Paper

- **Title**: Residual Dense Network for Image Super-Resolution
- **Authors**: Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, Yun Fu
- **Link**: https://arxiv.org/abs/1802.08797
- **Tags**: Residual Dense Network, Image Super-Resolution, hierarchial features, residual dense block, contiguous memory
- **Year**: 2018
- **Code**: https://github.com/yulunzhang/RDN

Summary

What

The authors propose to use Residual Dense Network (RDN) based on the Residual Dense Blocks (RDB) to solve the task of Image Super-Resolution - generate high-resolution image from its degraded low-resolution (LR). Comparing the current work with the related works authors note that the methods from the related works neglect to fully use information of each convolutional layer and neglect to use hierarchial features for reconstruction.

Experiments on benchmark datasets with different degradation models show that RDN achieves favorable performance against state-of-the-art methods.

How

RDN with proposed RDB makes use of all the hierarchial features from the original LR image. RDB consists dense connected layers and local feature fusion with local residual learning (Figure 1). The output of c-th COnv layer of d-th RDB can be formulated as $F_{d,c} = \sigma(W_{d,c}[F_{d-1}, F_{d,1}, ..., F_{d,c-1}])$, where σ denots the ReLU activation function and $W_{d,c}$ is the weights of the c-th Conv layer. $F_{d,c}$ consists of G feature-maps, $[F_{d-1}, F_{d,1}, ..., F_{d,c-1}]$ refers to the concatenation of the feature-maps $G_0 + (c-1)G$.

Then Local feature fusion is applied by using 1x1 convolutional layer to reduce the feature number.

Local residual learning: the final output of the d-th RDB can be obtained by $F_d = F_{d-1} + F_{d,LF}$, in the same way as in residual blocks.

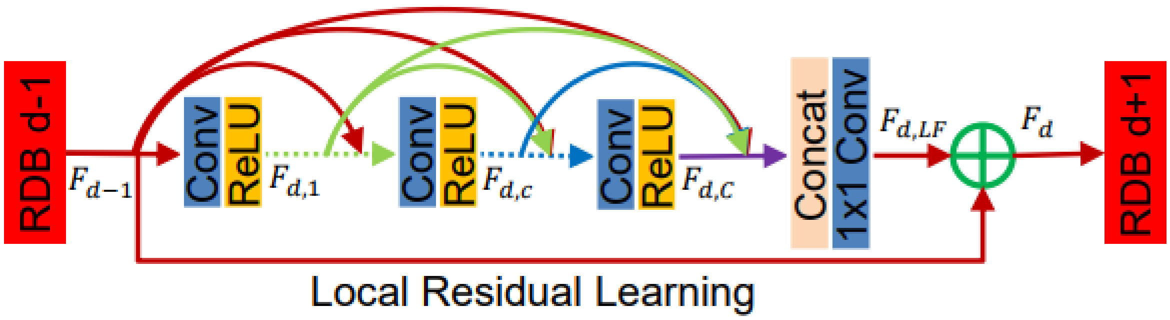

Figure 1: Residual dense block

The output of one RDB has direct access to each layer of the next RDB. Each convolutional layer in RDB has access to all the subsequent layers. Each layer has direct access to the original LR input (Figure 2).

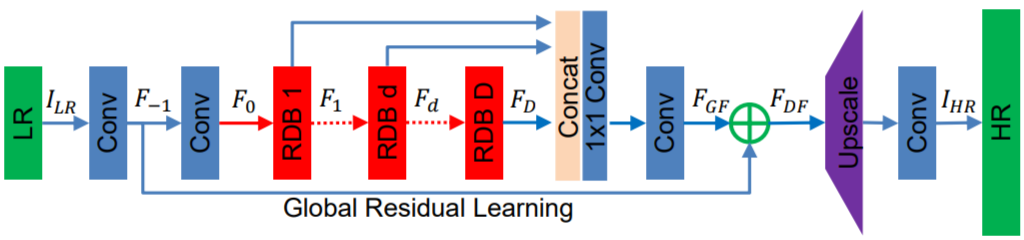

Figure 2: Residual dense network

RDN is constructed in a similar way as RDB:

- There is a Global feature fusion, that is almost similar to the Local feature fusion. The difference is that in RDB blocks are concatenated using Dense block approach, while in RDN features are simply concatenated after the last RDB block. And there is a similar 1x1 convolutional layer. The difference is that there is an additional 3x3 convolutional layer that is introduced to extract features for global residual learning
- There is a Global residual learning, that is similar to the Local residual learning.

At the end of the RDN there is an upscale part that is realized by using Efficient Sub-pixel Convolutional Neural Network (ESPCNN, implemented in the "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network" paper in CVPR, 2016) and it is followed by the conv layer with 3 output channels to output color HR images.

Results

For evaluation the dataset DIV2K was used.

The SR results are evaluated with PSNR and SSIM ("Image quality assessment: from error visibility to structural similarity") on Y channel of transformed YCbCr space.

To simulate LR images three degradation models were used:

- **BI** - bicubic downsampling by adopting the Matlab function *imresize* with the option *bicubic*. x2, x3 and x4 scaling factors are used
- **BD** - blurred downsampled image with scaling factor x3. Image is blurred by using the Gaussian kernel of size 7x7 with standard deviation 1.6
- **DN** - downsampled noised image. Image is first bicubic downsampled with factor x3 and then Gaussian noise is added with noise level 30

Dataset	Scale	Bicubic	SRCNN [3]	LapSRN [13]	DRRN [25]	SRDenseNet [31]	MemNet [26]	MDSR [17]	RDN (ours)	RDN+ (ours)
Set5	×2	33.66/0.9299	36.66/0.9542	37.52/0.9591	37.74/0.9591	-/-	37.78/0.9597	38.11/0.9602	38.24/0.9614	**38.30/0.9616**
	×3	30.39/0.8682	32.75/0.9090	33.82/0.9227	34.03/0.9244	-/-	34.09/0.9248	34.66/0.9280	34.71/0.9296	**34.78/0.9300**
	×4	28.42/0.8104	30.48/0.8628	31.54/0.8855	31.68/0.8888	32.02/0.8934	31.74/0.8893	32.50/0.8973	32.47/0.8990	**32.61/0.9003**
Set14	×2	30.24/0.8688	32.45/0.9067	33.08/0.9130	33.23/0.9136	-/-	33.28/0.9142	33.85/0.9198	34.01/0.9212	**34.10/0.9218**
	×3	27.55/0.7742	29.30/0.8215	29.79/0.8320	29.96/0.8349	-/-	30.00/0.8350	30.44/0.8452	30.57/0.8468	**30.67/0.8482**
	×4	26.00/0.7027	27.50/0.7513	28.19/0.7720	28.21/0.7721	28.50/0.7782	28.26/0.7723	28.72/0.7857	28.81/0.7871	**28.92/0.7893**
B100	×2	29.56/0.8431	31.36/0.8879	31.80/0.8950	32.05/0.8973	-/-	32.08/0.8978	32.29/0.9007	32.34/0.9017	**32.40/0.9022**
	×3	27.21/0.7385	28.41/0.7863	28.82/0.7973	28.95/0.8004	-/-	28.96/0.8001	29.25/0.8091	29.26/0.8093	**29.33/0.8105**
	×4	25.96/0.6675	26.90/0.7101	27.32/0.7280	27.38/0.7284	27.53/0.7337	27.40/0.7281	27.72/0.7418	27.72/0.7419	**27.80/0.7434**
Urban100	×2	26.88/0.8403	29.50/0.8946	30.41/0.9101	31.23/0.9188	-/-	31.31/0.9195	32.84/0.9347	32.89/0.9353	**33.09/0.9368**
	×3	24.46/0.7349	26.24/0.7989	27.07/0.8272	27.53/0.8378	-/-	27.56/0.8376	28.79/0.8655	28.80/0.8653	**29.00/0.8683**
	×4	23.14/0.6577	24.52/0.7221	25.21/0.7553	25.44/0.7638	26.05/0.7819	25.50/0.7630	26.67/0.8041	26.61/0.8028	**26.82/0.8069**
Manga109	×2	30.80/0.9339	35.60/0.9663	37.27/0.9740	37.60/0.9736	-/-	37.72/0.9740	38.96/0.9769	39.18/0.9780	**39.38/0.9784**
	×3	26.95/0.8556	30.48/0.9117	32.19/0.9334	32.42/0.9359	-/-	32.51/0.9369	34.17/0.9473	34.13/0.9484	**34.43/0.9498**
	×4	24.89/0.7866	27.58/0.8555	29.09/0.8893	29.18/0.8914	-/-	29.42/0.8942	31.11/0.9148	31.00/0.9151	**31.39/0.9184**

Figure 3: Benchmark results with BI degradation model. Average PSNR/SSIM values for scaling factor x2, x3, x4

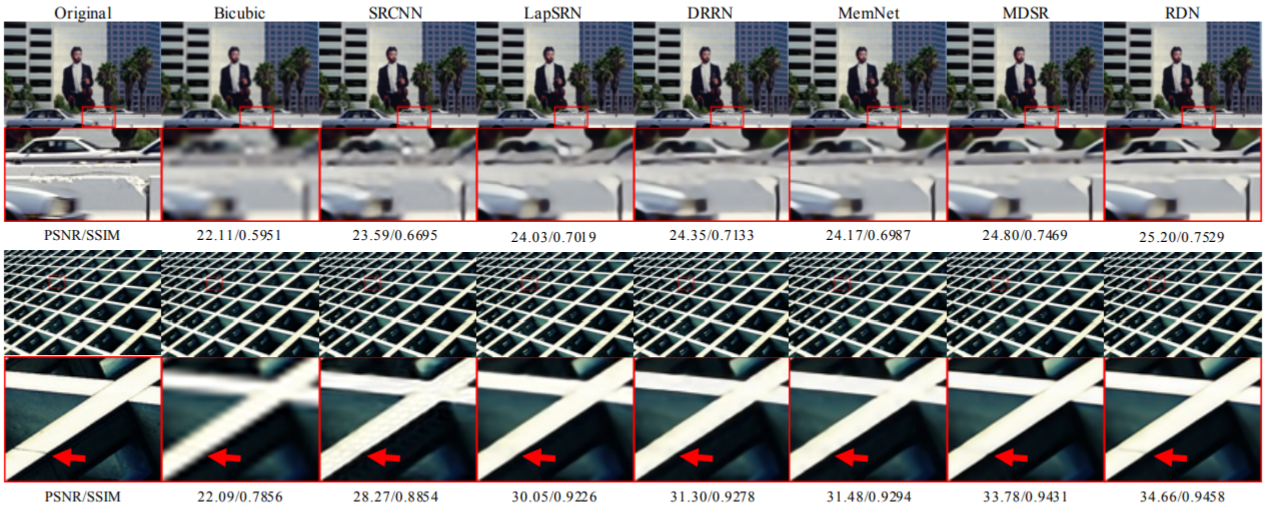

Figure 4: Visual results with BI model (x4). The SR results are for image "119082" from B100 and "img_043" from Urban100 respectively

Dataset	Model	Bicubic	SPMSR [20]	SRCNN [3]	FSRCNN [4]	VDSR [10]	IRCNN_G [38]	IRCNN_C [38]	RDN (ours)	RDN+ (ours)
Set5	**BD**	28.78/0.8308	32.21/0.9001	32.05/0.8944	26.23/0.8124	33.25/0.9150	33.38/0.9182	33.17/0.9157	34.58/0.9280	**34.70/0.9289**
	DN	24.01/0.5369	-/-	25.01/0.6950	24.18/0.6932	25.20/0.7183	25.70/0.7379	27.48/0.7925	28.47/0.8151	**28.55/0.8173**
Set14	**BD**	26.38/0.7271	28.89/0.8105	28.80/0.8074	24.44/0.7106	29.46/0.8244	29.63/0.8281	29.55/0.8271	30.53/0.8447	**30.64/0.8463**
	DN	22.87/0.4724	-/-	23.78/0.5898	23.02/0.5856	24.00/0.6112	24.45/0.6305	25.92/0.6932	26.60/0.7101	**26.67/0.7117**
B100	**BD**	26.33/0.6918	28.13/0.7740	28.13/0.7736	24.86/0.6832	28.57/0.7893	28.65/0.7922	28.49/0.7886	29.23/0.8079	**29.30/0.8093**
	DN	22.92/0.4449	-/-	23.76/0.5538	23.41/0.5556	24.00/0.5749	24.28/0.5900	25.55/0.6481	25.93/0.6573	**25.97/0.6587**
Urban100	**BD**	23.52/0.6862	25.84/0.7856	25.70/0.7770	22.04/0.6745	26.61/0.8136	26.77/0.8154	26.47/0.8081	28.46/0.8582	**28.67/0.8612**
	DN	21.63/0.4687	-/-	21.90/0.5737	21.15/0.5682	22.22/0.6096	22.90/0.6429	23.93/0.6950	24.92/0.7364	**25.05/0.7399**
Manga109	**BD**	25.46/0.8149	29.64/0.9003	29.47/0.8924	23.04/0.7927	31.06/0.9234	31.15/0.9245	31.13/0.9236	33.97/0.9465	**34.34/0.9483**
	DN	23.01/0.5381	-/-	23.75/0.7148	22.39/0.7111	24.20/0.7525	24.88/0.7765	26.07/0.8253	28.00/0.8591	**28.18/0.8621**

Figure 5: Benchmark results with BD and DN degradation models. Average PSNR/SSIM values for scaling factor x3

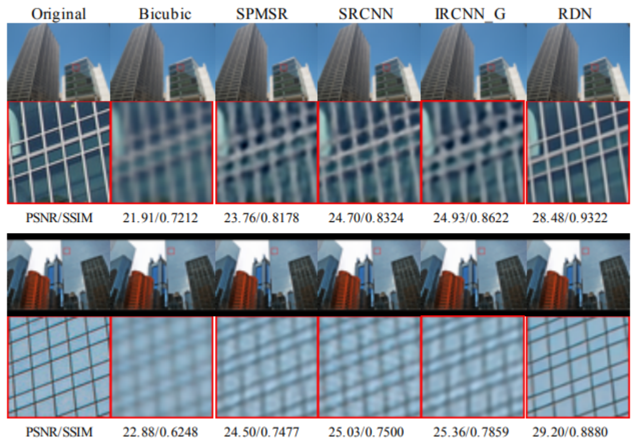

Figure 6: Visual results using BD degradation model with scaling factor x3. The SR results are for image "img_096" from Urban100 and "img_099" from Urban100 respectively

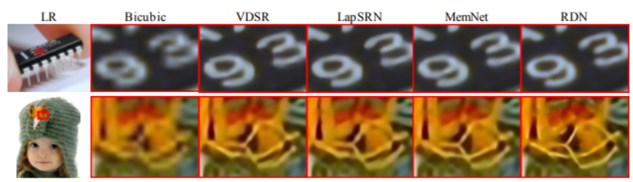

Figure 7: Visual results on real-world images with scaling factor x4. The two rows show SR results for images "chip" and "hatc" respectively

CNN Visualization

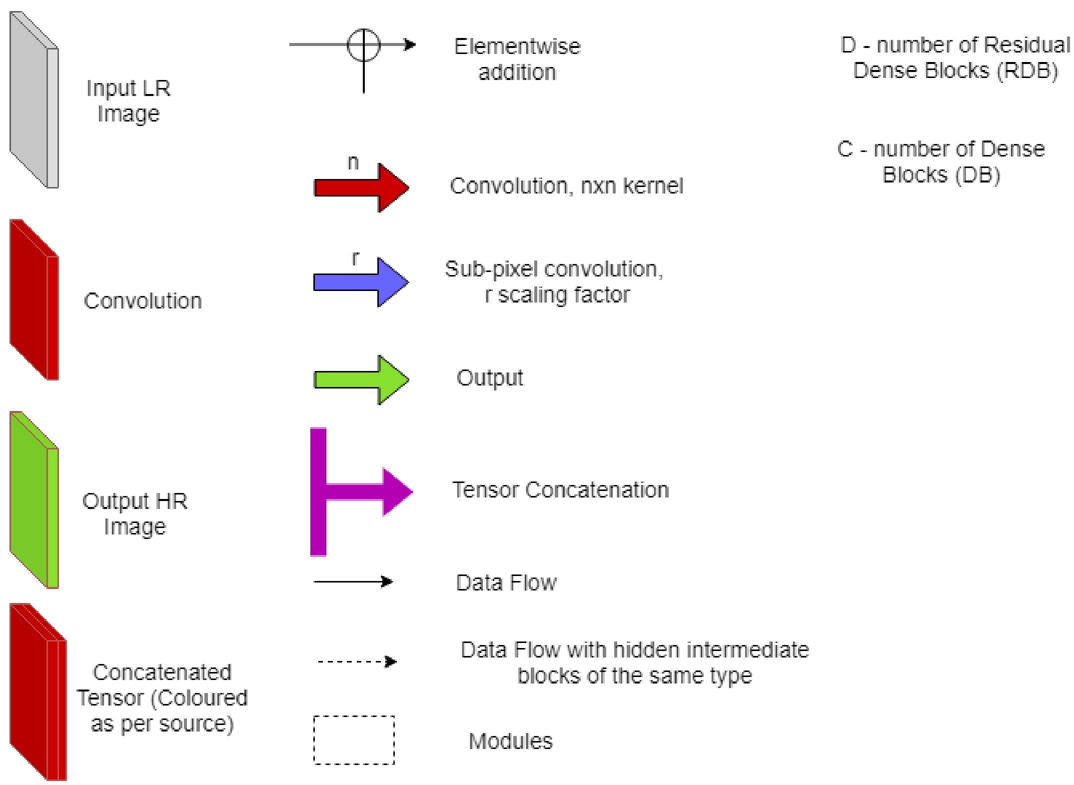

Figure 8: Legends for subsequent diagrams

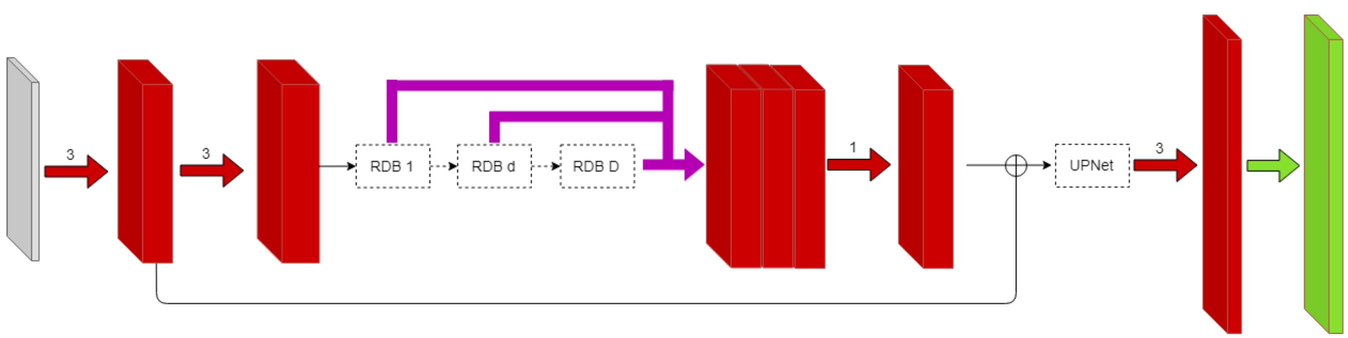

Figure 9: Residual Dense Network Architecture

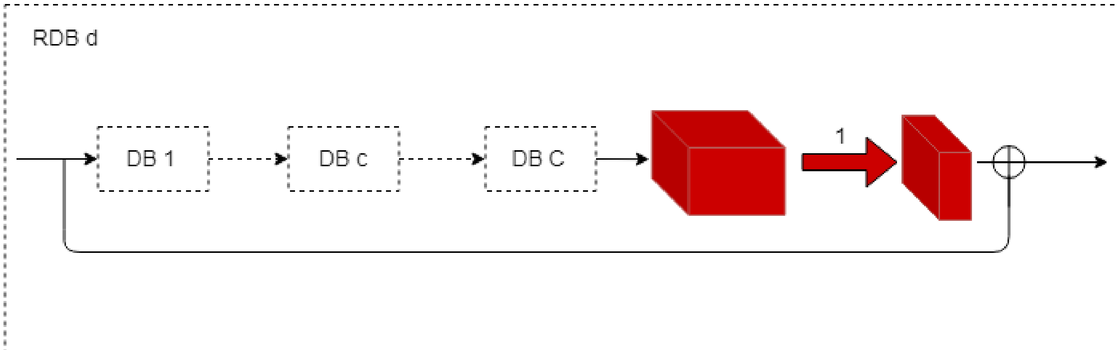

Figure 10: Residual Dense Block Architecture

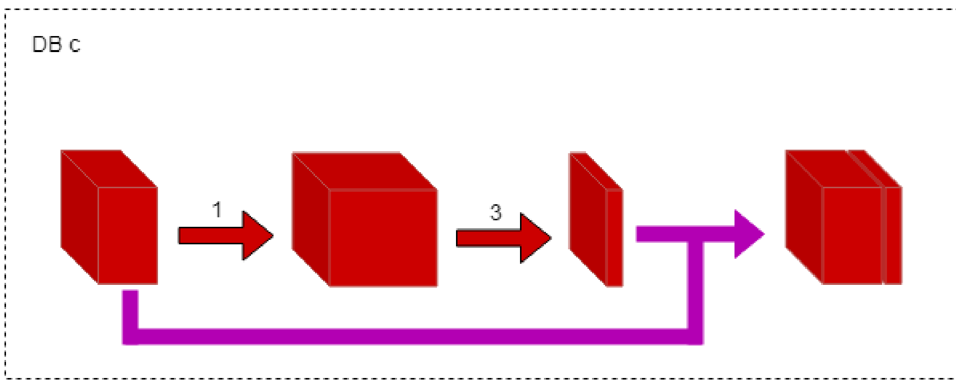

Figure 11: Dense Block Architecture

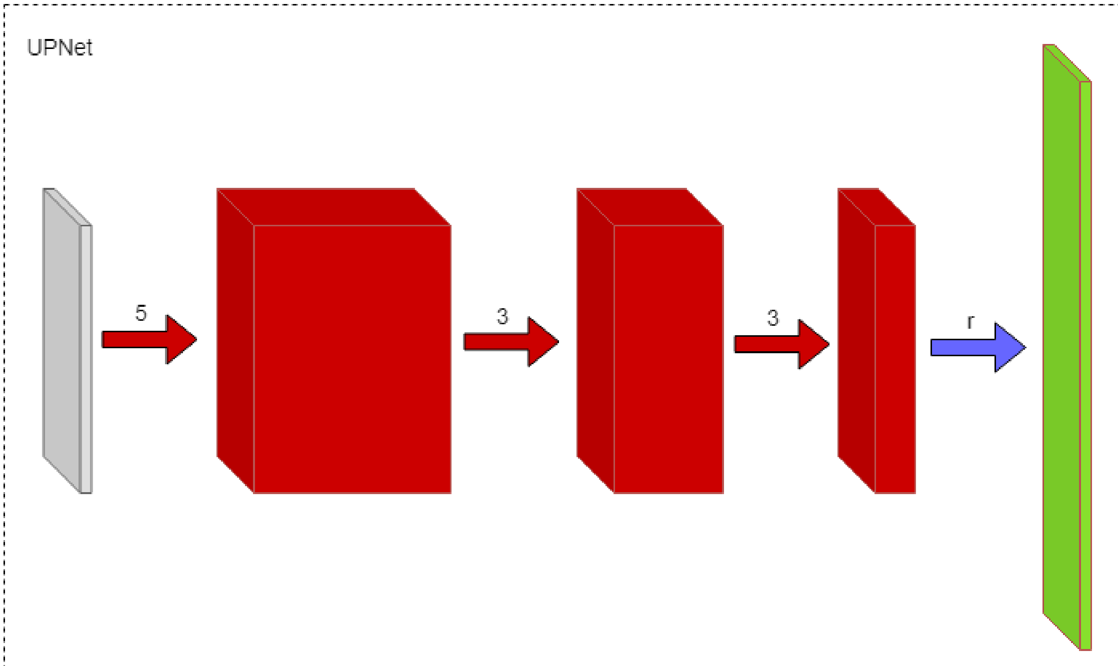

Figure 12: Efficient Sub-Pixel Convolutional Neural Network (ESPCNN), used as UPNet module