Supplementary Materials: Densely Residual Laplacian Super-Resolution

Saeed Anwar, *Member, IEEE,* and Nick Barnes, *Senior Member, IEEE*

1 Network Component Analysis

In this section, we provide analysis and differentiate between densely connected residual units and dense connections employed by other state-of-the-art methods. Similarly, we give an insight into the performance of channel attention and Laplacian attention. Furthermore, we compare the data used for training, time taken for inference, and the influence of the number of parameters.

1.1 Dense Connections Comparison

We provide the comparison between dense connections modules employed by state-of-the-art methods and our proposed network.

SRDenseNet: SRDenseNet [1] is inspired by dense connections of DenseNet [2] network where the layers operate on the output of all the previous layers of the block. Figure 1(a) shows the block structure of SRDenseNet [1].

RDN: Residual Dense Network [3] (RDN) employs residual connections (inspired by SRDenseNet [4]) at the local and global levels. The residual dense block (RDB) takes the input from the previous layers (or the input image) in the form of dense connection. The input to the RDB is added to the output of the RDB. Furthermore, the output of the RDB blocks is concatenated via convolution operations. The RDB is shown in the Figure 1(b).

ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) [5] removes the batch normalization layer and includes the dense connections in SRResNet [4] modules. The ESRGAN [5] modules are similar to RDN [3] modules where each layer connects in input from all the previous layers. Figure 1(b) shows the block structure for ESRGAN [5].

MemNet: MemNet [6], also known as persistent memory network, connects the blocks densely as opposed to layers, as mentioned above. The block of the MemNet is composed of the recursive unit and a gate and has no dense connection between the recursive unit but to the gate, as shown in Figure 1(c).

DRLN: Our proposed network's dense residual laplacian module (DRLM) is different from the competing dense modules. We outline the fundamental difference between our method module and other state-of-the-art, which employs dense connections.

- *Saeed Anwar is a research fellow with Data61-CSIRO.*
  E-mail: saeed.anwar@data61.csiro.au
- *Nick Barnes is team lead at CSIRO and Associate Professor at ANU.*

TABLE 1
Accuracy of super-resolution in terms of PSNR, when using Laplacian Attention and Channel Attention in a simple network with nine convolutional layers.

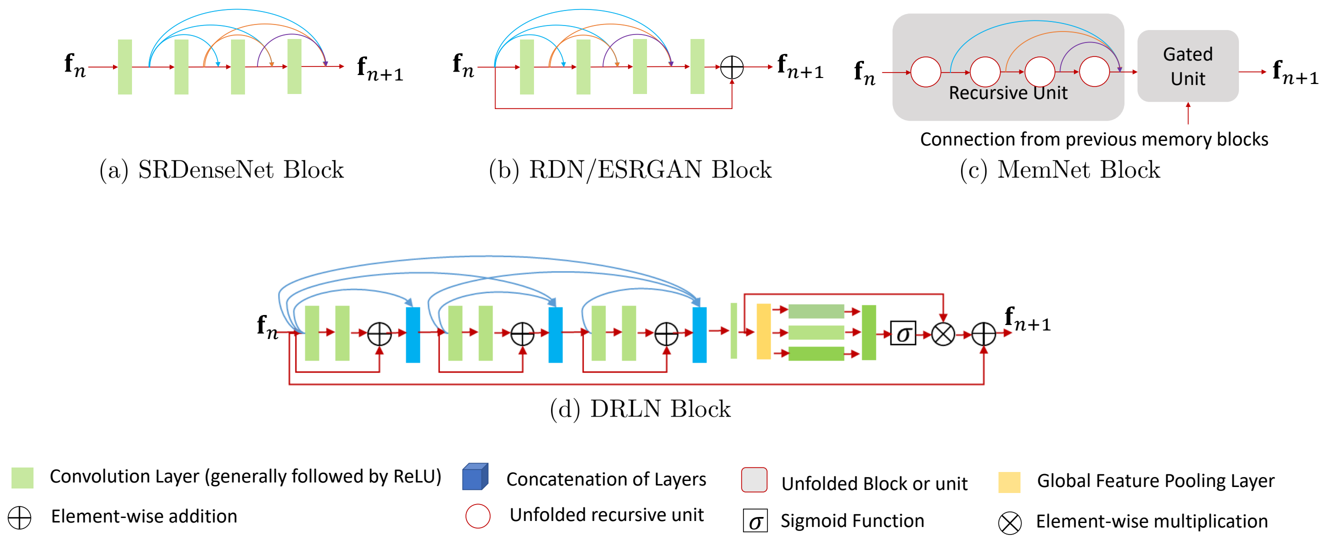
Method	Set5 [7]	Set14 [8]	BSD100 [9]	Urban100 [10]	Manga109 [11]
NetCA Channel Att.	26.34	24.33	24.69	22.83	23.02
NetLA Laplacian Att.	**26.41**	**24.40**	**24.71**	**22.99**	**23.21**

- Firstly, the whole module of RDN, SRResNet, and ESRGAN are only composed of dense connections, while our module has many units inside the DRLM module, where densely connected residual blocks unit is a part of the module. Our overall module is more complicated than a simple concatenation of the previous features.
- Secondly, DRLM employs three residual blocks in densely connected residual blocks unit while the mentioned methods have basic convolutional layers connected densely, directly inspired by DenseNet.
- Thirdly, our module concatenates the original features from each residual block while the methods mentioned above reduce the features to 64 channels after each concatenation.
- Lastly, other than dense connections, DRLM has many types of connections, which include cascading residual on residual connections, medium skip connections, residual, long skip connection, and local connections as opposed to the competing methods.

Our DRLM is more complicated and has more elements as opposed to just concatenation of previous features in the competing method modules. Our DRLM module is shown in Figure 1(d).

1.2 Laplacian Attention vs. Channel Attention

For a fair comparison with Laplacian Attention (LA) of our DRLN and Channel Attention (CA) proposed in RCAN, we attempt to use the same settings for Laplacian Attention (LA) and Channel Attention (CA). Firstly, we stack nine convolutional layers and place Channel Attention at the end before upsampling, and we call this network as NetCA. Next, we use the mentioned network but replace the attention layer with Laplacian Attention; we call this network as NetLA. We keep all the settings such as *i.e.* the training data, the number of features, the number of channels, filter size, the number of epochs *etc.*, the same. We report the



(a) SRDenseNet Block (b) RDN/ESRGAN Block (c) MemNet Block

(d) DRLN Block

Fig. 1. Type of dense connections utilized by competing methods and our DRLN.

TABLE 2
Comparison of different types of datasets employed by recent state-of-the-art for training the networks.

Method	Bicubic	SRCNN [12]	MemNet [6]	EDSR [13]	RDN [3]	RCAN [14]	CARN [15]	ESRGAN [5]	DRLN
Training Data	-	291	291	DIV2K	DIV2K	DIV2K	DIV2K	DF2K+OST	DF2K

results on the benchmark datasets for this setting in Table 1. It can be seen that with such a simple network, NetLA outperforms NetCA, which means Laplacian Attention helps to improve the performance compared to Channel Attention.

2 Training data comparison

There are many options available for training the super-resolution network. Initially, only 91 images [16] are used to train the network. However, with the advent of the convolutional neural network, more data is desired for training. For this purpose, SRCNN [12] used 200 more images with the mentioned 91 images [16] to train the network. Similarly, EDSR [13] showed performance improvement using DIV2K [17], and this trend is followed by many state-of-the-art algorithms such as CARN [15], RCAN [14]. Recently, ESRGAN [5] used two more datasets (Flicker [17], OutdoorSceneTraining (OST) [18]) to enhance the performance of their proposed model. Following the footsteps of state-of-the-art algorithms, we take a middle ground and employ the same datasets as ESRGAN (although we do not use the OST [18] dataset). A comparison of training datasets between state-of-the-art algorithm is shown in Table 2.

Further, the aim of training our network on the Flicker2K, in addition to DIV2K, is to make it robust and efficient to perform better on any future datasets.

3 Runtime Comparison

We have reported the testing time of the methods in Figure 5 of the main paper. Our method is faster as compared to RCAN [14]. Similarly, our method also trains faster than RCAN [14] as we have removed expensive addition operation and replaced it with concatenation. Keeping all the experimental settings such as batch size, learning rate *etc.* as constant, one epoch for RCAN takes around **373.6** seconds while our DRLN takes **957.5** seconds, which is much faster training time. In the following table, we present the actual training time for RCAN [14] and DRLN on four Tesla V100 GPUs, keeping all other settings the same.

TABLE 3
Runtime comparison between RCAN and DRLN for one epoch.

Batch	Computational time	
	RCAN [14]	DRLN
800/7200	137.5+2.5s	71.2+0.7s
1600/7200	104.1+0.2s	38.3+0.2s
2400/7200	104.2+0.2s	37.4+0.2s
3200/7200	102.2+0.2s	37.8+0.2s
4000/7200	101.4+0.2s	37.4+0.2s
4800/7200	101.6+0.2s	38.3+0.2s
5600/7200	101.4+0.2s	37.3+0.2s
6400/7200	102.6+0.2s	38.1+0.2s
7200/7200	102.5+0.2s	37.8+0.2s
Total time	957.5+4.1s	373.6+2.3s

Our DRLN is computationally less expensive not only in training, but it is also efficient during testing. For example, for the BSD100 [9] dataset, our method takes only **43.39s**, while RCAN requires **80.47s**.

4 Number of parameters

The number of parameters is higher in our case than RCAN [14]; however, only the number of parameters doesn't guarantee the performance as EDSR [13] have around 9 million more parameters than our DRLN, but it lacks in the performance when compared with our method. It should be noted that merely a high number of parameters do not guarantee good performance but a better design plays a vital role in achieving high accuracy (*e.g.* VGG [19] has 138M vs. ResNet50 [20] with 25.6M)

Our architecture is novel as compared to RCAN [14]. Our method performance is not merely due to the number of parameters, as pointed out that EDSR [13] has nine million more parameters than ours but has less performance than ours. Furthermore, Our method is efficient as shown earlier in Table 3 but we provide the following two strategies to decrease the number of parameters and achieve the same performance.

- Firstly, we suggest the group convolution of four, which decreases the number of parameters to six million while giving a negligible decrease in performance, keeping the length of the network the same. Similarly, if we change the network size to double with four group convolution making the parameters almost 12 million yields the equivalent performance.
- Secondly, by changing the concatenation to addition, the number of parameters becomes much lower than RCAN. Although this strategy also gives a slight decrease in performance, it also increases the computational cost. To achieve the same performance, the size of the network is increased to double (still having fewer parameters than RCAN); however, this technique requires more computations, hence takes more time during training and testing time because of additional operations instead of concatenation.

References

[1] T. Tong, G. Li, X. Liu, and Q. Gao, “Image super-resolution using dense skip connections,” in *ICCV*, 2017.

[2] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *CVPR*, 2017.

[3] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual dense network for image super-resolution,” in *CVPR*, 2018.

[4] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network.”

[5] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang, “Esrgan: Enhanced super-resolution generative adversarial networks,” *ECCV*, 2018.

[6] Y. Tai, J. Yang, X. Liu, and C. Xu, “Memnet: A persistent memory network for image restoration,” in *CVPR*, 2017.

[7] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” 2012.

[8] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in *ICCS*, 2010.

[9] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *ICCV*, 2001.

[10] J.-B. Huang, A. Singh, and N. Ahuja, “Single image super-resolution from transformed self-exemplars,” in *CVPR*, 2015.

[11] A. Fujimoto, T. Ogawa, K. Yamamoto, Y. Matsui, T. Yamasaki, and K. Aizawa, “Manga109 dataset and creation of metadata,” in *MANPU*, 2016.

[12] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *TPAMI*, 2016.

[13] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, “Enhanced deep residual networks for single image super-resolution,” in *CVPRW*, 2017.

[14] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, “Image super-resolution using very deep residual channel attention networks,” *ECCV*, 2018.

[15] N. Ahn, B. Kang, and K.-A. Sohn, “Fast, accurate, and, lightweight super-resolution with cascading residual network,” *ECCV*, 2018.

[16] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *TIP*, 2010.

[17] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, B. Lim, S. Son, H. Kim, S. Nah, K. M. Lee *et al.*, “Ntire 2017 challenge on single image super-resolution: Methods and results,” in *CVPRW*, 2017.

[18] X. Wang, K. Yu, C. Dong, and C. Change Loy, “Recovering realistic texture in image super-resolution by deep spatial feature transform,” in *CVPR*, 2018.

[19] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv*, 2014.

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.