Deep Convolutional Networks for Single Image Super-Resolution

Zheng Chunhang
20308154
zhengchh6@mail2.sysu.edu.cn

Xie Mingli
20308131
xiemli3@mail2.sysu.edu.cn

Xu Wenjie
20308133
xujay23@gmail.com

Abstract

Deep convolutional networks based super-resolution is a fast-growing field with numerous practical applications. In this exposition, we extensively compare three state-of-the-art super-resolution Convolutional Neural Networks (CNNs) [29] over two classical datasets to benchmark single image super-resolution.

We provide comparisons among three Convolutional Neural Networks in terms of network complexity, model input and output, learning details on different datasets and different scales.

The extensive evaluation performed, shows the consistent and rapid growth in the accuracy in the past few years along with a corresponding boost in model complexity and the availability of large-scale datasets. It is also observed that the pioneering methods identified as the benchmark have been significantly outperformed by the current contenders. Despite the progress in recent years, we identify several shortcomings of existing techniques and provide future research directions towards the solution of these open problems.

1. Introduction

In recent years, image super-resolution (SR) has gained significant attention in the field of computer vision and image processing. The goal of image SR is to enhance the resolution and quality of low-resolution images, thereby generating visually pleasing high-resolution images. This has numerous practical applications in various domains such as surveillance, medical imaging, remote sensing, and digital photography.

The demand for high-resolution images has increased rapidly due to advancements in display technologies and the need for detailed visual information. However, capturing high-resolution images directly is often limited by hardware constraints, such as sensor limitations or bandwidth restrictions. This has led to the development of image SR techniques, which aim to reconstruct high-resolution details from low-resolution observations.

Super-resolution methods can be broadly divided into two main categories: traditional methods and deep learning methods. [1, 2]

Traditionally, interpolation-based methods, such as bicubic interpolation, have been widely used for upscaling low-resolution images. However, these methods often result in blurred or unrealistic details, failing to capture the true high-frequency information present in the original high-resolution images. To overcome these limitations, recent research has focused on exploring the power of deep learning techniques, particularly deep convolutional neural networks (CNNs), for image SR.

Deep CNN-based super-resolution methods have shown remarkable performance improvements over traditional approaches. By leveraging large-scale datasets and powerful network architectures, these methods have demonstrated the ability to generate highly realistic and visually appealing high-resolution images. The inherent capability of deep CNNs to learn complex mappings and extract hierarchical features makes them well-suited for the challenging task of image SR.

In this paper, we aim to provide a comprehensive overview and analysis of state-of-the-art deep CNN-based including VDSR [18], SRResNet [21] and EDSR [23] image SR techniques. We will investigate and compare various approaches and architectures proposed in recent literature. Additionally, we will evaluate their performance on multiple benchmark datasets and different scales.

2. Bicubic

Bicubic interpolation is the most commonly used interpolation method in two-dimensional space, and it is often used to improve the resolution of images and as a comparison in many papers on super-resolution reconstruction. In this method, the value of the function f at the point (x,y) can be obtained by a weighted average of the nearest sixteen pixels (4×4). One method [17] of calculating the Bicubic interpolation is as follows:

1. Using Eq. (1), calculates the weights $W(x)$ of the 16 pixels.Variable a is usually set to -0.5 or -0.75.

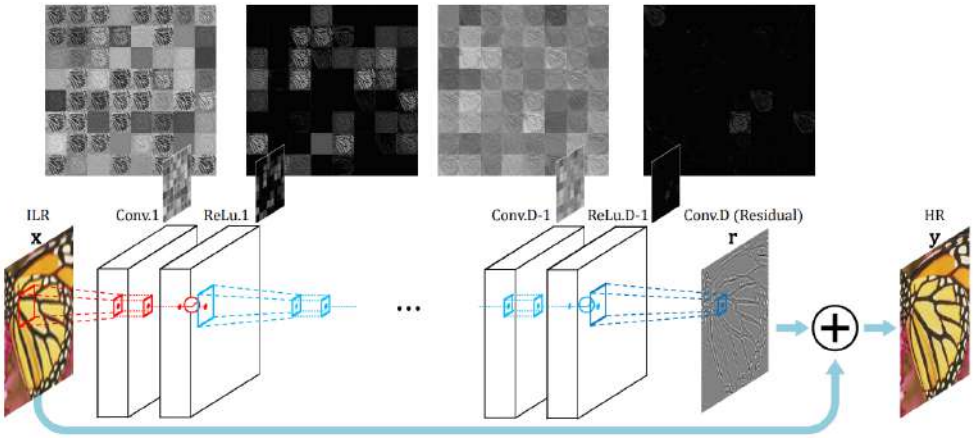

Figure 1. Architecture of VDSR

$$W(x)=\begin{cases}(a+2)|x|^3-(a+3)|x|^2+1 & \text{for } |x|\leq 1\\ a|x|^3-5a|x|^2+8a|x|-4a & \text{for } 1<|x|<2\\ 0 & \text{otherwise}\end{cases} \quad (1)$$

2. Then bring the above calculated $W(X)$ into Eq. (2), get the results of $f(x, y)$.

$$f(x,y)=\sum_{i=0}^{3}\sum_{j=0}^{3} f\left(x_i,y_j\right) W\left(x-x_i\right) W\left(y-y_j\right) \quad (2)$$

3. Very Deep Convolutional Networks

3.1. Previous Work

Super-Resolution Convolutional Neural Network(SRCNN) [6] is the first model aimed at single-image super-resolution based on Deep Learning.

The main objective of SRCNN is to learn an end-to-end mapping function using a deep CNN that can effectively enhance the resolution of low-resolution images. Unlike traditional interpolation-based methods, SRCNN directly learns the mapping from low-resolution to high-resolution images by leveraging the power of deep neural networks.

But the authors of VDSR thought SRCNN has several limitations:

Limited information for learning. Super-resolution (SR) is an ill-posed problem, and solving it requires a significant amount of learning information. Intuitively, the more data available for learning, the higher the potential accuracy of the results. In deep learning, larger receptive fields are associated with higher theoretical accuracy. However, SRCNN consists of only three layers, resulting in a limited receptive field of 13x13 pixels, which restricts its performance.

Slow convergence. Although SRCNN was considered relatively fast compared to other SR algorithms at the time, it still had limitations in meeting practical demands. The training process of SRCNN took up to a week to complete. while the application of SR encompasses various domains, To enable practical usage, it is necessary to improve the convergence speed of the algorithm.

Inability multi-scale. Although the magnification factor is predetermined during training in SRCNN, in real-life scenarios, people may need to enlarge an image by any arbitrary scale (including non-integer factors). SRCNN can only be trained for a specific magnification factor, making it impractical to train a separate SRCNN for every possible factor. Therefore, there is a need for a new network that can handle super-resolution at different scales.

3.2. Proposed Method

3.2.1 Residual Learning

The traditional CNN-based method such as SRCNN [6] is to learn a net that mapping directly from low-resolution to high-resolution, so the net learn the function f given as $\hat{y} = f(x)$, in which x denotes the low-resolution image and $\hat{y}$ denotes the predicted high-resolution image. But the author of VDSR [18] thought that the input low-resolution image and the output high-resolution image are largely similar. Specifically, the low-frequency information of the low-resolution image is similar to the low-frequency information of the high-resolution image. Therefore, it can be said that the low-resolution image and the high-resolution image only differ in terms of the high-frequency components. During the training process, if we focus only on training the high-frequency residual between the high-resolution and low-resolution images, then there is no need to spend too much time on the low-frequency components. To address this, the author introduces the idea of ResNet [12], in which

the net learn $\hat{r} = y - x$.

3.2.2 Deep Networks

The network architecture is deepened to 20 layers, allowing deeper layers to have a larger receptive field. The article uses 3×3 convolutional kernels, and a network with depth D has a receptive field of $(2D + 1) \times (2D + 1)$ in Fig. 1.

The author of this paper believes that a multi-layer network is very helpful in improving the accuracy of super-resolution, for two reasons: (i) a multi-layer network can achieve a very large receptive field. In theory, the larger the receptive field, the more information can be learned and the higher the accuracy. (ii) A multi-layer network can realize complex nonlinear mappings.

3.2.3 Adjustable Gradient Clipping

Due to the computation of the chain rule, when the network has a very large number of layers, the phenomenon of vanishing or exploding gradients often occurs. In the context of image processing, this is reflected in a cliff-like change in the loss function. If the learning rate is set too large, it may lead to excessive modifications and even the loss of previously optimized data like Fig. 2.

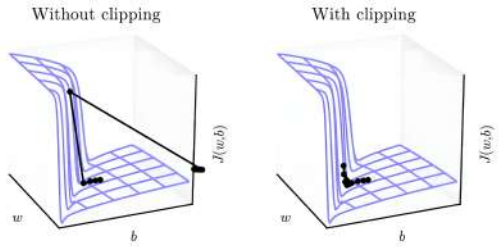

Figure 2. Comparison of without clipping and with clipping

Adaptive gradient clipping adjusts the angle of the gradients based on the learning rate to ensure the stability of convergence in Eq. (3) where θ and λ denote old gradients and current learning rate.

$$g \in [-\frac{\theta}{\lambda}, \frac{\theta}{\lambda}] \tag{3}$$

3.2.4 High Learning Rates

The most direct way to improve convergence speed is to increase the learning rate. However, simply increasing the learning rate can lead to convergence issues and instability. Although the authors have used adaptive gradient clipping to enhance convergence stability, they also employ a technique to enhance convergence stability when setting the learning rate. The specific approach is to initially set a large learning rate (in VDSR, each layer is set with the same learning rate), and then decrease the learning rate by a factor of ten every 20 epochs.

The rationale behind this approach is intuitive. At the beginning of training, the network's loss function is still relatively large, allowing for the use of a high learning rate to expedite convergence. As the training progresses and the model approaches convergence, continuing to use a high learning rate can disrupt the convergence stability. Therefore, a smaller learning rate is used to ensure network convergence. By repeatedly decreasing the learning rate in this manner, the convergence time is further reduced.

4. SRResNet

4.1. Previous work

4.1.1 Design of convolutional neural networks

After the groundbreaking work of Krizhevsky et al. [20], specialized CNN architectures have successfully addressed numerous computer vision problems.

Research has demonstrated that training deeper network architectures can be challenging, but they offer the potential for significant improvements in network accuracy by enabling the modeling of highly complex mappings [26, 28]. To effectively train these deeper architectures, batch normalization [15] is often employed to mitigate internal covariate shift. Deeper network architectures have also exhibited enhanced performance in Single Image Super-Resolution (SISR), as shown by Kim et al. [19]. One such example is the residual network, which introduced the concepts of residual blocks [12] and skip connections [14, 19]. Skip connections alleviate the burden of modeling the identity mapping in the network architecture, a task that is inherently simple but potentially difficult to represent using convolutional kernels.

4.1.2 Loss function

When dealing with the task of recovering lost high-frequency details, such as texture, pixel-wise loss functions like Mean Squared Error (MSE) face challenges in handling the inherent uncertainty. MSE tends to promote pixel-wise averaging of plausible solutions, resulting in overly-smooth outputs with poor perceptual quality.

To address this issue, Mathieu et al. [24] and Denton et al. [5] introduced the use of generative adversarial networks (GANs) [10] for image generation tasks. By incorporating a discriminator loss, they aimed to enhance the perceptual quality of the generated images. Additionally, Yu and Porikli [30] extended the traditional pixel-wise MSE loss by incorporating a discriminator loss to train a network specifically for super-resolving face images with large upscaling factors (8×).

Dosovitskiy and Brox [7] employed loss functions that utilize Euclidean distances calculated in the feature space of neural networks, combined with adversarial training. The study demonstrates that this proposed loss function enables the generation of visually superior images and can address the ill-posed inverse problem of decoding nonlinear feature representations. Similar to their approach, Johnson et al. [16] and Bruna et al. [4] suggest utilizing features extracted from a pretrained VGG network instead of relying on low-level pixel-wise error measurements. Specifically, the authors define a loss function based on the Euclidean distance between feature maps extracted from the VGG19 [26] network. Both super-resolution and artistic style-transfer tasks yielded perceptually more convincing outcomes [8, 9]. In a recent study, Li and Wand [22] also investigated the impact of comparing and blending patches in either pixel or VGG feature space.

4.2. Method

4.2.1 ResNet

ResNet was used for image classification at first, now widely used in image segmentation, target detection, etc. ResNet adds a residual structure to traditional convolutional neural networks, solving the problem of gradient dispersion and accuracy degradation in deeper networks. It enables the network to get deeper and deeper while ensuring accuracy and controlling speed. Fig. 3 shows the schematic diagram of the residual network.

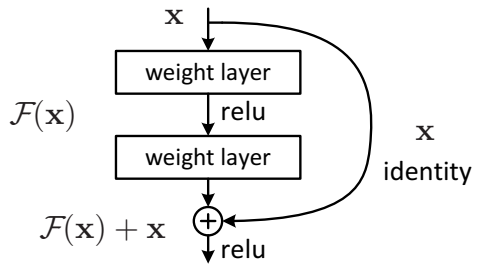

Figure 3. Residual learning: a building block

4.2.2 Sub-pixel convolution layer

Among the current methods of super-resolution reconstruction based on deep learning, for most of the methods, the low resolution (LR) input image is upscaled to the high resolution (HR) space using a single filter, commonly bicubic interpolation, before reconstruction. This means that the super-resolution (SR) operation is performed in HR space, which was shown to be sub-optimal and add computational complexity in Shi et al. [25]. This paper introduce an efficient sub-pixel convolution layer(Fig. 4) which learns an array of upscaling filters to upscale the final LR feature maps into the HR output. By doing so, it effectively replace the handcrafted bicubic filter in the SR pipeline with more complex upscaling filters specifically trained for each feature map, whilst also reducing the computational complexity of the overall SR operation.

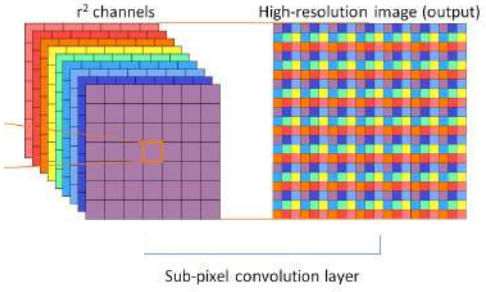

Figure 4. The sub-pixel convolution layer

4.2.3 Parametric Rectified Linear Unit(PReLU)

Rectified activation units (rectifiers) are essential for state-of-the-art neural networks.In Kaiming He et al. [13] the author propose a Parametric Rectified Linear Unit (PReLU) that generalizes the traditional rectified unit, can adaptively learns the parameters of the corrected linear cell and is able to improve accuracy at a negligible additional computational cost.The activation function is defined as:

$$f(y_i) = \begin{cases} y_i, & \text{if } y_i > 0 \\ a_i y_i, & \text{if } y_i \leq 0 \end{cases} \tag{4}$$

Here y_i is the input of the nonlinear activation f on the i th channel, and a_i is a coefficient controlling the slope of the negative part. The subscript i in a_i indicates that we allow the nonlinear activation to vary on different channels. When $a_i = 0$, it becomes ReLU; when a_i is a learnable parameter, we refer to Eq. (4) as Parametric ReLU (PReLU). Fig. 5 shows the shapes of ReLU and PReLU.

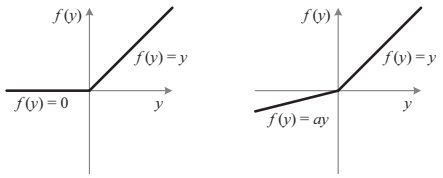

Figure 5. ReLU vs. PReLU. For PReLU, the coefficient of the negative part is not constant and is adaptively learned.

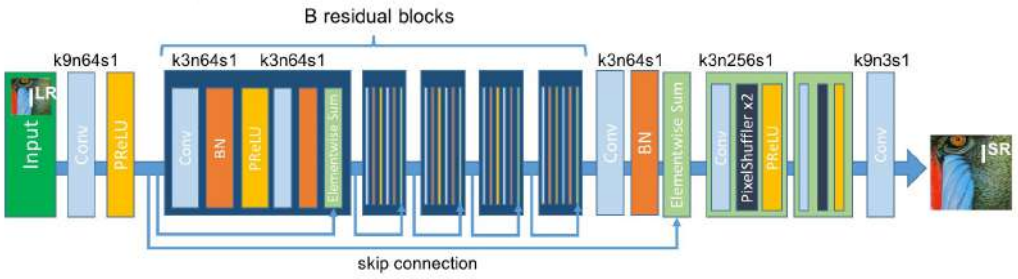

Figure 6. Architecture of SRResNet

4.2.4 Content loss

The pixel-wise MSE loss is calculated as:

$$l_{MSE}^{SR} = \frac{1}{r^2WH}\sum_{x=1}^{rW}\sum_{y=1}^{rH}\left(I_{x,y}^{HR} - G_{\theta_G}\left(I^{LR}\right)_{x,y}\right)^2 \quad (5)$$

This is the most widely used optimization target for image SR. However, while achieving particularly high PSNR, solutions of MSE optimization problems often lack high-frequency content which results in perceptually unsatisfying solutions with overly smooth textures.

Instead of relying solely on pixel-wise losses, author draw inspiration from the works of Gatys et al. [8], Bruna et al. [4], and Johnson et al. [16] to construct a loss function that emphasizes perceptual similarity. We introduce the VGG loss, which is based on the ReLU activation layers of the pre-trained 19-layer VGG network described in Simonyan and Zisserman [26]. Let $\phi_{i,j}$ represent the feature map obtained from the j -th convolution (after activation) before the i -th maxpooling layer within the VGG19 network, which we consider as given. The VGG loss is defined as the Euclidean distance between the feature representations of a reconstructed image $G_{\theta_G}\left(I^{LR}\right)$ and the reference image I^{HR} :

$$l_{VGG/i.j}^{SR} = \frac{1}{W_{i,j}H_{i,j}}\sum_{x=1}^{W_{i,j}}\sum_{y=1}^{H_{i,j}}\left(\phi_{i,j}\left(I^{HR}\right)_{x,y} - \phi_{i,j}\left(G_{\theta_G}\left(I^{LR}\right)\right)_{x,y}\right)^2 \quad (6)$$

Here, $W_{i,j}$ and $H_{i,j}$ represent the dimensions of the corresponding feature maps within the VGG network.

4.2.5 Architecture

The complete SRResNet network architecture is shown in Fig. 6. SRResNet employ the block layout proposed by Gross and Wilber [11]. Specifically, it use two convolutional layers with small 3×3 kernels and 64 feature maps followed by batch-normalization layers [32] and PReLU [13] as the activation function. Throughout the model input and output sections, there is a convolution module for data adjustment and enhancement.In the middle part of the model,there are B residual blocks. Two trained sub-pixel convolution layers near the output [25] are used to increase the resolution of the input image

In Fig. 6 ,k denotes the convolution kernel size, n denotes the number of output channels, and s denotes the step size.

5. Enhanced Deep Super-Resolution Network

5.1. Previous work

As mentioned above,VDSR(Sec. 3) can jointly process super-resolution at multiple scales in a single network. Training VDSR models with multiple scales can significantly improve performance and outperform scale-specific training, which means there is redundancy between scale-specific models. Nonetheless, VDSR-style architectures require bicubic interpolated images as input, which incurs more computation time and memory than architectures with scale-specific upsampling methods.

Although SRResNet(Sec. 4) successfully solves these time and memory issues with good performance, it just adopts the ResNet architecture without much modification. However, the original ResNet was proposed to solve higher-level computer vision problems such as image classification and detection. Therefore, it may not be optimal to directly apply the ResNet architecture to low-level vision problems such as super-resolution.

In order to solve these problems, Lim et al. proposed the Enhanced Deep Super-Resolution Network(**EDSR**) [23] based on the SRResNet architecture. First, they optimized it and simplified the network architecture by analyzing and deleting unnecessary modules. When the model is complex, training the network becomes important. Therefore, the authors train the network with an appropriate loss function and careful model modification at training time. Experiments show that the modified scheme produces better results. At

the same time, in order to utilize scale-independent information during training, the model trains a high-scale model from a pre-trained low-scale model.

5.2. Proposed Method

In this section, we introduce the model architecture of EDSR. We first compare a previously published super-resolution network with an enhanced version based on it with a simpler residual network architecture. Experiments show that the EDSR network outperforms the original network while exhibiting higher computational efficiency.

5.2.1 Residual blocks

Recently, residual networks have shown excellent performance in computer vision problems from low-level to high-level tasks.Although Ledig et al. [21]successfully applied the ResNet architecture to the super-resolution problem with SRResNet, EDSR further improve the performance by employing better ResNet structure.

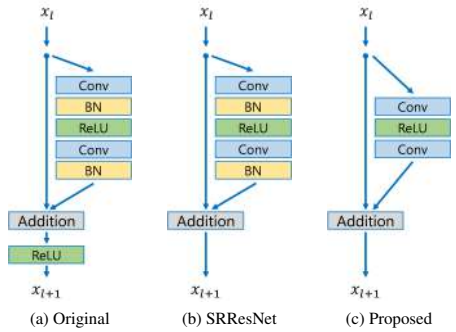

Figure 7. Comparison of residual blocks in original ResNet, SRResNet, and EDSR

We compare the residual blocks of each network model of the original ResNet], SRResNet and EDSR in Fig. 7. It can be seen that EDSR removes the batch normalization layer from the network. Since batch normalization layers normalize features, they get rid of the range flexibility of the network by normalizing features, so it is better to remove them. Experiments show that this simple modification improves performance considerably.

5.2.2 Single-scale model

In convolutional neural networks, model performance can be improved by stacking many layers or increasing the number of filters. A generic CNN architecture with depth (number of layers) B and width (number of feature channels) F takes approximately $O(BF)$ memory and $O(BF^2)$ parameters. Therefore, increasing F instead of B maximizes model capacity when considering limited computing resources.

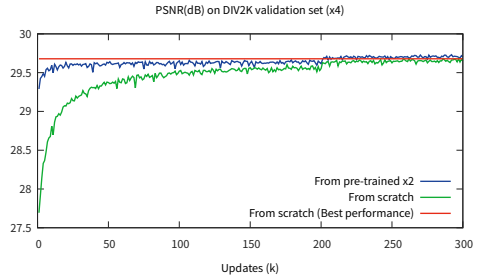

Figure 8. Effect of using pre-trained ×2 network for ×4 model (EDSR). The red line indicates the best performance of green line. 10 images are used for validation during training.

However, increasing the number of feature maps above a certain level makes the training process numerically unstable. A similar phenomenon was reported by Szegedy et al. [27] The authors of EDSR address this issue by employing a residual scaling [27] with a factor of 0.1. In the final single-scale model, the authors extend the baseline model by setting B = 32, F = 256 and a scaling factor of 0.1. The model architecture is shown in Fig. 9.

When training our model for upsampling factors ×3 and ×4, we use the pretrained ×2 network to initialize the model parameters. This pre-training strategy speeds up training and improves final performance. The result is shown in Fig. 8.For upscaling factor ×4, training with a pretrained scale×2 model (blue line) converges much faster than training from random initialization (green line).

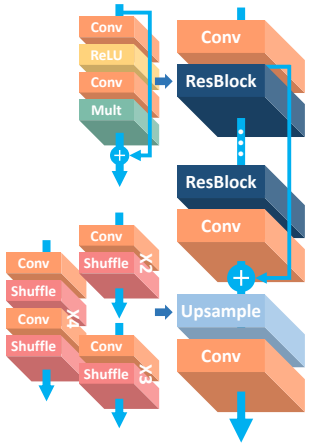

Figure 9. The architecture of EDSR. It is worth mentioning that the **Mult** is used for residual scaling, making the network more stable. The **Upsample** used the method introduced in Sec. 4.2.2

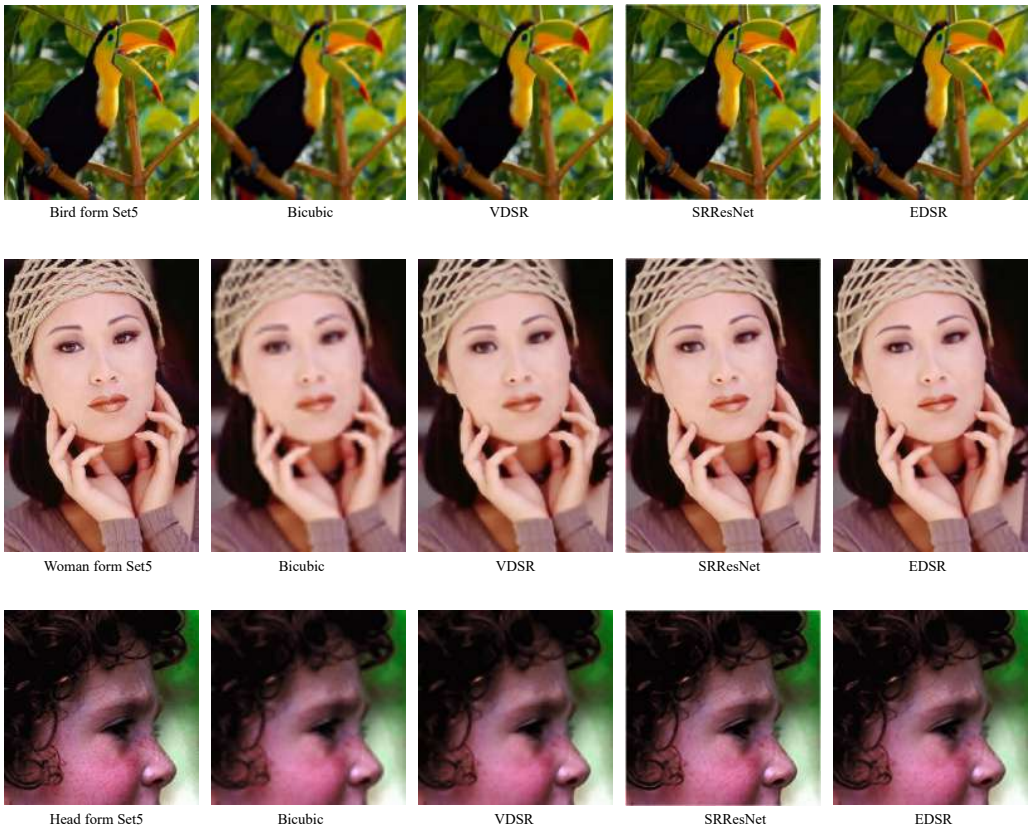

Figure 10. Qualitative comparison of four methods on ×4 super-resolution.

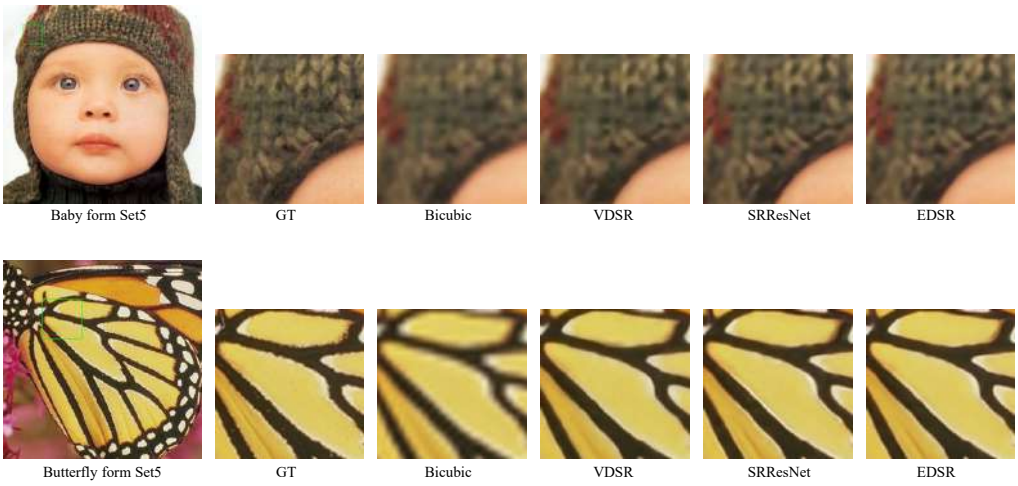

Figure 11. The results after partial zoom-in

6. Experiments

6.1. Datasets

Set5 [3] is a classic image dataset for image super-resolution reconstruction tasks. We run the network models corresponding to the three methods on this dataset, and compare the PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity) of the output images to obtain and compare the performance of the three methods.We also compare the performance on the other dataset:Set14 [31].

Dataset	Scale	Bicubic	VDSR	SRResNet	EDSR
Set5	× 2	33.66/0.9299	37. 39/0.9845	-/-	38.11/0.9634
	× 3	30.39/0.8682	32.98/0.9244	-/-	34.74/0.9247
	× 4	28.42/0.8134	31.87/0.8845	32.05/0.9034	32.87/0.8956
Set14	× 2	30.24/0.8688	33.53/0.9044	-/-	33.92/0.9195
	× 3	27.55/0.7742	29.75/0.8432	-/-	30.52/0.8462
	× 4	26.00/0.7027	27.89/0.7677	28.43/0.7834	28.80/0.7876

Figure 12. The average PSNR and SSIM

6.2. Evaluation

We set the magnification scale of super-resolution reconstruction as $\times 4$, and the results of super-resolution reconstruction on three models are shown in the Fig. 10. In Fig. 11, we enlarge some areas of the results, making the performance difference between the three methods and the bicubic interpolation algorithm more obvious. In Fig. 12, we list the average PSNR and SSIM of the three methods and the bicubic interpolation algorithm on the Set5 dataset, which can quantitatively compare the performance of the four methods.

7. Conclusion

In this paper, we have provided a comprehensive overview and analysis of deep convolutional networks for single image super-resolution. We compared three state-of-the-art super-resolution Convolutional Neural Networks (CNNs) including VDSR, SRResNet, and EDSR, and evaluated their performance on different scales.

We observed that deep CNN-based super-resolution methods have shown remarkable performance improvements over traditional interpolation-based methods. By leveraging large-scale datasets and powerful network architectures, these methods have demonstrated the ability to generate highly realistic and visually appealing high-resolution images.

Despite the significant progress in recent years, there are still several open problems and limitations in existing techniques. For example, the generalization ability of these methods to images with diverse content and characteristics is still an ongoing research challenge. Additionally, the computational cost of training and inference in deep CNN-based super-resolution methods remains high, limiting their real-time applicability in certain scenarios.

In conclusion, deep convolutional networks have revolutionized single image super-resolution and continue to push the boundaries of what is possible. Future research should focus on addressing the remaining challenges and limitations to further improve the accuracy, efficiency, and generalization ability of these methods.

References

[1] Saeed Anwar and Nick Barnes. Densely residual laplacian super-resolution. *arXiv preprint arXiv:1906.12021*, 2019. 1

[2] Saeed Anwar, Salman Khan, and Nick Barnes. A deep journey into super-resolution: A survey. *ACM Computing Surveys (ACMCSUR)*, 53(3), May 2020. 1

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 7

[4] Joan Bruna, Pablo Sprechmann, and Yann LeCun. Super-resolution with deep convolutional sufficient statistics. *arXiv preprint arXiv:1511.05666*, 2015. 4, 5

[5] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. *Advances in neural information processing systems*, 28, 2015. 3

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. *ECCV*, 2014. 2

[7] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29, 2016. 4

[8] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28, 2015. 4, 5

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 4

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3

[11] S. Gross and M. Wilber. Training and investigating residual nets. online. http://torch.ch/blog/2016/02/04/resnets.html 2016. 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 2, 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 4, 5

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016. 3

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 3

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. 4, 5

[17] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981. 1

[18] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. *CVPR*, 2016. 1, 2

[19] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016. 3

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017. 3

[21] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, 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*, pages 4681–4690, 2017. 1, 6

[22] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2479–2486, 2016. 4

[23] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. *CoRR*, abs/1707.02921, 2017. 1, 5

[24] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 3

[25] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 4, 5

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 4, 5

[27] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016. 6

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 3

[29] LeCun Y., Boser B., Denker, J.S., Henderson D., Howard, R.E., and L.D. Hubbard W., Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, pages 541–551, 1989. 1

[30] Xin Yu and Fatih Porikli. Ultra-resolving face images by discriminative generative networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, pages 318–333. Springer, 2016. 3

[31] Lei Zhang and Xiaolin Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE transactions on Image Processing*, 15(8):2226–2238, 2006. 7