Ablation and Integration Studies on RDN and DRLN

Ao Tang
Department of Computer Science
tangao@cs.toronto.edu

Luxi Zhao
Department of Computer Science
luxi.zhao@mail.utoronto.ca

Abstract

Single image super-resolution (SISR) is the task of recovering a high-resolution (HR) image from a low-resolution (LR) image. In this report, we perform ablation and integration studies on two existing SISR methods: Residual Dense Network [1] (RDN) and Densely Residual Laplacian Network [2] (DRLN). Our results show that DRLN performs comparably with RDN and composition of components that facilitate information and gradient flow do not necessarily lead to better performance, but may help stabilize training.

1 Introduction

Single image super-resolution (SISR) is an inverse problem aiming to generate a high-resolution (HR) image from a single low-resolution (LR) image. SISR is useful for many applications, including surveillance [3] and medical imaging [4]. A wide range of CNN-based deep networks have been proposed for SISR, including two state-of-the-art models, Residual Dense Network (RDN) [1] and Densely Residual Laplacian Network (DRLN) [2]. DRLN extends RDN with cascading blocks and Laplacian attention. This report first investigates the claim that DRLN performs better than RDN, then conducts ablation and integration studies on key components of the two networks.

2 Related Works

Initially, super-resolution (SR) methods focused on using linear network structures with no residual or dense connections, for example, SRCNN [5]. The emergence of skip connections allowed networks to grow in depth without suffering from losing information from shallower layers. VDSR [6] is one such example that leverages a deeper network along with residual learning. One limitation of listed models is that they upscale the low-resolution (LR) input before passing it into the network, which over-smooths the input. ESPCN [7] solves the problem by first extracting features in the LR space, then upsampling the final feature maps into the high-resolution output. Recently, DenseNet [8] proposes the idea of allowing direct connections between any two layers within the same dense block. RDN [1] builds on top of this by extracting additional hierarchical features to achieve a higher performance. In addition, DRLN [2] uses Laplacian attention to extract additional features to improve output quality.

3 Methodology

In this section, we provide an overview of the two chosen networks. Figure 1 and Figure 3 show the overall architectures of RDN and DRLN respectively. On a high level, the two networks both consist of three parts: initial convolutional layers for extracting shallow features from the low-resolution input, a series of residual dense blocks for further hierarchical feature extraction, and an upsampling component for upscaling coarse features to high resolution. In the following paragraphs, we describe the major architectural components introduced by the two papers. For brevity, we only describe a subset of the components studied. The rest can be found in Appendix B and Figure 1-4.

Contiguous memory (CM). RDN consists of a series of D building blocks named residual dense blocks (RDB), as shown in Figure 2. These blocks support the CM mechanism, which allows the state of the preceding block to have direct access to each Conv layer of the current block. With CM, the output of c-th Conv layer of the d-th block can be expressed as

$$F_{d,c} = \text{ReLU}(w_{d,c} * [F_{d-1}, F_{d,1}, \cdots, F_{d,c-1}] + b_{d,c}), \tag{1}$$

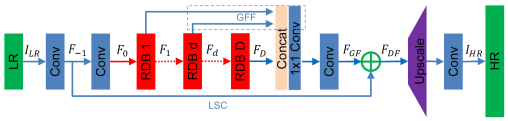


Figure 1: RDN [1] overall network architecture.

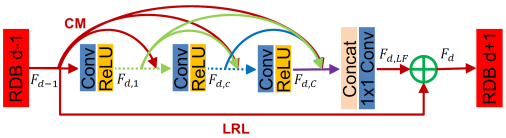


Figure 2: RDB as the building block of RDN.

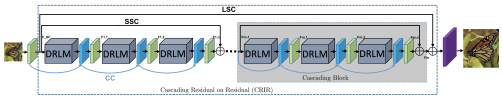


Figure 3: DRLN [2] overall network architecture.

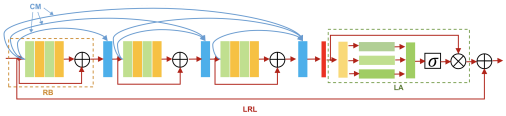


Figure 4: DRLM as the building block of DRLN.

where F_{d-1} is the output of the $(d-1)$-th block, $w_{d,c}$ and $b_{d,c}$ are the weights and biases of the c-th Conv layer. $[F_{d-1}, F_{d,1}, \cdots, F_{d,c-1}]$ is the concatenation of feature maps produced by the preceding Conv layers in the same block. In contrast, a network without CM, such as SRDenseNet [9], would exclude the term F_{d-1} in the concatenation.

Global feature fusion (GFF) is used in RDN to perform concatenation and convolution of feature maps produced by all RDBs:

$$F_{GF} = w_2 * (w_1 * [F_1, \cdots, F_d] + b_1) + b_2 \tag{2}$$

where $[F_1, \cdots, F_d]$ refers to the concatenation of all D building block outputs. w_1 and b_1 are the parameters of a 1×1 Conv layer. w_2 and b_2 are the parameters of a 3×3 Conv layer.

Cascading connections (CC). DRLN consists of a series of D building blocks named dense residual Laplacian modules (DLRM), as shown in Figure 4. These blocks are grouped together into larger cascaded blocks. Each cascaded block includes cascading connections indicated by the blue arrows in Figure 3. The implementation is a recursive structure expressed as

$$F_{m,n} = \text{ReLU}(w * [c_n, \text{DRLM}_n(x_n)] + b) \tag{3}$$

where $F_{m,n}$ is the input to the $(n+1)$-th DRLM of the m-th cascaded block. w and b are the weights and biases of a 3×3 Conv layer. If $n = 1$, then both x and c are equal to the output of the preceding cascaded block F_{m-1}. Otherwise, $x = F_{m,n-1}$ and $c_n = [c_{n-1}, \text{DRLM}_{n-1}(x_{n-1})]$

Residual blocks (RB) are one of the main components of DRLM. Each residual block consists of two Conv layer-ReLU combinations, followed by an addition of the input, as:

$$y = \text{ReLU}(w_2 * \text{ReLU}(w_1 * x + b_1) + b_2) + x \tag{4}$$

where y is the output of the residual unit.

Laplacian attention (LA) is the other main component of DRLM. It utilizes the Squeeze-and-Excitation block from SENet [10], which produces a set of weights to selectively emphasize different feature map channels. For a more detailed description, please refer to Appendix B.4.

After analyzing both architectures, we identify four components that are present in DRLN but absent in RDN: cascading connections, short skip connections, residual blocks, and Laplacian attention. There is only one component, global feature fusion, that is present in RDN but not DRLN (See Table 3). The following sections investigate the effects of these components on both networks.

4 Experimental Results

Our investigation is guided by the following **research questions**:

1. Does DRLN perform better than RDN?
2. How does each major architectural component impact the performance of RDN and DRLN?
3. Can the two models improve their performance by borrowing components from each other?

To answer these research questions, we reproduce ablation studies from the two papers (Table 1 of both RDN and DRLN). We then fuse key components of DRLN with RDN and vice versa, to observe whether the proposed benefits of one architecture generalize to the other one. To stay consistent with the two papers, super-resolution results are evaluated using peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). Detailed experiment settings are provided in Appendix C.

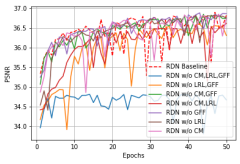


Figure 5: RDN Ablation Results

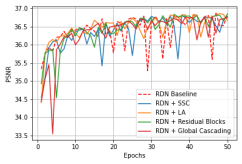


Figure 6: RDN Integration Results

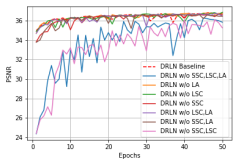


Figure 7: DRLN Ablation Results

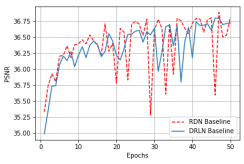


Figure 8: RDN v.s. DRLN Baseline

RDN: Table 1 and Figure 9 show the contribution of contiguous memory (CA), local residual learning (LRL), and global feature fusion (GFF) to RDN. The baseline model (no ablation) has the highest PSNR/SSIM performance across all four test datasets. The all-ablated model has the lowest performance and all other ablated models perform similarly as the baseline model. These observations are consistent with the convergence curves in Figure 5. When combined with components from DRLN, there is no significant increase in test performance, as demonstrated by the right-most columns of Table 1. However, all hybrid models except for the one with short skip connections show a more stable training process than the baseline, as demonstrated in Figure 6.

DRLN: Table 2 and Figure 10 show the contribution of Laplacian attention (LA), long skip connection (LSC), and short skip connection (SSC) to DRLN. Contrary to our expectations, the model shows a slight increase in performance with LSC removed for most datasets. The model without all three components performs the worst, as expected. Figure 7 shows the validation curves of the ablated models compared with the baseline. While other ablated models perform relatively the same as the baseline, the models without all three components and without SSC, LSC perform significantly worse than others. The right-most column in Table 2 shows the performance of adding the global feature fusion component from RDN to DRLN. The model performs better than the baseline only on the BSD100 dataset, while has a slightly lower performance on the other three datasets.

5 Discussion

Comparison with results from papers. For RDN, our ablation study results demonstrate an overall trend consistent with Table 1 of [1], with the baseline model having the highest performance, models with one component removed ranking the second, followed by those with two components removed and all-ablated. However, our PSNR values on Set 5 are consistently lower than the papers' results by an average of around 0.34 dB. This may be attributed to our smaller number of epochs (50 epochs) as compared to the paper (200 epochs). With a longer training time, we may be able to better reproduce the paper's results. For DRLN, our ablation study results are drastically different from Table 1 of [2], which reports PSNR values ranging from 31.92 to 32.27, a 5 dB drop from its same-setting counterpart in Table 2 [2]. Because of DRLN paper's inconsistency, we disregard the paper's ablation results and only focus on analyzing our results for the rest of this section.

Comparison between RDN and DRLN. Figure 8, along with the first columns of Table 1 and Table 2 demonstrate that there is no significant difference in performance of RDN and DRLN. The authors of DRLN claim that their model outperforms RDN, whereas we believe this is due to the fact that the RDN model they used has less parameters, therefore having less capacity. On the training side, DRLN is more stable than RDN, as there are significant drops in RDN's training process, while the curve of DRLN appears more stable.

Contribution of studied components. Our ablation studies show that performance only drops when multiple components are removed. Removing or adding a single component does not result in significant change in performance. One reason for this is that the architectures studied contain multiple components serving similar functions. Analysis on each component is outlined below:

Contiguous memory (CM): as introduced in Section 3, CM allows the state of the previous RDB to have direct access to each layer of the current RDB, facilitating information flow. Without CM, the

Table 1: RDN: PSNR and SSIM Results of Ablation and Integration Studies

RDN	**Ablation Studies**								**Integration Studies**			
CM	✓		✓			✓	✓					
LRL	✓			✓		✓		✓	Cascading Connections	Short Skip Connections	Residual Blocks	Laplacian Attention
GFF	✓				✓		✓	✓				
Set5	**37.66/0.877**	34.83/0.838	37.50/0.875	37.66/0.875	37.50/0.874	37.63/0.877	37.64/0.877	37.65/0.877	37.55/0.877	37.55/0.875	**37.69/0.878**	37.58/0.876
Set14	**33.23/0.802**	31.27/0.750	33.07/0.797	33.25/0.784	33.09/0.779	33.19/0.800	33.21/0.799	33.23/0.799	33.16/0.799	33.21/0.793	33.23/0.802	33.25/0.785
Urban100	**30.72/0.844**	27.51/0.768	30.20/0.833	30.59/0.832	30.39/0.833	30.65/0.844	30.64/0.840	30.69/0.843	30.60/0.843	30.50/0.838	**30.74/0.844**	30.73/0.841
BSD100	**31.98/0.800**	30.39/0.761	31.83/0.796	31.93/0.790	31.89/0.791	31.95/0.799	31.95/0.799	31.97/0.800	31.95/0.799	31.91/0.796	**31.98/0.800**	31.96/0.793

Table 2: DRLN: PSNR and SSIM Results of Ablation and Integration Studies

DRLN	**Ablation Studies**								**Integration**
SSC	✓		✓			✓	✓		Global
LSC	✓			✓		✓		✓	Feature
LA	✓				✓		✓	✓	Fusion
Set5	37.60/0.876	37.18/0.866	37.53/0.874	37.58/0.874	37.01/0.863	37.63/0.873	**37.68/0.877**	37.61/0.876	37.59/0.876
Set14	**33.22/0.783**	32.94/0.791	33.14/0.779	33.08/0.778	32.68/0.779	33.18/0.786	33.21/0.784	33.15/0.782	33.16/0.799
Urban100	30.62/0.839	30.18/0.829	30.55/0.837	30.35/0.832	29.95/0.824	30.60/0.836	**30.66/0.840**	30.51/0.837	30.61/0.842
BSD100	31.88/0.793	31.75/0.789	31.89/0.791	31.86/0.790	31.65/0.787	31.90/0.794	**31.93/0.794**	31.91/0.793	**31.89/0.796**

state of the preceding RDB, denoted as F_{d-1}, would be first passed into a Conv layer, the output of which is then concatenated with all subsequent layers. Since the output of the first Conv layer of each RDB still preserves information about the preceding RDB, even without accessing F_{d-1}, information is still passed along and local features are still being extracted.

Local residual learning (LRL) is introduced by [1] to further improve information and gradient flow. However, both dense connections (curved arrows in Figure 2) and LRL shuttle output feature maps of the previous RDB to the deeper layers of the current RDB, therefore information and gradient can still pass along with either of them being present.

Global feature fusion (GFF) is introduced to combine hierarchical features from all RDBs for global feature learning. This functionality partially overlaps with the concatenation and Conv layer in each RDB block, which also serve to preserve hierarchical features from earlier layers. Moreover, the residual connection in Figure 1 allows low-level features to be propagated to the last layer. For DRLN, GFF overlaps with SSC, LSC, and DRLM dense connections in terms of hierarchical feature preservation. These may explain the insignificant effect of removing GFF for RDN and adding GFF for DRLN.

Short/long skip connections (SSC/LSC), cascading connections (CC), and residual blocks (RB): both SSC and LSC are used in DRLN to pass features from shallower layers to deeper layers and provide an alternative path for the gradients. Therefore, as long as either component is present, the overall gradient flow is preserved. This is supported by Figure 7 where PSNR drops significantly with much more fluctuations when both components are removed, suggesting that skip connections may help stabilizing the training process. A similar stabilization effect can be observed with integrating CC and RB with RDN, as shown in Figure 6. SSC, in comparison, is not as effective as CC and RB, suggesting that shorter connections may be more suitable for facilitating gradient flow than longer connections.

Laplacian attention (LA) is proposed to re-weight each input feature map channel according to the importance of their corresponding sub-frequency-band. However, our analysis concludes that rather than extracting features at different frequencies, LA is actually simply a SEBlock [10] with three compression pathways. As shown in Appendix B.4, the input shape to each of the three "pyramid" layers is always 1×1. Convolutions are performed with increasing padding and dilation sizes. This suggests that at each convolution, only the center pixel potentially takes on a value, whereas all surrounding pixels are zero. This procedure resembles nothing like a Laplacian pyramid, where feature maps are blurred, subtracted from the original, and downsampled. Our results show that neither removing LA from DLRN nor adding LA to RDN has significant effects on test set performance, further disproving the authors' claims.

6 Summary

In this report, we analyze two recently-proposed single image super-resolution methods: RDN [1] and DRLN [2]. We investigate the effects of key components on each architecture by reproducing ablation studies and exchanging components between the two networks. Our report makes three important observations. First, we demonstrate that DRLN performs comparably with RDN. Second, our ablation and integration studies show that composition of components with similar functionalities, such as gradient flow facilitation, do not necessarily lead to better performance, but may help with the training process. Third, we debunk the DRLN paper's claims on Laplacian attention, which does not actually utilize Laplacian pyramids.

References

[1] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution, 2018.

[2] Saeed Anwar and Nick Barnes. Densely residual laplacian super-resolution, 2019.

[3] W. W. Z. Wilman and P. C. Yuen. Very low resolution face recognition problem. In *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–6, 2010.

[4] Wenzhe Shi, Jose Caballero, Christian Ledig, Xiahai Zhuang, Wenjia Bai, Kanwal Bhatia, Antonio M. Simoes Monteiro de Marvao, Tim Dawes, Declan O'Regan, and Daniel Rueckert. Cardiac image super-resolution with global correspondence using multi-atlas patchmatch. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, 2013.

[5] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *CoRR*, abs/1501.00092, 2015.

[6] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. *CoRR*, abs/1511.04587, 2015.

[7] 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, 2016.

[8] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[9] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[10] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

[11] Radu Timofte, Eirikur Agustsson, and Gool. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPRW 2017*. IEEE Computer Society, 2017.

[12] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie line Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference*, pages 135.1–135.10. BMVA Press, 2012.

[13] Roman Zeyde, Michael Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, 2010.

[14] 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 *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423 vol.2, 2001.

[15] J. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, 2015.

Attribution

All group members contributed equally.

Ao Tang: running DRLN baseline, ablation, and integration experiments.

Luxi Zhao: running RDN baseline, ablation, and integration experiments.

Appendix

A Code

The code implementation of RDN and DRLN along with the scripts to reproduce all the ablation and integration experiments can be found in this GitHub repository[1].

B Architectural Component Descriptions

Table 3: Architecture Comparision between RDN and DRLN

	Feature Extraction	CM	LRL	GFF	CC	SSC	LSC	RB	LA
RDN	Two Convs	✓	✓	✓	×	×	✓	×	×
DRLN	One Conv	✓	✓	×	✓	✓	✓	✓	✓

B.1 Local Residual Learning (LRL)

LRL is a component of RDN that adds the output of the preceding block to the output of the current block for forming the final block output: $F_d = F_{d,LF} + F_{d-1}$, where $F_{d,LF}$ is generated by passing the output of the last Conv layer of block d through a concatenation and fusion layer.

B.2 Short skip connections (SSC)

SSC is a component of DRLN that adds the output of the preceding cascaded block to the output of the current cascaded block: $F_m = F_{m,n} + F_{m-1}$, where F_m is the output of the m-th cascaded block.

B.3 Long skip connections (LSC)

LSC a component of DRLN that adds the output of the shallow feature extraction layer F_{SF} to the output of the final cascaded block F_M: $F_{LSC} = F_{SF} + F_M$.

B.4 Laplacian Attention (LA)

To produce the weights s for the attention mechanism, we perform the following steps. For each channel of an input of shape $H \times W \times C$, we reduce the $H \times W$ feature map to a single scalar descriptor to capture global information of the input:

$$g_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} f_c(i, j) \tag{5}$$

where g_c is the $1 \times 1 \times 1$ global descriptor for channel c. Aggregating all C descriptors into a vector g of shape $1 \times 1 \times C$, we then pass g into three Conv layers, followed by another convolution of the concatenated feature maps:

$$r_3 = \text{ReLU}(\text{conv}_3(g)) \tag{6}$$
$$r_5 = \text{ReLU}(\text{conv}_5(g)) \tag{7}$$
$$r_7 = \text{ReLU}(\text{conv}_7(g)) \tag{8}$$
$$s = = \sigma(\text{conv}_u([r_3, r_5, r_7])) \tag{9}$$

[1] `https://github.com/Luxi-Zhao/SingleImageSuperResolution`

Table 4: Parameters Configurations of RDN and DRLN

		D	C	G	G0	Epochs	Batch Size	Scale	# Parameters
RDN	Original Paper	16	8	64	64	200	16	x2x3x4	22123395
	Our Baseline	20	9	64	64	50	16	x2	34316675
DRLN	Original Paper	20	9	64	64	Unknown	16	x2x3x4x8	34430131
	Our Baseline	20	9	64	64	50	16	x2	34430131

where $conv_3$, $conv_5$, $conv_7$, and $conv_u$ are defined as the following:

Algorithm 1: LA Conv Layer Instantiations

```
reduction = 16
conv3 = Conv(in_channels=C, out_channels=C/reduction, kernel_size=3, padding=3, dilation=3)
conv5 = Conv(in_channels=C, out_channels=C/reduction, kernel_size=3, padding=5, dilation=5)
conv7 = Conv(in_channels=C, out_channels=C/reduction, kernel_size=3, padding=7, dilation=7)
conv_u = Conv(in_channels=(C/reduction)*3, out_channels=C, kernel_size=3, padding=1,
  dilation=1)
```

The output of the Laplacian Attention module is therefore the input feature map f_c re-weighted by weights s:

$$\hat{f}_c = s \times f_c \tag{10}$$

where f_c is the c-th channel of the input feature maps and $\hat{f}_c$ is the corresponding channel of the output feature maps.

C Experiment Settings

C.1 Benchmarks & Metrics

We train the two networks on the same DIV2K dataset. This is a high-quality (2K resolution) dataset that is widely used for image restoration applications [11]. It includes 800 training images, 100 validation images, and 100 test images. We choose to use Set5 [12], Set14 [13], B100 [14], Urban100 [15] as our benchmarks. These datasets consist a total of 219 images, and we believe it covers sufficient cases to validate the generality of the models.

In terms of metrics, We use PSNR and SSIM for evaluating the performance the models. PSNR measures the ratio between the maximum possible power of a signal and the power of corrupting noise that affects fidelity of its representation. SSIM is a perception-based model that measures the similarity between the two images. These two metrics are treated as the standard metrics for measuring the performance of the image super-resolution task, therefore we use them to measure the performance of the two chosen models.

C.2 Configuration Parameters

As shown in Table 4, the number of parameters for the original paper implementations of RDN and DRLN differ significantly. Therefore, for a fair comparison, we increase the number of parameters for RDN while keeping the parameters of DRLN the same as the paper's so that the two models have relatively the same amount of parameters. D is the number of building blocks, RDB for RDN and DRLM for DRLN. C is the number of Conv layers per block. G is the growth rate, which is the number of feature maps for each Conv layer inside a building block. G0 is the number of feature maps for Conv layers outside of the building blocks. In our experiments, we ran fewer epochs (50) than the original papers (200) due to the tight time constraint. We believe running for 50 epochs is sufficient for the purpose of our comparison studies because, from Figure 5 of the RDN paper [1], we observe that the training curves already differentiated at 50 epochs. In addition, we have observed that the model improves insignificantly after 50 epochs.

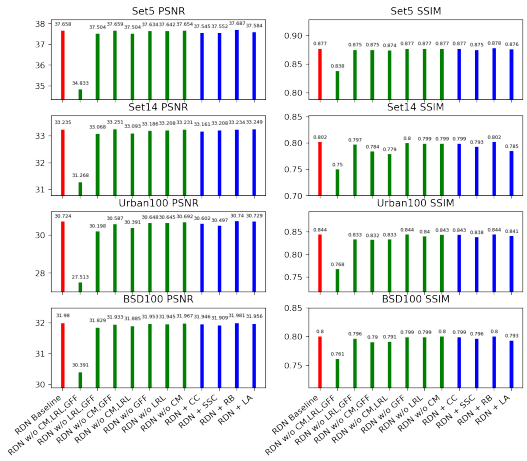


Figure 9: RDN Ablation Chart Bar

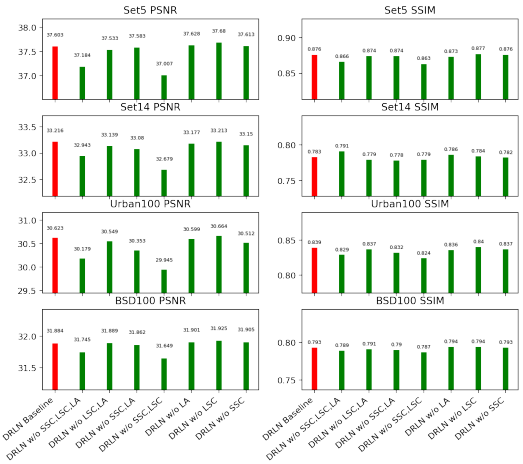
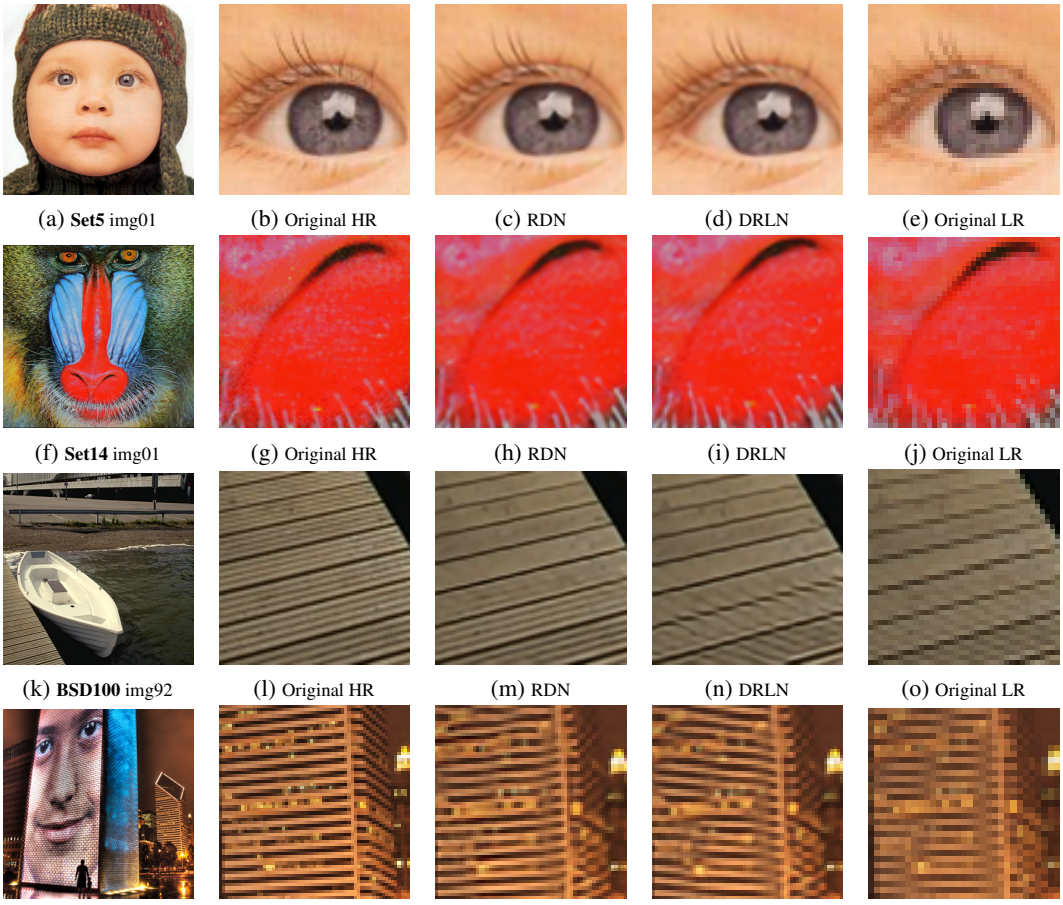


Figure 10: DRLN Ablation Chart Bar



(a) **Set5** img01 (b) Original HR (c) RDN (d) DRLN (e) Original LR

(f) **Set14** img01 (g) Original HR (h) RDN (i) DRLN (j) Original LR

(k) **BSD100** img92 (l) Original HR (m) RDN (n) DRLN (o) Original LR

(p) **Urban100** img76 (q) Original HR (r) RDN (s) DRLN (t) Original LR

Figure 11: Visual Comparison

D Result Visualizations

We visualize the ablation and integration experiment results of RDN in Figure 9 and DRLN in Figure 10. The red bars represent the baseline model, green bars represent the ablated models, and blue bar represent the hybrid models. Removing all three key architectural components (The second column) leads to significant PSNR drop. However, as long as the model has at least one of the three components, the performance can be very close to the baseline model. The improvement of combining RDN and DRLN together is not very significant according to these plots. We also provide the qualitative results of RDN and DRLN models compared against the ground-truth images on four test datasets in Figure 11.

E Future Directions

For Laplacian attention, our ablation study removed two Conv layers from DRLN's implementation, only leaving the first 3×3 Conv layer with padding and dilation. There are GitHub issues [2] claiming that not only do we not need three Conv layers, but we also don't need dilation and padding. Laplacian attention essentially boils down to RCAN's visual attention mechanism [3]. We could try to prove this as well.

For experiment settings, due to time constraints, we only ran each experiment once with 50 epochs. Should there be sufficient time and computing resources, we would like to verify our results by repeating for more trials and running for more epochs.

[2] `https://github.com/saeed-anwar/DRLN/issues/15`

[3] `https://github.com/yulunzhang/RCAN/blob/master/RCAN_TrainCode/code/model/rcan.py`