9167 |
| Set5 | 4 | 30.71/0.8657 | 32.14/0.9071 |
| Set14 | 4 | 27.59/0.7535 | 29.44/0.8417 |
| BSD200 | 4 | 26.98/0.7398 | 30.19/0.8832 |

**Table 3.** The results of PSNR (dB) on three test datasets for baseline model without residual blocks and apply ELU as activation function, baseline model and apply ELU as activation function.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Dataset | Upscaling Factor | w/o res | w/o res+ELU | base | base+ELU |
| PSNR | PSNR | PSNR | PSNR |
| Set5 |  | 31.51 | 31.50 | 31.68 | 31.51 |
| Set14 |  | 27.99 | 27.97 | 28.06 | 27.99 |
| BSD200 |  | 28.62 | 28.63 | 28.71 | 28.64 |
| Set5 | 3 | 27.21 | 27.22 | 27.27 | 27.32 |
| Set14 | 3 | 25.06 | 25.07 | 25.08 | 25.13 |
| BSD200 | 3 | 25.10 | 25.11 | 25.12 | 25.16 |
| Set5 | 4 | 26.19 | 26.16 | 26.31 | 26.23 |
| Set14 | 4 | 24.28 | 24.26 | 24.33 | 24.32 |
| BSD200 | 4 | 23.38 | 23.39 | 23.37 | 23.40 |



**Figure 2.** (a) Original image (b) FSRCNN [7] (c) Our model



**Figure 3.** (a) Original image (b) FSRCNN [7] (c) Our model

参考文献格式不对，按下面要求改（年份位置、页码用pp等， 作者之间也没标点啊，注意细节！！！）

1. Sze S M 1969 *Physics of Semiconductor Devices* (New York: Wiley–Interscience)
2. Dorman L I 1975 *Variations of Galactic Cosmic Rays* (Moscow: Moscow State University Press) p 103
3. Caplar R and Kulisic P 1973 *Proc. Int. Conf. on Nuclear Physics (Munich)* vol 1 (Amsterdam: North-Holland/American Elsevier) p 517
4. Szytula A and Leciejewicz J 1989 *Handbook on the Physics and Chemistry of Rare Earths* vol 12, ed K A Gschneidner Jr and L Erwin (Amsterdam: Elsevier) p 133
5. Kuhn T 1998 Density matrix theory of coherent ultrafast dynamics *Theory of Transport Properties of Semiconductor Nanostructures* *(Electronic Materials* vol 4*)* ed E Schöll (London: Chapman and Hall) chapter 6 pp 173–214

References

1. Chunwei T Ruibin Z Zhihao W Yong X Wangmeng Z Chen C and Chia-Wen L 2020 Lightweight image super-resolution with enhanced CNN *arXiv* eess.IV(2020)2007.04344v3
2. Chao D, Chen C L, Kaiming H, and Xiaoou T Image super-resolution using deep convolutional networks *arXiv* cs.CV (2015)1501.00092v3
3. Jin Y Shigesumi K and Takio K Fast and accurate image super resolution by deep CNN with skip connection and network in network *arXiv* cs.CV (2017) 1707.05425v7
4. Jiwon K Jung K L and Kyoung M L Deeply-recursive convolutional network for image super-resolution *arXiv* cs.CV (2016)1501.04491v2
5. Christian L Lucas T Ferenc H Jose C Andrew C Alejandro A Andrew A Alykhan T Johannes T Zehan W and Wenzhe S Photo-realistic single image super-resolution using a generative adversarial network *arXiv* cs.CV (2017)1609.04802v5
6. Zhisheng L Juncheng L Hong L Chaoyan H Linlin Z and Tieyong Z Transformer for single image super-resolution *arXiv* cs.CV (2022)2108.11084v3
7. Chao D Chen C L and Xiaoou T Accelerating the super-resolution convolutional neural network *arXiv* cs.CV (2016)1608.00367v1
8. Zhisheng L Juncheng L Hong L Chaoyan H Linlin Z and Tieyong Z Transformer for single image super-resolution *arXiv* cs.CV (2022) 2108.11084v3
9. Jie H Li S Samuel Al Gang S and Enhua W Squeeze-and-excitation networks *arXiv* cs.CV (2019) 1709.01507v4
10. Kaiming H Xiangyu Z Shaoqing R and Jian S Deep residual learning for image recognition *arXiv* cs.CV (2015)1512.03385v1
11. Jianrui C Hui Z Hongwei Y Zisheng C and Lei Z Toward real-world single image super-resolution: a new benchmark and a new model *arXiv* cs.CV (2019) 1904.00523
12. Yang J Wright J Huang T S and Ma Y Image super resolution via sparse representation *arXiv* IEEE Transactions on Image Processing, vol. 19, no. 11, pp. 2861-2873 (2010)
13. Alain H and Djemel Z Image quality metrics: PSNR vs. SSIM *2010 International Conference on Pattern Recognition*
14. Honggang C Xiaohai H Linbo Q Yuanyuan W Chao R and Ce Z Real-world single image super-resolution: a brief review *arXiv* eess.IV(2021 2103.02368v1
15. Djork-Arne ́ C Thomas U and Sepp H Fast and accurate deep network learning by exponential linear units (ELUS) *arXiv* cs.LG(2016) 1511.07289v5
16. Jiali W Zhengchun L Ian F Won C Rajkumar K and V. Rao K Fast and accurate learned multiresolution dynamical downscaling for precipitation *arXiv* cs.LG(2021) 2101.06813v1
17. Hang Z Orazio G Iuri F and Jan K Loss functions for image restoration with neural networks *arXiv* cs.LG(2021) 1511.08861v3
18. Xuezhe M Apollo: an adaptive parameter-wise diagonal quasi-newton method for nonconvex stochastic optimization *arXiv* cs.LG (2016)2009.13586v6
19. Bevilacqua M Roumy A Guillemot C and Morel M.L.A. Low-complexity single-image super-resolution based on nonnegative neighbor embedding *arXiv* BMVC. (2012)
20. Zeyde R Elad M and Protter M On single image scale-up using sparse-representations *arXiv* *Curves and Surfaces* (2012) 711–730
21. Martin D Fowlkes C Tal D and Malik J A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics *arXiv* ICCV Volume 2 (2001) 416–423