Image Super Resolution Based on Fusing Multiple Convolution Neural Networks

Haoyu Ren, Mostafa El-Khamy, Jungwon Lee
SAMSUNG SEMICONDUCTOR INC.
4921 Directors Place, San Diego, CA, US
{haoyu.ren, mostafa.e, jungwon2.lee}@samsung.com

Abstract

In this paper, we focus on constructing an accurate super resolution system based on multiple Convolution Neural Networks (CNNs). Each individual CNN is trained separately with different network structure. A Context-wise Network Fusion (CNF) approach is proposed to integrate the outputs of individual networks by additional convolution layers. With fine-tuning the whole fused network, the accuracy is significantly improved compared to the individual networks. We also discuss other network fusion schemes, including Pixel-Wise network Fusion (PWF) and Progressive Network Fusion (PNF). The experimental results show that the CNF outperforms PWF and PNF. Using SRCNN as individual network, the CNF network achieves the state-of-the-art accuracy on benchmark image datasets.

1. Introduction

Image Super Resolution (SR) is a process for recovering a High-Resolution image (HR) from a single Low-Resolution (LR) image. Most modern single image super-resolution methods rely on machine learning techniques. These methods focus on learning the relationship between LR and HR image patches. A popular class of such algorithms uses an external database of natural images to extract training patches, and then constructs an LR to HR mapping. Various learning algorithms have been used, including nearest neighbor approaches [22], manifold learning [7], dictionary learning [11][10], and locally linear regression [34]. Recently, the convolutional neural networks become popular.

Many CNNs are introduced into SR in these years. Some of them utilize very deep networks [17][18] or specific network architectures such as deep ResNet [19][20]. These works ensure high-accuracy but result in a relatively large network size and low efficiency. The training is also difficult due to the hyperparamter tuning, especially for large diverse training set. Other researchers adopt less layers and simple network structure, such as SRCNN [13]. Training such network is relatively easy, even with random weight initialization and very large training sets. In contrast, the accuracy of these networks will be lower.

In this paper, we discuss how to construct a super resolution system based on fusing different individual networks. Our contributions are two folds. First, we propose a Context-wise Network Fusion (CNF) framework to fuse multiple individual networks. Each individual network constructs a mapping from LR to SR space. Since the output feature maps of the individual networks might have different context characteristic, additional convolution layers are applied on these feature maps. The outputs of these convolution layers are summed as the final output. CNF has no limitations on the architecture of the individual networks, so it could be used for fusing any type of SR networks. Second, we discuss other network fusion schemes, including Pixel-Wise network Fusion (PWF), and Progressive Network Fusion (PNF). In the experiments, multiple layers are added into SRCNN [13] as individual networks. The results show that the proposed CNF network significantly outperforms other fusion schemes. The CNF constructed by three SRCNNs achieves the state-of-the-art performance, with acceptable efficiency.

The rest of this paper is structured as follows. Section 2 gives the related works on super resolution. Section 3 describes different ways of network fusion. Section 4 gives the implementation details of training individual networks and fused networks. Section 5 shows the experimental results on benchmark image dataset, Set5, Set14, BSD100, and NTIRE challenge dataset. Conclusions are in the last section.

2. Related Work

In these years, many SR methods have been developed by the computer vision community [1][2]. Early methods use efficient interpolations such as bilinear, bicubic, or Lanczos filtering [21]. These filtering algorithms can generate smooth HR outputs, which however lack high frequency information. Later, structural and shape prior are introduced to enhance these interpolations [5][6][24]. Although they might retrieve the details of a few edges and contours from the smooth output, the overall quality is still low.

Later, the learning based methods are widely used to construct a mapping from the LR space to the HR space. Some approaches focus on the internal patch redundancy [3][23]. Glasner et al. [3] exploited such redundancies across scales and integrated the example-based SR with multi-image SR. Huang et al. [23] utilized the geometric variations and explicitly localized planes in the scene to expand the internal patch search space. The detected perspective geometry was adopted to guide the patch search process. Others prefer using external dictionaries to learn regression functions to represent this mapping. The typical methods include neighbor embedding [7], Markov network [22], kernel regression [4], random forest [26], sparse coding [25][9][10][12], and convolution neural networks.

In the pioneering work SRCNN [13][15], Dong et al. introduced a CNN constructing a mapping from the bicubic upsampled LR space to HR space. SRCNN is relatively efficient, but the accuracy is limited due to the 3-layer structure. To enhance the accuracy, some researchers focus on designing more complicate networks. Kim et al. [17] utilized residual learning and trained a 20-layer network with small filters and gradient clipping. Wang et al. [27] integrated CNN with sparse representation prior. The network was trained by the learned iterative shrinkage and thresholding algorithm. Others focus on utilizing perceptual loss function [31] instead of the Mean Square Error (MSE) to get the HR results similar to natural images. Sonderby et al. [30] proposed a CNN network based on maximum posterier estimation. Ledig et al. [19] employed a deep residual network (ResNet) as the discriminator of the Super Resolution Generative Adversarial Network (SRGAN). Dahl et al. [20] combined the ResNet with a pixel recursive network, which showed promising results on face and bed images. The major problem of these networks is the difficult hyperparameter tuning in the training, such as the weight initialization, the weight decay rate, and the learning rate. With inappropriate parameters, the training might have higher risk for falling into the local minimum, especially for a large diverse training set.

Other researchers investigating improving the efficiency of the CNNs by learning the upscaling filters. Dong et al. [29] proposed the fast version of SRCNN with less filters and smaller kernels. Shi et al. [28] designed a sub-pixel convolution layer, which consists of an array of upscaling filters. These methods start the super resolution from smaller feature maps and receptive fields. Although the efficiency is improved, the accuracy is not as good as the networks working on larger feature maps.

There are also some researchers working on using multiple models together for super resolution. Wu et al. [35] proposed the 3D convolutional fusion (3DCF) method using the exact same convolutional network architecture to address both image denoising and single image super resolution. Timofte [37] designed a locally adaptive fusion of the anchored regressors. In this paper, we focus on improving the accuracy of simple individual networks (e.g., SRCNN) by the proposed context-wise network fusion. Multiple SRCNNs are fused by additional convolution layers. The output SR network achieves the state-of-the-art performance, with a relatively simple learning procedure.

3. Fusing multiple convolution neural networks

Let $\mathbf{x}$ denote a LR image and y denote a HR image. Given N training samples $\{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$, suppose we have M individual CNNs denoted as $\{S_j, j = 1, \ldots, M\}$. Each of them gives a prediction

$$\hat{y_{ij}} = S_j(\mathbf{x}_i), i = 1, \ldots, N, j = 1, \ldots, M. \quad (1)$$

Our goal is to construct a network $S = F_{\{S_1,\ldots,S_M\}}$ based on these individual networks, which makes prediction

$$\hat{y_i} = S(\mathbf{x}_i) = F_{\{S_1,\ldots,S_M\}}(\mathbf{x}_i), \quad (2)$$

while minimizing the error function

$$L(S) = \sum_{i=1}^{n} l(y_i, \hat{y_i}). \quad (3)$$

To construct the $F_{\{S_1,\ldots,S_M\}}$, a straightforward way is using the pixel-wise operation, which is called Pixel-Wise network Fusion (PWF), as shown in Fig. 1. The fused output is the pixel-wise weighted sum of the outputs of individual networks. Such fusion scheme could be denoted as

$$\hat{y_i} = F_{\{S_1,\ldots,S_M\}}(\mathbf{x}_i) = \sum_{j=1}^{M} w_j S_j(\mathbf{x}_i) + b_j, \quad (4)$$

where w_j and b_j are constants.

The PWF might work well for some cases. Similar fusion scheme has been used for object detection [14][36] or classification [16]. In the super resolution, since the output is a feature map with relatively complicate characteristics, pixel-wise fusion will not fit well.

Another way is progressively organizing the individual networks, e.g., using the output of the previous network as the input of the next network. It is inspired by cascade refining the SR output [12]. This fusion is called Progressive Network Fusion (PNF), as defined in equation (5)

$$\hat{y_i} = F_{\{S_1,\ldots,S_M\}}(\mathbf{x}_i) = S_1(S_2(\ldots S_M(\mathbf{x}_i))). \quad (5)$$

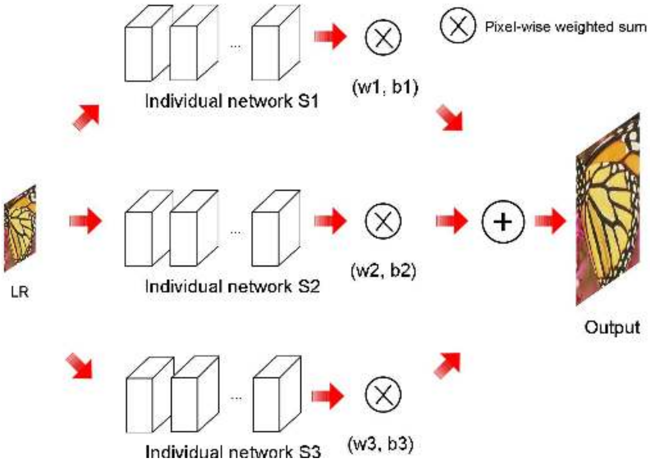

Figure 1. Pixel-Wise network Fusion (PWF) based on 3 individual networks.

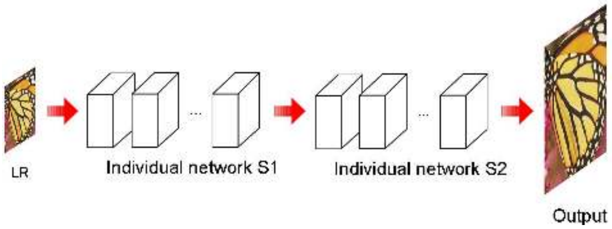

Figure 2. Progressive Network Fusion (PNF) with 2 individual networks.

For the PNF, the input of each progressive individual network will be different. So it requires generating multiple training sets. An advantage of the PNF is that it motivates us for a 'preview' stage in the super resolution system, e.g., use an efficient network to roughly SR the LR image to a coarse HR image, and then apply a complicate network to refine the coarse HR image to the final output. Such SR preview will be very useful in the real SR application.

We propose the Context-wise Network Fusion (CNF) as follows

$$\hat{y}_i = F_{\{S_1,\ldots,S_M\}}(\mathbf{x}_i) = \sum_{j=1}^{M} W_j * S_j(\mathbf{x}_i) + b_j. \qquad (6)$$

In equation (6), $\{W_j, b_j\}$ are the fusion layers constructed by convolution kernel(s). The weights of the fusion layers could be learned by fine-tuning the whole network. In the fine-tuning, the weights of the individual networks could be either frozen or not. Fig. 3 gives an example of two CNFs constructed by 3 individual networks. The CNF in Fig. 3(a) only learns the weights of the fusion layers, while freezing the weights of the individual networks. So the output could be considered as the fusion of three intermediate HR images from individual networks. The S_j in equation (6) will not be modified in the CNF training. The CNF in Fig. 3(b) fine-tunes both the fusion layers and the individual networks. This will result in different S_j after the CNF training.

In the experiments, we show that both of these two CNFs improve the accuracy compared to individual networks. Training without freezing the weights of the individual networks leads to larger gain of the accuracy.

4. Implementation

4.1. Learning individual network

SRCNN [13][15] is a representative baseline method for deep learning-based SR approach. SRCNN consists of three layers: patch extraction/representation, non-linear mapping, and reconstruction. It could be trained on large diverse training set (e.g., ImageNet) with random weight initialization and fixed learning rate. In this paper, we utilize SRCNN as the individual network.

In [15], Dong et al. tested deeper model but did not find superior performance. They concluded that deeper networks did not always result in better performance. Kim [17] succeeded in training a 20-layer CNN with specific weight initialization, higher learning rate and gradient clipping. But careful parameter tuning is required. We argue that with the same simple parameters as conventional 3-layer SRCNN, deeper network still gives better performance.

We first train a 3-layer SRCNN as the baseline following the SRCNN 9-5-5 structure in [15]. $\{64, 32, 32\}$ filters of spatial sizes 9×9, 5×5, and 5×5 are used respectively for the first, second, and third layer. To get deeper SRCNN, a feasible way is to use the existing weights of current SRCNN. So we insert new layers into the SRCNN 9-5-5, e.g., adding a layer with 32 3×3 filters in the middle. This results in a 4-layer SRCNN as SRCNN 9-5-3-5. Different from [15] that randomly initializes the weights for all the layers, we inherit the weights of the existing 9×9, 5×5, and 5×5 layers, and only randomly initializes the new 3×3 filter. We find that training such a 4-layer SRCNN with the same learning rate as 3-layer SRCNN is easy. The convergence is fast, and the accuracy is significantly better compared to 3-layer SRCNN. This idea could be applied iteratively to generate deeper SRCNN, e.g., from 4-layer SRCNN 9-5-3-5 to 5-layer SRCNN 9-5-3-3-5... An illustration of the trained deeper SRCNN is in Fig. 4. The LR image is upsampled to the desired size, and then feeded into SRCNN to get the output HR image.

In the implementation, we insert two 3×3 layers each time, which results in SRCNNs from 3 layers to 15 layers. Zero padding is applied for each 3×3 layer to make the size of the output feature map consistent. To accelerate the training, the 3-layer SRCNN is trained to 50 epochs, and other SRCNNs are trained around 15 epochs after inheriting the weights. All learning rates are fixed to 0.0001 without any

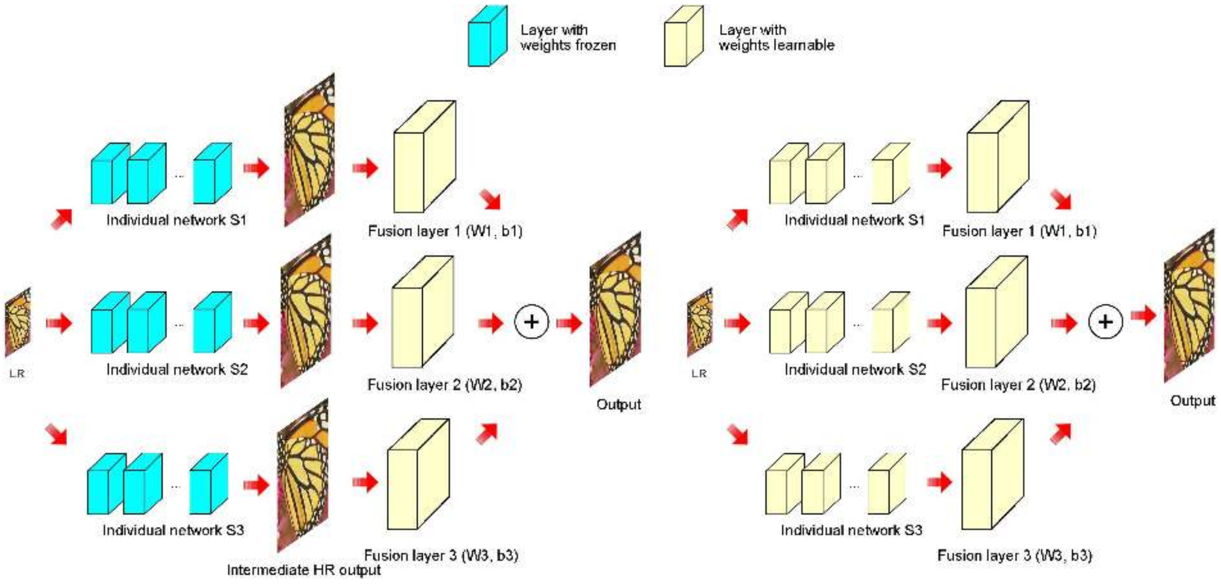

Figure 3. Context-wise Network Fusion (CNF) with 3 individual networks. (a) Learning the fusion layers while freezing the weights of the individual networks. (b) Learning the fusion layers without any frozen weights in the whole network.

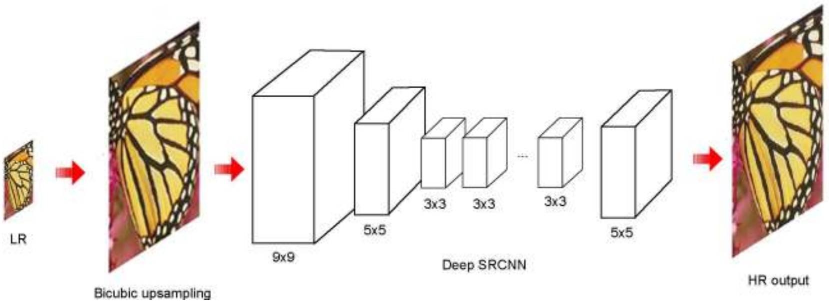

Figure 4. SRCNN with more layers.

decay. The training minimizes the MSE over the training set

$$l(y, \hat{y}) = \frac{1}{2}\sum_{i=1}^{N} ||y_i - \hat{y}_i||^2. \tag{7}$$

4.2. Learning the fused network

After learning multi-layer SRCNNs as individual networks, we train three kinds of fused networks, the PWF, the PNF, and the CNF. For CNF, three SRCNNs are utilized in each fused network. We simply set each fusion layer $\{W_j, b_j\}$ to a layer with single 3×3 convolution kernel and zero padding. The overall learnable parameters in the fusion layers will be $3 \times (3 \times 3 + 1) = 30$. In the CNF training, the weights of the individual networks might be frozen or not, as shown in Fig. 3. The weights of the fusion layers are randomly initialized by a zero mean Gaussian distribution with standard deviation 0.001.

Besides training the CNFs by bicubic downsampled LR images, we also utilize the unknown blur dataset from NTIRE 2017 challenge [38]. To get a CNF specifically for SR on unknown blurred images, we fine-tune SRCNNs trained by bicubic downsampled LR images on this dataset. Then similar way is adopted to construct the CNF based on these fine-tuned SRCNNs.

For PWF, three SRCNNs are used in each fused network. Similar to CNF, the weights of the individual networks might also be frozen in the PWF training. The weights of $\{w_j, b_j\}$ are initialized as $\{1/3, 0\}$.

For PNF, we arrange two SRCNNs in a chain order. The second SRCNN is trained using the output of the first SRCNN as input. The learning rate and weight initialization are the same as the first SRCNN. Zero padding is applied to the layers of the second SRCNN to keep the size of the output feature maps. If we do not freeze the weights of the first individual network, PNF is equal to fine-tuning a deeper network with layer number the same as the sum of the individual networks. So the weights of the first SRCNN will be frozen in training PNF.

All learning rates of the fused network are fixed to 0.0001. MSE between the fused HR output and ground truth is used as the loss function. The training of the fused network will proceed 10 epochs.

5. Experimental Results

5.1. Datasets

We generate the training patches from the Open Image Dataset [40]. This dataset consists of 9 million URLs to HD images that have been annotated with labels spanning over 6,000 categories. Due to the overlarge size, we refer to the validation set which consists of images with resolution around 640 × 480. Following the setting in SRCNN, 33 × 33 patches in bicubic upsampled LR images and 17 × 17 patches in HR images are cropped. This results in a training set with around 20 million LR-HR pairs. For testing, the commonly-used benchmark image datasets, Set5 [8], Set14 [32], and Berkeley Segmentation Dataset test set (BSD100) [33] are used. The images are bicubic downsampled as the LR input. Only the luminance channel is utilized for both the training and testing. [1]

We also refer to the dataset provided by the NTIRE 2017 challenge [38][39]. This challenge provides a large DIV2K dataset with DIVerse 2K resolution images. A training set with 800 images and a validation set with 100 images are released. Different from using bicubic downsampled HR images as input, this dataset also gives LR images generated by unknown blur. We will use these unknown blurred LR images to verify the effectiveness of SR systems.

5.2. Experiments on different fusion schemes

We first evaluate the networks constructed by different fusion schemes, PWF, PNF, and CNF. Table 1 gives the experimental results of fusing 3/5/7 layers SRCNN for super resolution with scale x3. We first observe that the accuracy of SRCNN increases with using more layers. For the pixel-wise fusion PWF, there is no significant gain on the accuracy, no matter whether the weights of the individual SRCNNs are frozen or not. This indicates that the pixel-wise fusion is not appropriate for super resolution. For PNF, the accuracy is improved based on the first individual SRCNN. We realized that the accuracy of PNF 3+5 is lower compared to individual 7-layer SRCNN. A potential reason is that the learning of our 7-layer SRCNN is similar to a PNF without freezing any weights. Since more parameters could be fine-tuned in the 7-layer SRCNN compared to the PNF 3+5, the accuracy is expected to be better.

Table 1. PSNR/SSIM evaluation of different fusion schemes based on 3/5/7 layers SRCNNs for super resolution with scale x3. 'Freeze' is whether freezing the weights of the individual SRCNNs in training the fused network. For CNF, single 3 × 3 kernel is utilized for each fusion layer.

Fusion	Layer	Freeze	Set5	Set14	BSD100
SRCNN	3	-	32.96/0.9123	29.44/0.8229	28.48/0.7893
SRCNN	5	-	33.13/0.9141	29.56/0.8258	28.54/0.7912
SRCNN	7	-	33.32/0.9169	29.68/0.8284	28.62/0.7938
PWF	3+5+7	y	33.32/0.9168	29.67/0.8285	28.61/0.7935
PWF	3+5+7	n	33.31/0.9170	29.68/0.8284	28.63/0.7938
PNF	3+3	y	33.15/0.9145	29.57/0.8263	28.54/0.7915
PNF	3+5	y	33.17/0.9148	29.61/0.8267	28.57/0.7919
PNF	5+3	y	33.18/0.9149	29.61/0.8268	28.59/0.7922
PNF	5+5	y	33.20/0.9151	29.63/0.8270	28.63/0.7924
CNF	3+5+7	y	33.38/0.9174	29.74/0.8293	28.67/0.7945
CNF	3+5+7	n	33.47/0.9184	29.79/0.8301	28.74/0.7954

It could be seen that CNF is able to consistently improve the PSNR and SSIM compared to individual SRCNNs. Learning by freezing the weights of the individual SRCNNs may improve the PSNR 0.05dB and SSIM 0.0005-0.001. If we fine-tune the whole network without any frozen weights, the gain is increased to more than 0.1dB PSNR and 0.001 SSIM. This shows the effectiveness of fusing network by convolution layers.

Next, we evaluate the CNF using different individual networks and different fusion layers. In Table 2, we notice that CNF 7+9+11 and CNF 9+11 perform better compared to individual 11-layer SRCNN. This is consistent with the results in Table 1. It could be seen that CNF 3+5+7 and 7+9+11 are still better compared to CNF 5+7 and 9+11 respectively. This indicates that using more networks might contribute to the accuracy. CNF 7+9+11 achieves better performance compared to 3+5+7, but worse compared to 11+13+15. This implies that the deeper models we use for CNF, the better accuracy we may get. We also find that CNFs with using two 3 × 3 kernels (organized in a chain order) in the fusion layers are better compared to the CNFs with using single 3 × 3 kernel in the fusion layers. This encourages us to find better ways to construct the fusion layers in the future.

Moreover, we test the CNFs with different weight initialization. In Table 3, it could be seen that both of the CNF 3+5+7 and the CNF 7+9+11 trained from unsupervised weights [17] perform much worse compared to the corresponding CNFs fine-tuned from existing models. Due to the unsupervised weights initialization, the convergence will be more difficult compared to inheriting the weights from individual networks. This shows the advantage of fusing multiple networks compared to a single end-to-end network.

As summary, CNF shows better accuracy compared to PWF and PNF. It could consistently improve the performance compared to individual networks. The accuracy of CNF could be enhanced by using deeper individual networks, more networks, or more convolution kernels in the

[1] In our case, training SR network for all three channels (yCbCr) will not improve the accuracy much compared to SR on y-channel only, plus bicubic upsampled Cb and Cr channels.

Table 2. PSNR/SSIM evaluation of CNF with different individual SRCNNs and fusion layers for super resolution with scale x3. 'Freeze' is whether freezing the weight of the individual SRCNNs when training the fused network.

Fusion	Layer	Freeze	Fusion layer for each individual network	Set5	Set14	BSD100
SRCNN	7	-	-	33.32/0.9169	29.68/0.8284	28.62/0.7938
SRCNN	11	-	-	33.59/0.9204	29.81/0.8307	28.69/0.7963
CNF	5+7	n	3×3	33.41/0.9178	29.76/0.8396	28.69/0.7947
CNF	3+5+7	n	3×3	33.47/0.9184	29.79/0.8301	28.74/0.7954
CNF	3+5+7	n	$3 \times 3 \times 2$	33.52/0.9189	29.83/0.8307	28.76/0.7958
CNF	9+11	n	3×3	33.65/0.9214	29.85/0.8314	28.76/0.7972
CNF	7+9+11	n	3×3	33.69/0.9220	29.87/0.8318	28.78/0.7975
CNF	7+9+11	n	$3 \times 3 \times 2$	33.74/0.9226	29.90/0.8322	28.82/0.7980
CNF	11+13+15	n	3×3	33.82/0.9230	29.98/0.8331	28.86/0.7986

Table 3. PSNR/SSIM evaluation of CNFs with different weights initialization for super resolution with scale x3. 'unsupervised' is unsupervised initialization, and 'fine-tune' is inheriting the weights from existing models.

Fusion	Layer	Freeze	Weight initialization	Set5	Set14	BSD100
CNF	3+5+7	n	unsupervised	33.29/0.9140	29.53/0.8254	28.52/0.7909
CNF	3+5+7	n	fine-tune	33.47/0.9184	29.79/0.8301	28.74/0.7954
CNF	7+9+11	n	unsupervised	33.45/0.9179	29.57/0.8280	28.49/0.7921
CNF	7+9+11	n	fine-tune	33.69/0.9220	29.87/0.8318	28.78/0.7975

fusion layer.

5.3. Comparison to the state-of-the-art

In Table 4, we compare the CNF networks with the state-of-the-art SR algorithms A+ [10], SRCNN [15], VDSR [17], and DRCN [18]. It could be seen that CNF 7+9+11 achieves competitive accuracy to these algorithms for all the scales. Specifically, the CNF 11+13+15 x3 scale outperforms the other methods for scale x3. We also train the 20-layer VDSR for scale x3, using the same training set as CNF. The accuracy will decrease more than 0.1dB compared to VDSR trained on 291 images. One potential reason is that it will be difficult for the hyperparamter tuning of deeper networks, especially when using a very large training set. The learning rate and weight decay rate need to be carefully designed. Fig. 4 visualizes the output HR images for different SR methods. It could be seen that CNF 11+13+15 is able to retrieve more details compared to other SR methods.

Due to using SRCNN as individual network, the network size of the CNF is not very large. The most time-consuming CNF 11+13+15 consists of 448,059 parameters, which is still smaller compared to VDSR (20 layers, 650K+ parameters) and DRCN (deep recursive network). It takes about 0.06-0.11ms per image in average on single TITAN X GPU, which is faster compared to VDSR and DRCN.

5.4. Experiments on the unknown blur dataset of NTIRE challenge

We select the 'Track 2: unknown downscaling x3 competition' in the NTIRE challenge to evaluate the proposed CNF network. We collect unknown blur LR and HR patches from the 800 training images. The 100 images in the validation set are utilized for evaluation. Following the way described in section 4, we fine-tune SRCNNs and then construct the CNF 11+13+15 for this unknown blur dataset. In Table 5, we may find that CNF 11+13+15 still outperforms 11/13/15 layer SRCNNs fine-tuned on the unknown blur dataset. This also shows the effectiveness of CNF.

6. Conclusion

In this paper, we discuss several ways to construct a super resolution system using multiple individual convolution neural networks. The pixel-wised network fusion and progressive network fusion are first introduced. A context-wise network fusion framework based on adding convolution layers after each individual network is further proposed. Experimental results on both the bicubic downsampled images and the unknown blurred images show that the proposed context-wise network fusion could improve the accuracy compared to the individual networks. Our method could be also generalized to the CNNs for other applications, such as deblur/denoising.

References

[1] Nasrollahi, Kamal and Moeslund, Thomas. Super-resolution: a comprehensive survey. *Machine vision and applications*, 25:1423–1468, 2014.

[2] Yang, Chih-Yuan and Ma, Chao and Yang, Ming-Hsuan. Single-image super-resolution: A benchmark. *ECCV*, 372–386, 2014.

[3] Glasner, Daniel and Bagon, Shai and Irani, Michal. Image Super-resolution from a single image. *CVPR*, 349–356, 2009.

[4] Kim, Kwang In and Kwon, Younghee. Single-image super-resolution using sparse regression and natural image prior. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 32:1127–1133, 2010.

[5] Allebach, Jan and Wong, Ping Wah. Edge-directed interpolation. *ICIP*, 707–710, 1996.

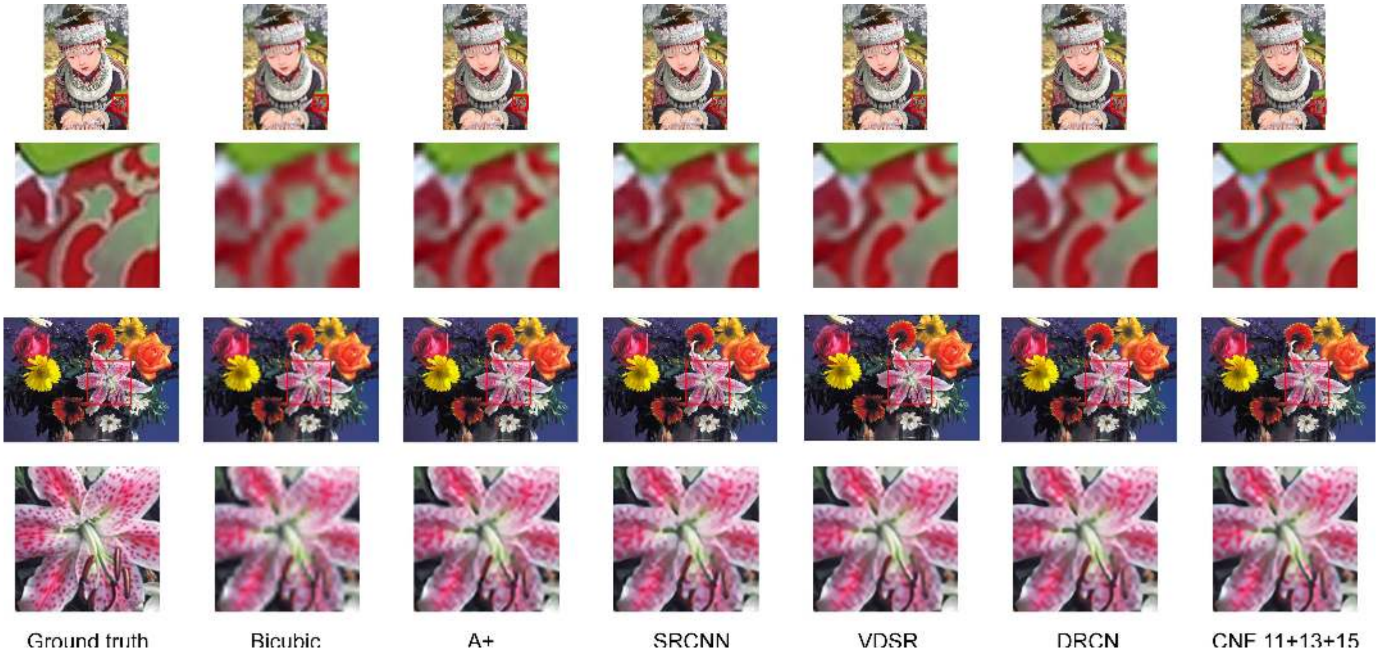

Figure 5. Visualization of the outputs HR images of different SR algorithms for scale x3.

Table 4. PSIM/SSIM/time evaluation of different SR algorithms. Red indicates the best, blue indicates the second best.

Dataset	Scale	bicubic	A+ [10]	SRCNN [15]	VDSR [17]	DRCN [18]	CNF 7+9+11	CNF 11+13+15
Set5	2	33.66/0.9299/0.00	36.54/0.9544/0.58	36.66/0.9542/0.012	37.53/0.9587/0.11	37.63/0.9588/2.05	37.66/0.9590/0.042	-
	3	30.39/0.8682/0.00	32.58/0.9088/0.32	32.75/0.9090/0.012	33.66/0.9213/0.11	33.82/0.9226/2.08	33.74/0.9226/0.041	33.82/0.9230/0.058
	4	28.42/0.8104/0.00	30.28/0.8603/0.24	30.48/0.8628/0.011	31.35/0.8838/0.11	31.53/0.8854/2.07	31.55/0.8856/0.042	-
Set14	2	30.24/0.8688/0.00	32.28/0.9056/0.86	32.42/0.9063/0.021	33.03/0.9124/0.22	33.04/0.9118/3.54	33.38/0.9136/0.072	-
	3	27.55/0.7742/0.00	29.13/0.8188/0.56	29.28/0.8209/0.021	29.77/0.8314/0.23	29.76/0.8311/3.49	29.90/0.8322/0.075	29.98/0.8331/0.109
	4	26.00/0.7027/0.00	27.32/0.7491/0.38	27.49/0.7503/0.020	28.01/0.7674/0.22	28.02/0.7670/3.51	28.15/0.7680/0.074	-
BSD100	2	29.56/0.8431/0.00	31.21/0.8863/0.59	31.36/0.8879/0.012	31.90/0.8960/0.18	31.85/0.8942/2.46	31.91/0.8962/0.049	-
	3	27.21/0.7385/0.00	28.29/0.7835/0.33	28.41/0.7863/0.012	28.82/0.7976/0.19	28.80/0.7963/2.49	28.82/0.7980/0.046	28.86/0.7986/0.088
	4	25.96/0.6675/0.00	26.82/0.7087/0.26	26.90/0.7101/0.012	27.29/0.7251/0.18	27.23/0.7233/2.53	27.32/0.7253/0.048	-

Table 5. PSNR/SSIM/time evaluation of NTIRE unknown downscaling x3.

Fusion	Layer	PSNR	SSIM	time
SRCNN	11	30.15	0.8548	0.114
SRCNN	13	30.27	0.8577	0.143
SRCNN	15	30.36	0.8600	0.168
CNF	11+13+15	30.51	0.8636	0.413

[6] Li, Xin and Orchard, Michael. New edge-directed interpolation. *IEEE Transaction on Image Processing*, 10:1521–1527, 2001.

[7] Chang, Hong and Yeung, Dit-Yan and Xiong, Yimin. Super-resolution through neighbor embedding. *CVPR*, 2004.

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

[9] Timofte, Radu and De Smet, Vincent and Van Gool, Luc. Anchored neighborhood regression for fast example-based super-resolution. *ICCV*, 2013.

[10] Timofte, Radu and De Smet, Vincent and Van Gool, Luc. Anchored neighborhood regression for fast example-based super-resolution. *ACCV*, 2014.

[11] Yang, Jianchao and Wright, John and Huang, Thomas and Ma, Yi. Image super-resolution via sparse representation. *IEEE Transaction on Image Processing*, 19:2861-2873, 2010.

[12] Timofte, Radu and Rothe, Rasmus and Van Gool, Luc. Seven ways to improve example-based single image super resolution. *CVPR*, 1865-1873, 2016.

[13] Dong, Chao and Loy, Chen Change and He, Kaiming and Tang, Xiaoou. Learning a deep convolutional network for image super-resolution. *ECCV*, 184-199, 2014.

[14] Ren, Haoyu and Li, Ze-Nian. Object Detection Using Generalization and Efficiency Balanced Co-Occurrence Features. *ICCV*, 46-54, 2015.

[15] Dong, Chao and Loy, Chen Change and He, Kaiming and Tang, Xiaoou. Image super-resolution using

deep convolutional networks. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 38:295-307, 2016.

[16] Ren, Haoyu and Li, Ze-Nian. Age estimation based on complexity-aware features. *ACCV*, 38:115-128, 2014.

[17] Kim, Jiwon and Kwon Lee, Jung and Mu Lee, Kyoung. Accurate image super-resolution using very deep convolutional networks. *CVPR*, 1646-1654, 2016.

[18] Kim, Jiwon and Kwon Lee, Jung and Mu Lee, Kyoung. Deeply-recursive convolutional network for image super-resolution. *CVPR*, 1637-1645, 2016.

[19] Ledig, Christian and Theis, Lucas and Huszár, Ferenc and Caballero, Jose and Cunningham, Andrew and Acosta, Alejandro and Aitken, Andrew and Tejani, Alykhan and Totz, Johannes and Wang, Zehan and others. Photo-realistic single image super-resolution using a generative adversarial network. *Arxiv*, 1609.04802, 2016.

[20] Dahl, Ryan and Norouzi, Mohammad and Shlens, Jonathon. Pixel Recursive Super Resolution. *Arxiv*, 1702.00783, 2017.

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

[22] Freeman, William and Jones, Thouis and Pasztor, Egon. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22:56–65, 2002.

[23] Huang, Jia-Bin and Singh, Abhishek and Ahuja, Narendra. Single image super-resolution from transformed self-exemplars. *CVPR*, 5197–5206, 2015.

[24] Tai, Yu-Wing and Liu, Shuaicheng and Brown, Michael and Lin, Stephen. Super resolution using edge prior and single image detail synthesis. *CVPR*, 2400–2407, 2010.

[25] Zhang, Kaibing and Gao, Xinbo and Tao, Dacheng and Li, Xuelong. Multi-scale dictionary for single image super-resolution. *CVPR*, 1114–1121, 2012.

[26] Schulter, Samuel and Leistner, Christian and Bischof, Horst. Fast and accurate image upscaling with super-resolution forests. *CVPR*, 3791–3799, 2015.

[27] Wang, Zhaowen and Liu, Ding and Yang, Jianchao and Han, Wei and Huang, Thomas. Deep networks for image super-resolution with sparse prior. *ICCV*, 370–378, 2015.

[28] Shi, Wenzhe and Caballero, Jose and Huszár, Ferenc and Totz, Johannes and Aitken, Andrew P and Bishop, Rob and Rueckert, Daniel and Wang, Zehan. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *CVPR*, 1874–1883, 2016.

[29] Dong, Chao and Loy, Chen Change and Tang, Xiaoou. Accelerating the super-resolution convolutional neural network. *ECCV*, 391–407, 2016.

[30] Sønderby, Casper Kaae and Caballero, Jose and Theis, Lucas and Shi, Wenzhe and Huszár, Ferenc. Amortised MAP Inference for Image Super-resolution. *ArXiv*, 1610.04490, 2016.

[31] Johnson, Justin and Alahi, Alexandre and Fei-Fei, Li. Perceptual losses for real-time style transfer and super-resolution. *ECCV*, 694–711, 2016.

[32] Zeyde, Roman and Elad, Michael and Protter, Matan. On single image scale-up using sparse-representations. *International conference on curves and surfaces*, 711–730, 2010.

[33] Martin, David and Fowlkes, Charless and Tal, Doron and Malik, Jitendra. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *ICCV*, 416–423, 2001.

[34] Yang, Chih-Yuan and Yang, Ming-Hsuan. Fast direct super-resolution by simple functions. *ICCV*, 561–568, 2013.

[35] Wu, Jiqing and Timofte, Radu and Van Gool, Luc. Generic 3D Convolutional Fusion for image restoration. *ArXiv*, preprint arXiv:1607.07561, 2016.

[36] Du, Xianzhi and El-Khamy, Mostafa and Lee, Jungwon and Davis, Larry S. Fused DNN: A deep neural network fusion approach to fast and robust pedestrian detection. *WACV*, 2017.

[37] Timofte, Radu. Anchored fusion for image restoration. *ICPR*, 2016.

[38] http://www.vision.ee.ethz.ch/ntire17/

[39] Timofte, Radu and Agustsson, Eirikur and Van Gool, Luc and Yang, Ming-Hsuan and Zhang, Lei and others. NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results/ *CVPR Workshop*, 2017.

[40] https://github.com/openimages/dataset