Single Image Super-Resolution via Squeeze and Excitation Network

Tao Jiang[123]
fengqingyunyang@gmail.com
Yu Zhang*[123]
zhangyu82@snnu.edu.cn
Xiaojun Wu*[123]
xjwu@snnu.edu.cn
Yuan Rao[4]
raoyuan@mail.xjtu.edu.cn
Mingquan Zhou[3]
mqzhou@bnu.edu.cn

[1] Key Laboratory of Modern Teaching Technology, Ministry of Education, Shaanxi Normal University, Xi'an 710062, China
[2] Engineering Laboratory of Teaching Information Technology of Shaanxi Province, SNNU , Xi'an 710119, China
[3] School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
[4] School of Software School, Xi'an Jiaotong University, Xi'an 710049, China
(* Corresponding Author)

Abstract

Single Image Super-Resolution (SISR) has obtained unprecedented breakthrough with the development of Convolutional Neural Networks (CNN). A majority of these methods try to increase the depth of the network to obtain a larger receptive field. However, we found that blindly stacking feature maps and the simple cascading structure can not achieve a high rate of utilization in super-resolution reconstruction. In this paper, we propose a refined fully convolutional network for Single Image Super Resolution. Based on the assumption that features maps from different depths or the same depth but different channels have different contributions during image reconstruct, we introduce the Squeeze and Excitation (SE) network to evaluate the importance of different feature maps while building the network. Besides, densely connection operation is also conducted in the framework for a better use of the contextual information and feature maps. Extensive experiments demonstrates that the proposed method can enhance the restoration performance and achieve the state-of-the-art results in super-resolution task.

1 Introduction

Image super-resolution reconstruction (SR) is a classic problem in computer vision. It mainly contains two subfields: *Single Image Reconstruction* and *Sequence Images Reconstruction*. In this paper, we mainly talk about the *Single Image Super Resolution* (SISR), which aims to recover a high resolution image from its degraded one [7]. The task can be formulated as

$$Y_{hr} = \Theta(X_{lr}) \tag{1}$$

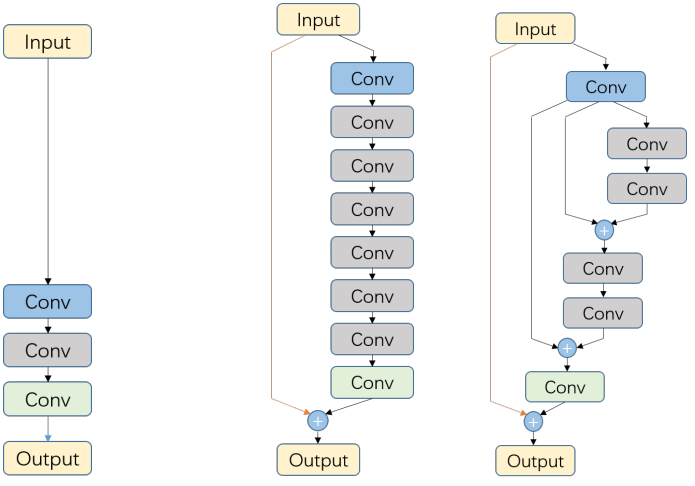


Figure 1: Simple outlines of SRCNN, VDSR (8 convolutional layers) and DRRN (2 recursive blocks). Block **Conv** denotes a standard convolutional block which consists of Batch Normalization (BN), convolution (CN) and ReLU in order. While in DRRN, it represents a pre-activation convolutional block contains BN, ReLU and CN in sequence. These architectures can be divides into 3 modules: Features Extraction (blue), Nonlinear Mapping (grey) and Image Reconstruction (green).

in which X_{lr} and Y_{hr} denote the low-level resolution input image and high-level resolution output image, respectively. Function Θ is an abstract description referring to the reconstruction process. Considering that the degraded image losses plenty of contextual information and details, SISR is a typical ill-posed inverse problem.

In the early time, interpolation methods such as bicubic interpolation and Lanczos resampling [2], more technically, statistical image priors [9, 17], are used in SISR. Currently, with the prosperity of machine learning especially deep learning technology, excellent methods based on deep learning [1, 8, 10, 18] have shown a dominant performance in this field.

Dong *et al.* first proposed a convolutional network structure (SRCNN) to generate better reconstructed images utilizing the bicubic interpolation results as input [1]. Compared with the previous methods, SRCNN can take good advantage of the contextual information through its simple architecture when recovering the high resolution (HR) image. It mainly consists of three convolutional layers which called patch extraction layer, non-linear mapping layer and reconstruction layer, respectively. It is worth noticing that these three convolutional layers in SRCNN stand for three different modules: features extracting module, features mapping module and image reconstructing module and there are still many approaches that continue utilizing these modules nowadays. Excluding the succeed of SRCNN in SISR, its three-layer structure can not make fully use of the inner information of the input and quantities of experiments indicate that the model is hard to learn during training phase.

To deal with this issue, Kim *et al.* introduced a very deep convolutional networks for super-resolution (VDSR) [8]. In this work, the authors built a deep model with 20 layers acting as the features mapping module. Different from the previous methods, they draw on the experience of residual learning and tried to let the model mainly focus on learning the

Method	Mathematical Formulation
SRCNN[1]	$y = f_3(f_2(f_1(x)))$
VDSR[8]	$y = f_{Rec}(f_n(f_{n-1}(...f_1(x)...))) + x$
DRRN[13]	$y = f_{Rec}(R_N(R_{N-1}(...R_1(x)...))) + x$

Table 1: Three related methods mentioned in Sec.2. function f denotes convolutional layer and R denotes recursive block.

residual part of the input. Experiments shows that VDSR shows a better performance than SRCNN. Moreover, the residual learning part in VDSR makes it easy to converge during the training phase. Although VDSR has gained great progress in SISR, it has it own limitation: feature maps far from the reconstructing module are less considered when restoring the residual part of the input patch.

Recently, a new approach utilizing *Deep Recursive Residual Network* (DRRN) has been introduced for single image reconstruction [13]. The network continues learning the residual parts w.r.t the input patches. In DRRN, Tai *et al.* firstly utilized a convolutional layer to extract the original features, and then used these features to make a constrain to the feature maps produced by the feature mapping layers. To be specific, the authors proposed an stacked recursive blocks structure in a cascading way. Each of these recursive blocks are trained to learn the residual part of the original features. By adopting these methodologies, DRRN provided an improved performance compared with VDSR.

In this paper, we design a *Squeeze and Excitation Residual Network* for super resolution (SENSR). The model is built to learn the residual part of an input image. Based on the assumption that feature maps from different layers have different contributions to the restoring module, we introduce the key component of *Squeeze and Excitation Network* [5] for the importance evaluation. By labeling the feature maps with different contribution rates, the residual reconstructing part can handle these feature maps in a smart way. Compared with VDSR, the proposed method can make better use of these intermediate feature maps and achieve the sate-of-the-art performance.

2 Related Work

In this section we mainly discuss three mainstream methods mentioned in Sec.1: SRCNN, VDSR and DRRN.

2.1 SRCNN

SRCNN [1] is the first work that introduced deep learning to the super-resolution reconstruction filed. It only has 3 separate layers, which called patch extraction and representation layer, non-linear mapping and reconstruction layer, respectively. The first layer is to extract high dimensional vector from the low-resolution input patch and the second is about the feature mapping module that maps the original vector to another high dimensional vectors. The last reconstruction module aims to generate the high-resolution output image. These three layers have filters of spatial sizes of 9×9, 1×1 and 5×5, separately, which enable the neurons of the last layer has a limited receptive size (13×13). SRCNN has gained the state-of-the-art performance when proposed. However, It is because of the shallow network depth that puts its learning ability into a bottleneck.

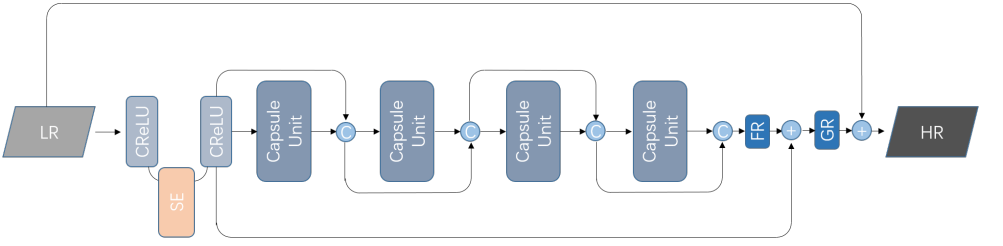


Figure 2: A simplified cascading architecture of proposed methodology with 4 capsule units. **SE** denotes a squeeze and excitation component. **FR** and **GR** stand for features residual and global residual, respectively. Symbol **C** represents a concatenation operation.

2.2 VDSR

To strengthen the learning ability of *Deep Learning* based model, Kim *et al.* proposed a very deep convolutional network (VDSR) in [8]. The authors argued that increasing network′s depth can significantly get a better features representation and boost the performance. Motivated by [4], they use a residual structure to make the net much deeper (20 layers). Compared with SRCNN, VDSR has a larger receptive field (41×41). Besides, VDSR introduces an conception of Global Residual Leaning (GRL), which architecture enables the net to learn the residual parts of the input image. In addition, the VDSR model is robust to multi-scale image reconstruction due to the scaling augmentation and experiments shows that it provides an excellent performance in image restoration.

2.3 DRRN

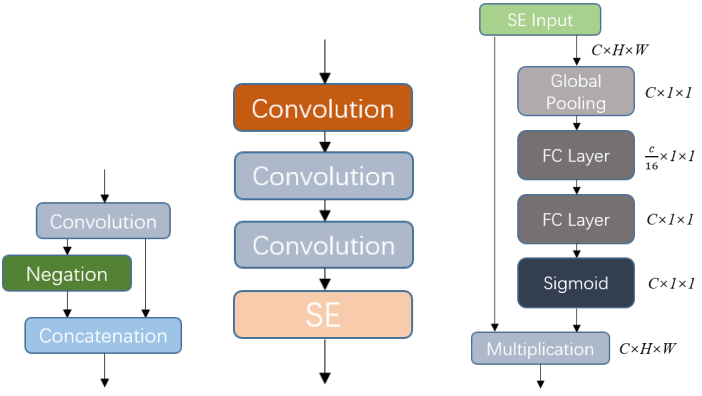
The architecture of *Deep Recursive Residual Network* (DRRN) [13] can be regarded as an recursive residual network, in which all the outputs of residual blocks is added to the original feature maps extracted by a simple convolutional block. In contrast to these traditional networks, Tai et al. utilized an pre-activation structure described in [4] for a better performance. Same as [8], the network aims to learn the residual parts of the input patch. Inspired by [11], they use a multi-path mode in DRRN to facilitate the learning and prevents overfitting. Compared with VDSR, DRRN can get the state-of-the-art performance with fewer parameters.

3 Proposed Method

Here we talk about the details of our proposed method. The network takes the bicubic interpolated LR image as input and then predict the HR output. This section includes 3 parts: global residual learning, capsule unit and an overview of the network outline.

3.1 Global Residual Learning

Same as [8], we follow the idea of *Global Residual Learning* in VDSR. In contrast to the origin implementation of *Global Residual* (GR), we divide the residual learning module into two stages: (I) features residual learning and (II) global residual learning.



(a) CReLU (b) Capsule Unit (c) SE Component

Figure 3: Structure of CReLU, Capsule Unit and SE Component. Convolution here denotes a convolutional block contains a batch normalization layer, a ReLU layer and a convolutional layer in sequence.

In the proposed model, we use an concatenated ReLU (CReLU) structure introduced by [13] for the origin features extraction. The proposed CReLU structure can extract more complete features compared with the traditional convolutional layers as Shang *et al.* argued that there is a negative correlation among the filters in the lower convolutional layers. The origin feature extraction can be formulated as:

$$\gamma = f_{\mathrm{CReLU}}(f_{\mathrm{SE}}(f_{\mathrm{CReLU}}(x))) \tag{2}$$

where γ denotes the origin features, f_{CReLU} stands for an concatenated ReLU unit and f_{SE} is the symbol of *Squeeze and Excitation* component.

For the features residual learning, the cascading path is to learn the residual parts of the origin features. And then construct the global residual part through these features residual (see in Fig. 2), which can be formulated as:

$$y = f_{\mathrm{GR}}(f_{\mathrm{FR}}(\varphi(\gamma)) + \gamma) + x \tag{3}$$

where f_{GR} and f_{FR} are the global residual learning layer and features residual learning layer, respectively. φ denotes the cascading Capsule Unit.

3.2 Capsule Unit

As illustrated in Fig 2, the cell called capsule is the basic unit in our proposed method. It contains three convolution blocks and a squeeze and excitation component (see in Fig 3.b).

The first convolutional block in Capsule is used for decreasing the dimension of feature maps. From Fig 2, we can observe that each of the Capsule units will take the concatenation of the previous Capsule's outputs as input. The dimension of Capsule input will arise linearly with the increasing depth. To tackle this issue, we utilize a convolutional layer (Transition Down Block) with a spatial size of 1×1 to unified the input dimension.

The 2nd and 3rd convolutional layers are designed for the feature mapping process. In each capsule, we use 2 cascading convolutional layers as the feature mapping unit.

The output of Capsule will pass through a Squeeze and Excitation component. The SE part is design for evaluating the produced feature maps. To be specific, it labels each of the feature maps with a list of scores ranging from 0 to 1. The score denotes the importance of each feature map.

As is demonstrated in Fig 3.c, the structure of SE component is simple and easy to implementation. The fist global pooling layer is utilized for squeeze every two dimension feature map to a single number. Each of the numbers has a complete receptive field for the channel they are in. Then we use a one dimensional encode and decode operation for the information interaction. While implementing this procedure, we use two fully connected layer to simplify the operation. Next, we exploit the sigmoid activation to compress the output to (0, 1). Finally, we re-adjust the feature maps by multiplying the corresponding scores.

3.3 Network Outline

Here we give an overview of the proposed network (Fig 2).(1)We utilize the bicubic interpolation result of a patch as the input of the network.(2)These two concatenation ReLU units act as the origin features extraction module.(3)There are a number of cascading Capsule Unit stacked for the origin features residual learning.(4)The global residual of the input are computed through the origin features residual part.

4 Experiment

In this section, we are going to talk about the data we choose and the training details in the experiment.

4.1 Training Data

As with the previous methods, we turn the image into Ycbcr format, and then utilized the model to learn the mapping relations of the luminance layer. The training data contains 291 images, where 91 images are from [19] and the others are contributed by [12]. In addition, data augmentation technics are used in our experiment. For generating the training set, we rotate the 291 images by 90°, 180°, 270° and make a symmetry transformation on them. Then, we clip these images into 32×32 patches for training. Multi-scaling augmentation (×2, ×3, ×4) is also introduced to the training set to enhance the robustness to diverse scaling inputs. There are 4 popular data sets used for the performance evaluation: Set5, Set14, BSD100 and Urban100.

4.2 Training

We implement the proposed model with Caffe and utilize Stochastic Gradient Descent (SGD) as the optimizer.

4.2.1 Gradient Clipping

The adjustable gradient clipping technique is usually used for recurrent networking training to avoid gradient explosion [13] and it has been conducted in some of the previous methods

Data Set		PSNRs/SSIMs							
		Bicubic	SRCNN[8]	SelfEx[8]	RFL[15]	VDSR[8]	DRRN_B1U9[23]	DRRN_B1U25[23]	SENSR (ours)
SET5	×2	33.66/0.9299	36.66/0.9542	36.49/0.9537	37.53/0.9587	37.59/0.9591	37.66/0.9589	37.74/0.9591	37.80/0.9596
	×3	30.39/0.8682	32.75/0.9090	32.58/0.9093	33.66/0.9213	33.78/0.9226	33.93/0.9234	34.03/0.9244	33.98/0.9242
	×4	28.42/0.8104	30.48/0.8628	30.31/0.8619	31.35/0.8838	31.44/0.8849	31.58/0.8864	31.68/0.8888	31.67/0.8883
SET14	×2	30.23/0.9384	32.45/0.9067	32.22/0.9034	33.03/0.9124	33.13/0.9132	33.19/0.9133	33.23/0.9136	33.25/0.9141
	×3	27.54/0.7736	29.30/0.8215	29.16/0.8196	29.77/0.8314	29.85/0.8330	29.94/0.8339	29.96/0.8349	29.95/0.8343
	×4	26.00/0.7018	27.50/0.7513	27.40/0.7518	28.01/0.7674	28.09/0.7689	28.18/0.7701	28.21/0.7720	28.22/0.7709
BSD100	×2	29.56/0.8431	31.36/0.8879	31.18/0.8855	31.90/0.8960	31.96/0.8965	32.01/0.8969	32.05/0.8973	32.05/0.8972
	×3	27.21/0.7388	28.41/0.7863	28.22/0.7806	28.82/0.7976	28.87/0.7988	28.91/0.7992	28.95/0.8004	28.95/0.7999
	×4	25.96/0.6674	26.90/0.7101	26.75/0.7054	27.29/0.7251	27.31/0.7259	27.35/0.7262	27.38/0.7284	27.39/0.7271
URBAN100	×2	26.87/0.8401	29.50/0.8946	29.11/0.8904	30.76/0.9140	30.95/0.9159	31.02/0.9164	31.23/0.9188	31.20/0.9180
	×3	24.46/0.7344	26.24/0.7989	25.86/0.7900	27.14/0.8279	27.26/0.8309	27.38/0.8331	27.53/0.8378	27.53/0.8357
	×4	23.14/0.6570	24.52/0.7221	24.19/0.7096	25.18/0.7524	25.25/0.7552	25.35/0.7576	25.44/0.7638	25.49/0.7612

Table 2: Performance on four benchmark data sets

[8, 13]. Specifically, it helps to clip the gradients to a predefined range $[-\frac{\xi}{r}, \frac{\xi}{r}]$, where ξ is a predefined scalar and r denotes the current learning rate. We keep this trick in our implementation to avoid gradient explosion and accelerate the training phase.

4.2.2 Multi-Scaling Augmentation

Reference [8] is the first work that introduced the concept that a single model is robust to different scaling factors. In order to follow this principle, we mix up the patches from different scaling factors when generating training set.

4.2.3 Training

We use **Caffe** to implement SENSR on a PC with 4 Nvidia 1080Ti GPUs. While in the experiment, we stacked 16 Capsule units to training the model with the batch size of 16. SGD with momentum of 0.9 is utilized as the optimizer for the gradient back propagation. The initial learning rate and weight decay are set to 0.1 and 0.0001, respectively. The spatial filter sizes of convolutional layers except the transition down layers are set to $128 \times 3 \times 3$. For the weights initialization, we adopt the method introduced in [3]. To achieve the goal of facilitating training phase and avoiding gradient explosion, adjustable gradient clipping technique is a must. This method makes it possible to set a higher initial learning rate (0.1) to accelerate the convergence process.

5 Comparison with State-of-the-Art Models

In comparison with previous methods, we introduced Urban100 for the performance evaluation except for the three widely used benchmark data sets (Set5, Set14 and BSD100) in super resolution field. As for evaluating criterion, both PSNR (Peak signal-to-noise ratio) and SSIM (Structure Similarity) are considered as the comparison standards. Besides, we introduce a extra criteria called *Information Fidelity Criterion* (IFC) [16] for a complete evaluation.

We provide quantities of statistics in Tab 2, in which the we re-trained the VDSR model and the rest scores are cited from [13]. Besides, we further make an comparison of IFC among these methods in Tab 3. From these two tables, we observe that the proposed method achieves a competitive performance with the former models.

Data Set		Bicubic	SRCNN[8]	SelfEx[8]	RFL[22]	VDSR[8]	DRRN_B1U9[23]	DRRN_B1U25[23]	SENSR (ours)
SET5	×2	6.083	8.036	7.811	8.556	8.569	8.583	8.671	8.821
	×3	3.580	4.658	4.748	4.926	5.221	5.241	5.397	5.468
	×4	2.329	2.991	3.166	3.191	3.547	3.581	3.703	3.757
SET14	×2	6.105	7.784	7.591	8.175	8.178	8.181	8.320	8.371
	×3	3.473	4.338	4.371	4.531	4.730	4.732	4.878	4.882
	×4	2.237	2.751	2.893	2.919	3.133	3.147	3.252	3.269
URBAN100	×2	6.245	7.989	7.937	8.450	8.645	8.653	8.917	9.047
	×3	3.620	4.584	4.843	4.801	5.194	5.259	5.456	5.516
	×4	2.361	2.963	3.314	3.110	3.496	3.536	3.676	3.737

Table 3: IFCs of different models on benchmark data sets

Data Set		PSNRs/SSIM/IFC		
		SENSR_No_Dense	SENSR_No_SE	SENSR_FULL
SET5	×2	37.54/0.9594/8.3226	37.70/0.9600/8.7069	37.80/0.9596/8.821
	×3	33.69/0.9224/5.1241	33.86/0.9239/5.3730	33.98/0.9242/5.468
	×4	31.34/0.8865/3.5450	31.48/0.8890/3.6911	31.67/0.8883/3.757
SET14	×2	33.07/0.9130/7.9954	33.17/0.9138/8.3028	33.25/0.9141/8.371
	×3	29.82/0.8328/4.7038	29.89/0.8337/4.8175	29.95/0.8343/4.882
	×4	28.05/0.7681/3.1485	28.12/0.7690/3.2032	28.22/0.7709/3.269
URBAN100	×2	30.91/0.9152/8.4958	31.15/0.8175/8.9312	31.20/0.9180/9.047
	×3	27.25/0.8298/5.2112	27.40/0.8323/5.4195	27.53/0.8357/5.516
	×4	25.29/0.7553/3.5317	25.39/0.7577/3.6417	25.49/0.7612/3.737

Table 4: Ablation studies of the proposed Squeeze and Excitation Network.

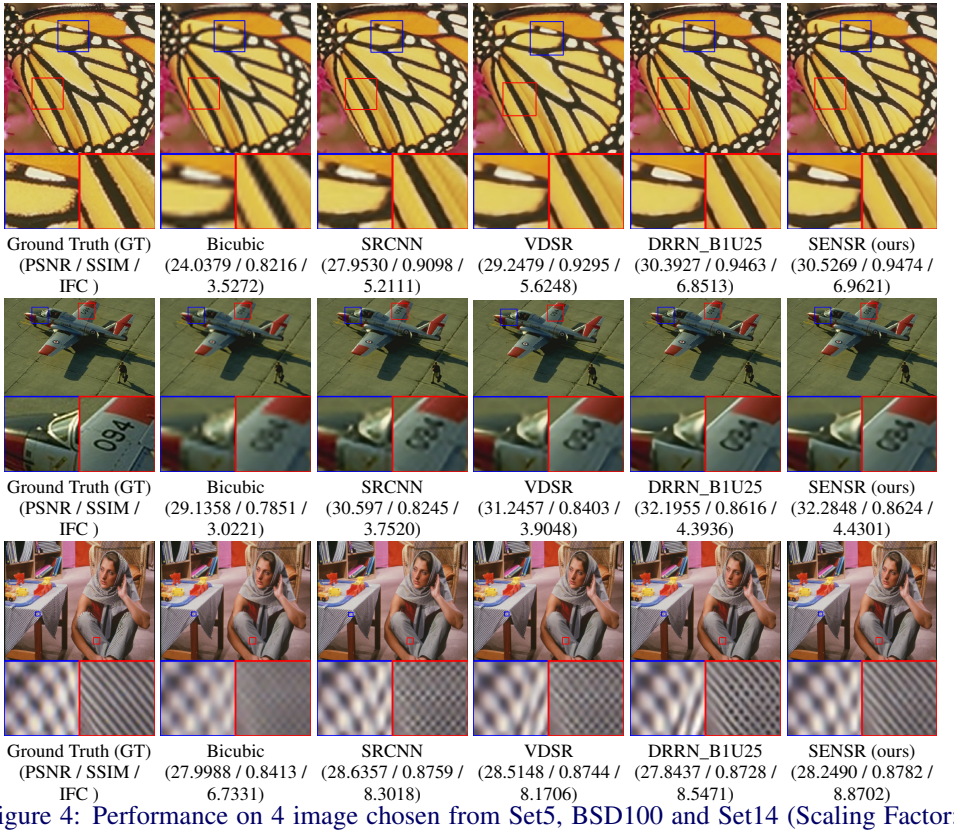


Figure 4: Performance on 4 image chosen from Set5, BSD100 and Set14 (Scaling Factor: rows 1,2 in ×3 and 4,5 in ×2).

5.1 Comparison with VDSR

As illustrated in Fig 1, The simple cascading structure in VDSR can not make fully use of these feature maps produced by the feature mapping module. In order to mitigate this issue, we propose *Capsule Unit* acting as the feature mapping unit and then concatenate the outputs of all the capsule units when reconstructing the features residual. Based on this theory, we can see that the performance of SENSR has an overwhelming advantage in all the assessment indicators compared with VDSR.

5.2 Comparison with DRRN

Tai *et al.* proposed 2 model with different network depth in [18] and we make a contrast with these 2 models. It is obvious that the scores of SENSR are superior to the DRRN_B1U9 model with different scaling factors. While compared with DRRN_B1U25, SENSR can also achieve a comparable performance. Moreover, when referring to the criterion of *Information Fidelity Criterion* (Tab 3), SENSR has an absolute advantage over the performance of these 4 benchmark data sets. Further more, the IFCs of SENSR on the Urban100 which contains 100 images exceed those in DRRN_B1U25 by 0.13, 0.06 , 0.061 with the scaling factors of $\times 2$, $\times 3$ and $\times 4$, respectively.

Fig 4 presents the comparison of the construction details in the testing data. The fist two images are selected form Set5 and BSD100 with scaling factor 3. We can see that SENSR can fully utilized the contextual information and achieve state-of-the-art performance compared with the previous method. The rest two images are under the scaling factor 2. From the recovering images in row 4, we find that the proposed method can handle the complex texture details well while the results of the other methods have varying degrees of distortion. Considering that the loss of contextual information will grow exponentially with the increase of scaling factor, how to ensure that the recovering details are not distorted is still a knotty problem.

5.3 Further Discussion

We also conducted ablation studies on the proposed *Squeeze and Excitation Network*. During the experiments, we removed the Squeeze and Excitation Component and Dense Connection separately to identify the roles they act in the proposed network (see in Tab 4). It is obvious that after removing the Dense Connection pathes, its performance was almost the same as that of VDSR. We found that the Dense Connection pathes can significantly improve the reconstruction performance, and the introduction of Squeeze and Excitation Component can further enhance PSNR/SSIM/IFC scores.

6 Conclusions

In this paper, we propose a refined and effective network for the *Single Image Super Resolution*. In *Super Resolution* field, the current mainstream approach is to gain greater receptive field by deepening network depth. However, we find that with the increase of network depth, the influence of the shallow level features on the image reconstruction layer is weakened. To resolve this issue, we introduce a *Capsule Unit* and connect these capsule outputs directly to the reconstruction layer. There is a tiny but elegant *Squeeze and Excitation* component inside

the capsule unit acting as an importance assessment system for the produced feature maps. By utilizing the features concatenation operation, SENSR establishes a strong association with the features from different level and can easily make better use of all these feature maps while recovering the residual outputs. Experimental results show that the proposed model can achieve the state-of-the-art performance and exhibits robustness to complex texture images.

Despite the promising performance in the Super Resolution field, there are still a lot of problems remain to be solved. The current model can not handle the complex the images with complex textures well as the low resolution image only retains high-frequency information and lose most of the details. How to use the model to learn extra knowledge to infer details when restoring the high resolution image is still worth a continuous work.

Acknowledgement

This work is supported by National key research and development program (2017YFB1402102), National Natural Science Foundation of China (11772178, 61741208), Shaanxi Natural Science Foundation of China (2017JQ6074), Fundamental Research Funds for the Central Universities (GK201801004, GK201803089).

References

[1] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

[2] Claude E Duchon. Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology*, 18(8):1016–1022, 1979.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[5] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017.

[6] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015.

[7] Michal Irani and Shmuel Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 53(3):231–239, 1991.

[8] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016.

[9] Kwang In Kim and Younghee Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE transactions on pattern analysis and machine intelligence*, 32(6):1127–1133, 2010.

[10] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 624–632, 2017.

[11] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3367–3375, 2015.

[12] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416–423. IEEE, 2001.

[13] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[14] Samuel Schulter, Christian Leistner, and Horst Bischof. Fast and accurate image upscaling with super-resolution forests. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3791–3799, 2015.

[15] Wenling Shang, Kihyuk Sohn, Diogo Almeida, and Honglak Lee. Understanding and improving convolutional neural networks via concatenated rectified linear units. In *International Conference on Machine Learning*, pages 2217–2225, 2016.

[16] Hamid R Sheikh, Alan C Bovik, and Gustavo De Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Transactions on image processing*, 14(12):2117–2128, 2005.

[17] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[18] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.