Efficient, Lightweight, Coordinate-Based Network for Image Super Resolution

Kassiani Zafeirouli
Information Technologies Institute
Centre for Research and Technology Hellas
Thessaloniki, Greece
cassie.zaf@iti.gr

Anastasios Dimou
Information Technologies Institute
Centre for Research and Technology Hellas
Thessaloniki, Greece
dimou@iti.gr

Apostolos Axenopoulos
Information Technologies Institute
Centre for Research and Technology Hellas
Thessaloniki, Greece
axenop@iti.gr

Petros Daras, Senior Member, IEEE
Information Technologies Institute
Centre for Research and Technology Hellas
Thessaloniki, Greece
daras@iti.gr

***Abstract*—Deep learning approaches have recently proven their effectiveness in the task of image Super Resolution (SR). In most cases, very deep structures have been adopted to increase the models' performance, leading to neural networks with a high parameter count that require large computational resources. In this paper, we propose an efficient, lightweight model, which leverages the benefits of recursive architectures. The structure of our network is based on progressive reconstruction, which strengthens the information flow by taking advantage of dense and residual connections. Moreover, since SR is a problem that involves spatial representations and transformations, we exploit the pixel position information to reinforce the reconstruction task. To achieve that, we use the Coordinate Convolutional layer, which exploits coordinate information allowing the network to learn the translation dependency required by the SR task. We show that the proposed method performs favorably compared to lightweight state-of-the-art methods on public benchmark datasets.**

***Index Terms*—super resolution, deep learning, coordinate convolution, recursive network**

I. Introduction

Single Image Super Resolution (SISR) aims to derive a high-resolution (HR) image from a single low-resolution (LR) version of it. SISR is an active research field, widely used in application fields that require detailed images such as surveillance, medical imaging, broadcasting. The problem is inherently ill-posed, since different HR images can produce the same LR image by down-scaling. Over the years, various computational techniques have been proposed in order to mitigate the ill-posedness nature of the problem [1], [2], [3].

Recently, deep learning methods based on convolutional neural networks (CNN) achieved significant performance improvement over the traditional techniques. Despite the fact that the first proposed networks for the SR task had few parameters [4], [5] (57K and 12K parameters respectively), most state-of-the-art (SoA) models are very deep and computationally heavy. In order to achieve higher reconstruction performance, they tend to have an ever-increasing number of stacked convolutional layers and blocks. This model structure leads to a dramatic increase in computational cost and memory consumption. For instance, the EDSR net [6] that won the first prize at NTIRE 2017 has about 43M parameters. Despite producing high quality images, very deep and heavy models are not computationally efficient and are not suitable for real time processing. For applications where limited computational resources are available or real-time performance is a prerequisite, lightweight models with a small number of parameters are required.

On the other hand, although most recent SR CNN-based methods have proposed various model structures and training approaches, the information of the pixels' position has not yet been explicitly exploited in the existing literature. The convolutional layer (Conv) learns a translation-invariant function, which means that multiple translated inputs could produce a single output. In tasks, such as image classification, where the final output is independent from the position of the input image, translation invariance is essential. However, in SR task, which involves spatial representations and transformations, this kind of translation invariance could be restrictive. The addition of the position information allows the model to adjust between various degrees of translation dependence according to the task. It is argued here that, especially for high-frequency information content (e.g edges), the coordinate information could lead to more efficient inside representations and therefore to better reconstruction.

In this work we propose a novel lightweight architecture that offers high quality HR images, while keeping the parameter count low. To achieve this, a recursive architecture is utilized, whose core element is a densely connected recursive block. Our model follows a progressive reconstruction approach, which recovers a high resolution image in intermediate steps. To ease the information flow and the training process, we introduce local skip connections at block level and long skip

This work has been funded by the EU H2020 project ASGARD Grant No 700381

978-1-7281-3401-7/18/$31.00 ©2019 IEEE

connections at pyramid level. To take a further step, our method applies coordinate convolutional layers (CoordConv) [7] to insert position information to the network by adding extra coordinate channels. This extra information leads to more efficient reconstruction of high-frequency parts of the image.

Overall, our main contributions are: 1) We propose a lightweight, recursive, densely connected coordinate-based network for high quality image restoration. Our model handles the image features more efficiently than the other SoA lightweight networks and achieves better SR performance. 2) We propose CoordConv layers to exploit pixel position information and to further improve the representation ability of our network. 3) We introduce an advanced reconstruction sub-net with more than one convolutional layer, which takes as extra input bicubic upscaled features from the previous level to be used as a guide to the restoration process.

II. Related Work

Classic vision approaches used interpolation techniques based on sampling theory, such as bicubic and Lanczos [8], to solve the SR problem. Despite their low complexity and computational cost, these methods suffer from over-smoothness and lack of high frequency information. More advanced methods introduced example-based models that learn complex mappings between low and high resolution pairs of image patches. These patches are obtained either directly from the input image, as in [9], [10], [11], [12] or from an external database. Neighbor embedding [2], [13], [14] and sparse coding [3], [15], [16], [17] are two common external-based learning SR techniques.

In 2014, for the first time, Dong et al. proposed SRCNN [4], an end-to-end 3-layer convolution network, for single image super resolution. The main idea was that the conventional sparse-coding-based SR method could be viewed as a convolution neural network. Later, based on SRCNN, Kim et al. suggested a deeper network, the VDSR [18], with more stacked convolutional layers and residual learning. DRCN [19] and DRRN [20] networks exploited recursive learning and weight sharing to achieve higher performance. In the MemNet [21] model, Tai et al introduced a memory block with skip connections in order to achieve persistent, long-term memory. The main disadvantage of the above methods is that an interpolated version of the LR image is used as input to the network instead of the original LR image. This pre-processing step leads to dramatic increase of computational cost.

To overcome this problem, several methods were applied on the original LR images by upscaling them at the end of the network using a transposed or a subpixel convolution layer [22]. This upscaling method was followed by EDSR [6], a very deep network consisting of consecutive residual blocks, with 43M parameters. Inpsired by DenseNet [23], Tong et al. introduced SRDenseNet [24], by removing the pooling layers and adding skip connections. The RDN [25] improved the SRDenseNet performance by exploiting the local and global residual learning techniques. Unlike the previous approaches, Haris et al. proposed a network with iterative upsampling and downsampling units inspired by the conventional back-projection method [26]. IDN [27], suggested by Hui et al, combines local long and short-path features to strengthen the information extraction. In [28], a lightweight network was developed that consists of local and global cascading modules with shared parameters. Recently, in RCAN [29] and RAM [30] a channel attention mechanism [31] is used to perform channel-wise feature re-calibration.

Regarding the progressive upsampling method, which is also followed by our model, Lai et al. suggested a Deep Laplacian Pyramid Network (LapSRN), based on the cascade convolutional networks [32]. Their approach reconstructs progressively residual images at multiple scales using multiple losses to guide the prediction at each level. The LapSR model was improved by adopting the design of recursive layers and multi-scale training [33]. The ProSR network [34] proposes an asymmetric pyramidal architecture without intermediate supervision. Each pyramid consists of dense blocks in order to simplify the information flow within the network.

The aforementioned methods use as reconstruction error the l2-norm or the Charbonnier penalty function. Johnson et al. [35] and Bruna et al. [36] proposed a perceptual loss based on features obtained from a pre-trained image classification network. In SRGAN [37], Ledig et al. combines a generative adversarial network with perceptual loss for more photo-realistic images. This baseline was further improved in ENEt [38] by introducing a texture matching loss. Recent studies exploit dense and residual connections to develop more efficient GAN frameworks [39].

III. Proposed Method

A. Pyramid Structure

Our model follows a progressive reconstruction approach, as shown in Fig. 1. It takes as input a LR image and progressively recovers the HR image in a coarse to fine way. Each level consists of a number of densely connected blocks followed by a sub-pixel convolutional layer. Inspired by [34], which suggests a more efficient asymmetric pyramid structure, we use more dense units at lower pyramid levels. For better supervision we follow the multi-loss logic in which each level has its own loss function. The intermediate HR images were obtained from ground truth HR image by downsampling.

B. Recursive Dense Block

The core element of our network is a dense block, as depicted in Fig. 2. Like residual connections, dense connections strengthen the information flow through the network by exploiting features from both lower and higher layers. Motivated by [23], each dense block comprises a number of bottleneck layers, which are composed of 3 consecutive operations: Conv(1,1)-ReLU-Conv(3,3). As in recent SR methods [6], [34], we remove the Batch Normalization (BN) operation. The growth rate, which determines how much new information is obtained from a layer, was set at $k = 32$ and each $(1, 1)$ Conv layer, within the bottleneck layer, produces $4 \times k$ feature maps. At the end of each dense block we use a bottleneck

978-1-7281-3401-7/18/$31.00 ©2019 IEEE

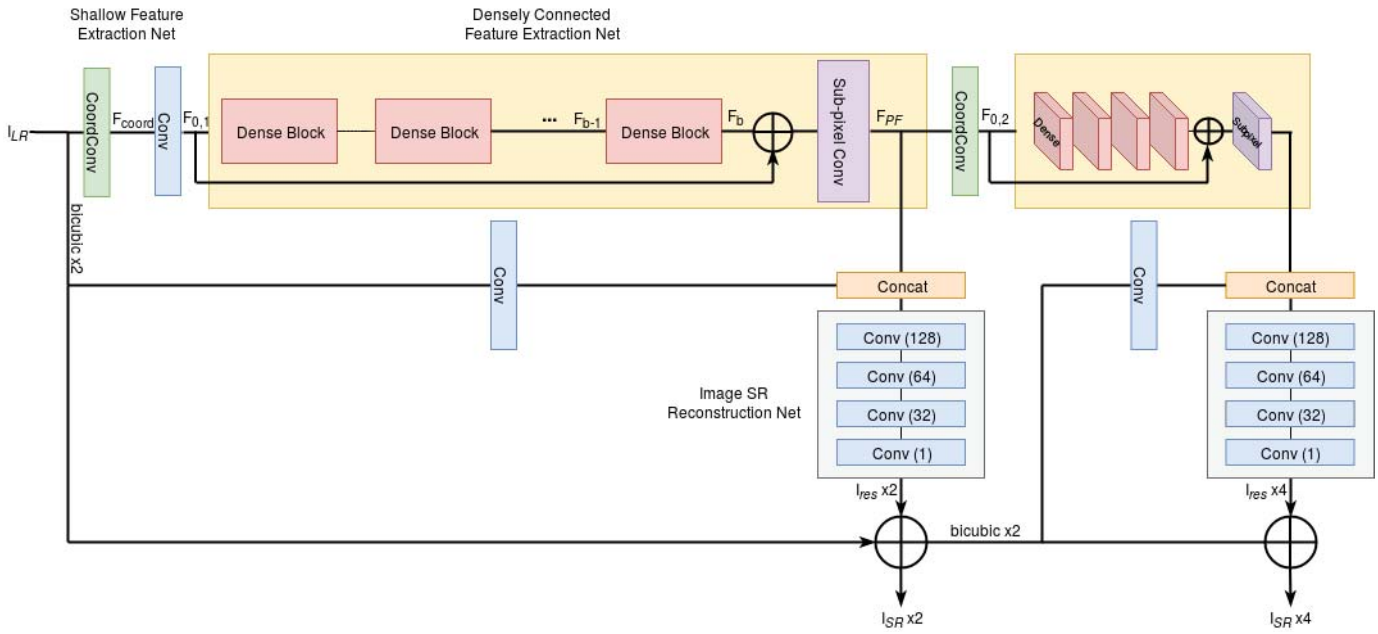

Fig. 1: Network Architecture

(1,1) layer to break the dense connection. To improve the information flow we add local skip connections, where the input to the identity branch of the residual unit is the output of the previous block at each stage.

C. Parameter Sharing

Our strategy is to share the network parameters both within and across pyramid levels. Each level consists of a number of recursive dense blocks and each dense block consists of 6 distinct bottleneck layers. The weights of the 6 bottleneck layers are shared among the recursive blocks at all levels. The reconstruction sub-net shares a common structure and task at each stage, which is to produce a 2x residual super resolution image from an incoming representation. Therefore, in order to make the model lighter, we share the reconstruction sub-net parameters across all pyramid levels.

D. Coordinate Convolutional Layer (CoordConv)

The CoordConv layer proposed by Liu et al. [7], depicted in Fig. 3, follows a different approach from standard convolution layer by explicitly exploiting spatial coordinates. Based on the idea to discard the translation invariance, a core property in CNNs, extra channels that contain coordinate information are concatenated to the input of the layer. The CoordConv layer allows the network, through the training process, to decide on the effectiveness of the translation dependence. The potential improvement due to CoordConv layers has been evaluated on various computer vision tasks including image classification, object detection and generative modeling. For processes whose performance depends on translation invariance, such as image classification, the effect of CoordConv layer is not significant. Conversely, in object detection and generative modeling, which involve a coordinate transform problem, the CoordConv addition achieved important improvement. We investigate the impact of translation dependence in the SR task by adding a CoordConv layer at the input of each pyramid level.

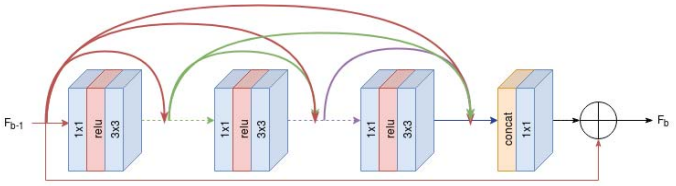

Fig. 2: Recursive Dense Block

E. Network Architecture

We denote the input, the output and the ground truth image of our network as I_{LR}, I_{SR} and I_{GT} respectively . The I_{LR} image is extracted from the I_{GT} by bicubic downscaling using the Matlab Toolbox.

1) Shallow Feature Extraction Net: Initially, one CoordConv layer and one Conv layer are used for shallow feature extraction. The CoordConv layer takes as input the I_{LR} image, adds the two extra coordinate channels and applies a Conv layer (Fig. 3).

$$F_{coord} = H_{coord} * (I_{LR}), \tag{1}$$

where H_{coord} is the coordinate convolution operation. The output channels are fed to the traditional Conv layer for further shallow processing. This layer produces 32 feature maps, which are used for global residual learning and as input to the first pyramidal level.

$$F_{0_s} = H_0 * (F_{coord}), \tag{2}$$

where H_0 denotes the convolution operation.

978-1-7281-3401-7/18/$31.00 ©2019 IEEE

2) Pyramidal Densely Connected Feature Extraction Net: Each pyramid level comprises a number of recursive dense blocks (RDBs) followed by a sub-pixel convolutional layer (SP) for 2x upscaling. The output of the sub-pixel layer at each level is connected with two different sub-nets: a) the reconstruction net that produces a residual image at current level s and b) a CoordConv layer, which is the input to the next s+1 feature extraction stage. Each pyramid level has its own global residual skip connection which combines lower and higher feature representations. At level s, the output of the b-th RDB can be obtained by

$$F_{b_s} = H_{RDB_b} * (F_{b-1_s}), \tag{3}$$

where H_{RDB_b} denotes the operation of the recursive dense block. The output of the s pyramid level can be formulated as

$$F_{PF_s} = H_{SP_s}(H_{RDB_b}(H_{RDB_{b-1}}(...(H_{RDB_1}(F_{0_s})))) + F_{0_s}), \tag{4}$$

where F_{PF_s} are the extracted feature maps from the s pyramidal level and H_{SP} is the sub-pixel operation.

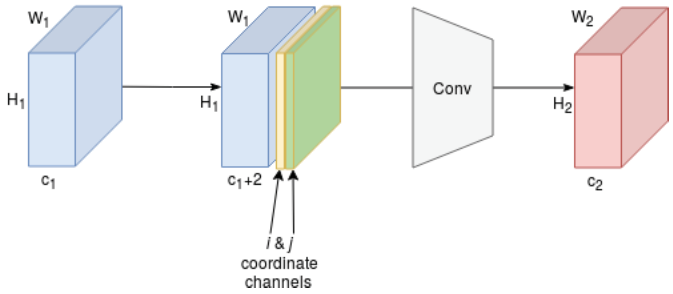

Fig. 3: Coordinate Convolutional layer

3) Image SR Reconstruction Net: Most of the SR networks commonly use as reconstruction net a single Conv layer that produces either a 3 or single channel residual image depending on the input I_{LR} image (RGB or Y-channel). In order to strengthen the reconstruction process we use more Conv layers at this sub-net.

$$H_{recon} = Conv_1(Conv_{32}(Conv_{64}(Conv_{128}))), \tag{5}$$

where H_{recon} denotes the reconstruction net and consists of consecutive convolutional operations.

Furthermore, instead of using only the feature maps from the pyramidal feature extractor as input to the reconstruction net, we propose the concatenation with features from the bicubic upscaled image of the previous level, as shown in Fig. 1. For the first level, the bicubic upscaled I_{LR} is a initial estimation of the 2x I_{SR}. Therefore the features produced by this image could be used as a guide, to this level extracted features, in order to recover a refined 2x I_{SR} image. Next levels follow the same reconstruction procedure.

The network is designed to output the residual I_{res_s}

$$I_{res_s} = H_{recon}(F_{concat}), \tag{6}$$

where F_{concat} are produced by feature concatenation as described above. The final I_{SR_s} image can be computed as

$$I_{SR_s} = I_{res_s} + \varphi(I_{SR_{s-1}}), \tag{7}$$

where φ is the bicubic interpolation operation.

F. Multi-scale Training

We follow two training strategies, the single-scale and the multi-scale one. In the single-scale one, two separate networks were built for scale factors s = 2 and s = 4 in order to be trained individually. On the other hand, in the multi-scale one, one common model was trained for the two scales {2,4}. This model predicts both 2x and 4x SR images from different layers. During training, we split alternately a batch of samples for 2x and 4x upsampling scale.

Inspired by recent models [6], [25] that use the L_1 norm as loss function, instead of the commonly used L_2 norm, we use the Charbonnier penalty function, such as in LapSRN [32]. The overall loss is formulated as:

$$L(\hat{Y}, Y) = \frac{1}{N}\sum_{i=1}^{L}\sum_{s=1}^{L}\rho(\hat{Y}_s^i - Y_s^i), \tag{8}$$

where $\rho(x) = \sqrt{x^2 + \epsilon^2}$ is the Charbonnier loss function, $\hat{Y}$, Y are the estimated SR image and the corresponding HR image respectively, N is the number of samples per batch and L is the number of pyramid levels. We choose the ϵ parameter to be 10^{-3}.

IV. Experimental Results

A. Training and Testing Datasets

All of our models were trained with the DIV2K dataset [40], which consists of 800 high quality (2K resolution) training images and 100 validation images. This dataset is commonly used in recent SR [25], [34] methods because of the diversity of images included. We use the standard benchmark datasets including Set5 [13], Set14 [15], B100 [41] and Urban100 [11] for testing. The performance of our model was evaluated with PSNR and SSIM quality metrics using only the Y channel (luminance) of the YCbCr color space. For RGB images the Cb and Cr channels are bicubically upsampled. By following [6], [26], at testing phase we crop s pixels around image boundary, where s is the scale factor.

B. Implementation and Training Details

In the proposed network, the number of filters at each (3,3) Conv layer is 32. We use 6 dense blocks at first pyramid level and 4 at second level for an asymmetric pyramidal structure. The CoordConv layer add two extra channels at input representations. The first channel contains the *i* coordinate at Cartesian space and the second channel the *j* coordinate. The (0,0) starting point corresponds to the upper left corner of the image and the coordinate values are normalized to range [0,1].

During the training phase, we randomly crop the training LR images into 48x48 patches for scale factor 2 and into 36x36 patches for scale factor 4 and set the mini-batch size at 16.

978-1-7281-3401-7/18/$31.00 ©2019 IEEE

TABLE I: Gain of multi-scale training w.r.t single-scale training on all datasets

Improvement w.r.t single-scale 2x/4x	Set5	Set14	BSD100	Urban100
single	37.74 / 31.93	33.35 / 28.39	32.06 / 27.46	31.79 / 25.88
multi-scale	+0.05 / +0.05	+0.09 / +0.02	+0.04 / +0.03	+0.14 / +0.10

These extracted patches were augmented with horizontal and vertical flipping and with 90 degree rotation. For weights initialization we use a uniform distribution. The Adam optimizer is used for model parameters update with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The initial learning rate is set to 10^{-4} and decreases by a half if there is not validation PSNR improvement for 40 epochs.

TABLE II: Investigation of CoordConv layer on all datasets (4x)

	w/o CoordConv	with CoordConv
PSNR/SSIM on Set5 (4x)	31.87 / 0.8893	31.93 / 0.8900
PSNR/SSIM on Set14 (4x)	28.37 / 0.7755	28.39 / 0.7758
PSNR/SSIM on BSDS100 (4x)	27.45 / 0.7311	27.46 / 0.7314
PSNR/SSIM on Urban100 (4x)	25.84 / 0.7767	25.88 / 0.7781

(a) Urban100(4x): img_076 (b) w/o CoordConv (c) with CoordConv

Fig. 4: Visual comparison for 4x SR, with and without CoordConv layer

C. Model Analysis

1) Coordinate Convolutional Layer: To demonstrate the effectiveness of CoordConv layer in SR task, we construct two models, with and without CoordConv layers, respectively. The two models have the same dense blocks number, bottleneck layers number per block and growth rate and are trained with scale factor 4x. The extra channels added by the CoordConv, contain the *i,j* coordinates of the input representation. By comparing the results, in Table II, we show that the model with the CoordConv layers attains higher performance than that without CoordConv, both on datasets with natural images (Set5), and on datasets with more structured images (Urban100). In Fig. 4, we show a visual comparison on scale 4x. We observe that the CoordConv model achieves to recover a finer image with more structural details. The latter confirms that the coordinate information addition allows the network to better maintain object structures, leading to improved SR images.

2) Multi-scale training: As mentioned in subsection III-F, we follow a multi-scale training strategy in order to exploit the benefits of the inter-scale correlations. To quantify the benefits of simultaneous multi-scale training over single-scale learning we train our model with the {2x}, {4x} and {2x,4x} scale combinations. As Table I shows, multi-scale training improves the reconstruction quality and outperforms the single-scale models on all datasets with major difference on datasets that contain details in various frequencies, such as the Urban100.

3) Execution Time: We evaluate the inference time on the machine with 4GHz Intel i7 CPU (32G RAM) and Nvidia GeForce GTX 1070 GPU (8G memory). Table III shows the average execution time on four benchmark datasets. The speed of our model is mainly determined by the size of the input images. The processing time increases when the resolution of the input image increases.

TABLE III: The running time (sec) on the 4 benchmark datasets with scale factor 4x

Dataset	scale	(avg) Inference time (sec)	(typ.) Image resolution
Set5	4x	0.07	120x80
Set14	4x	0.09	120x120
BSDS100	4x	0.07	120x80
Urban100	4x	0.3	256x256

D. Comparison with State-of-the-art Methods

We evaluate our proposed model by comparing with 8 state-of-the-art SR methods including SRCNN [4], VDSR [18], LapSR [32], MSLapSR [33], DRRN [20], MemNet [21], IDN [27], CARN-M [28]. The aforementioned models are lightweight with similar number of parameters with our model. We perform extensive experiments on the four benchmark datasets. Each dataset has different characteristics, namely Set5, Set14 and BDS100 contain natural images, while Urban100 contains more challenging structured images with details in various frequency bands.

In table IV, we provide a summary of quantitative comparison, in terms of PSNR and SSIM. Our final multi-scale model performs favorably against existing SoA methods both on 2x and on 4x scale factor. Especially for the challenging dataset Urban100, it outperforms significantly the existing methods. The PSNR gain of our model over the second best model is 0.62 dB and 0.36 for 2x and 4x, respectively.

Fig. 5 provides a visual comparison of the images reconstructed by the proposed model and other SoT methods.

978-1-7281-3401-7/18/$31.00 ©2019 IEEE

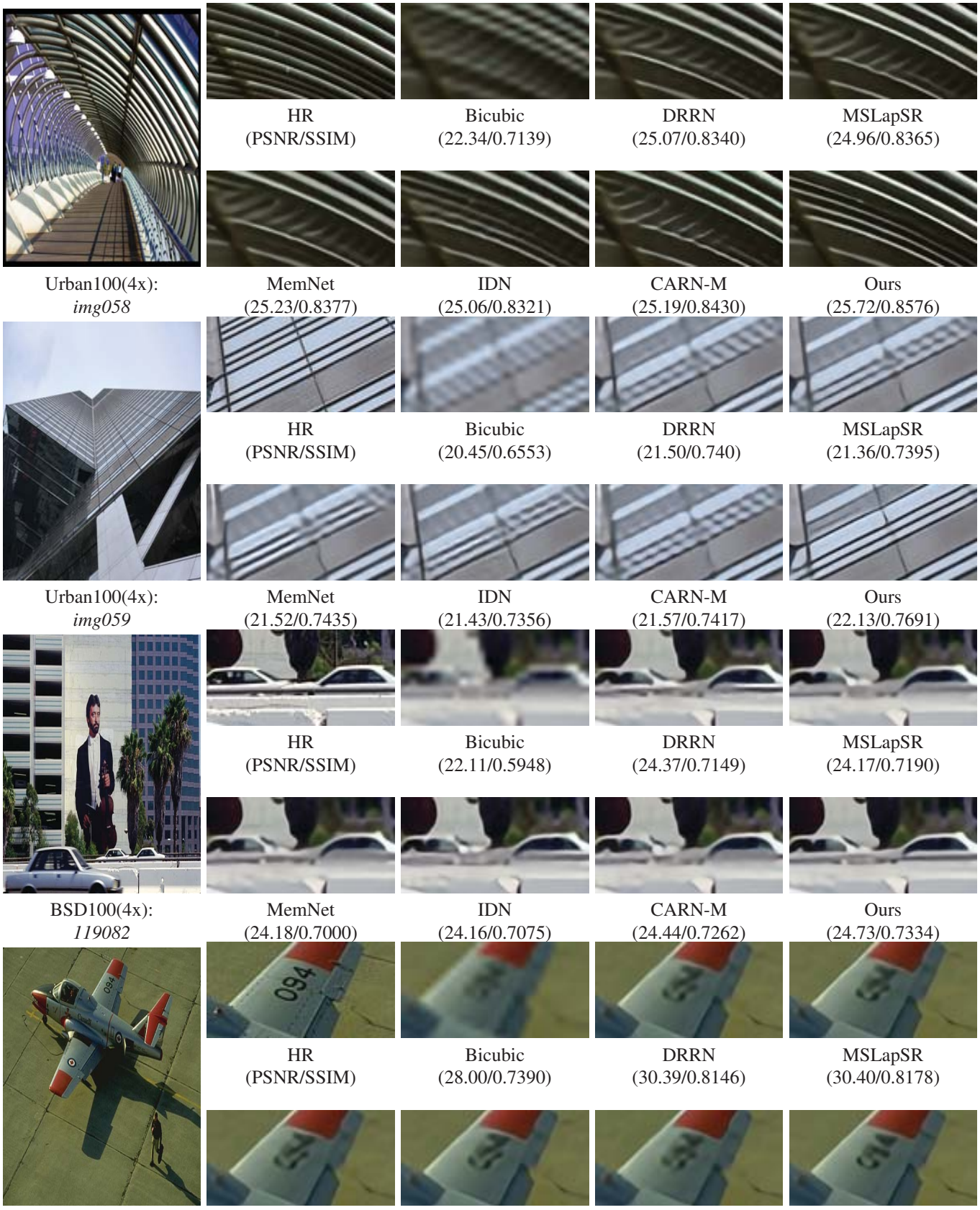

Fig. 5: Visual results of images obtained by our model and other SoT methods on datasets Urban100 and BDS100.

978-1-7281-3401-7/18/$31.00 ©2019 IEEE

TABLE IV: Quantitative evaluation of state-of-the-art SR methods. Red and blue indicate the best and the second best performance, respectively.

Scale	Method	# params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM
2x	SRCNN [4]	57K	36.66 / 0.9542	32.42 / 0.9063	31.36 / 0.8879	29.50 / 0.8946
	VDSR [18]	666K	37.53 / 0.9587	33.03 / 0.9088	31.90 / 0.8960	30.76 / 0.9140
	LapSR [32]	436K	37.52 / 0.9590	33.08 / 0.9130	31.80 / 0.8950	30.41 / 0.9100
	DRRN [20]	298K	37.74 / 0.9590	33.23 / 0.9136	32.05 / 0.8973	31.23 / 0.9188
	MSLapSR [33]	220K	37.78 / 0.9600	33.28 / 0.9150	32.05 / 0.8980	31.15 / 0.9190
	MemNet [21]	677K	37.78 / 0.9597	33.28 / 0.9142	32.08 / 0.8978	31.31 / 0.9195
	IDN [27]	553K	37.83 / 0.9600	33.30 / 0.9148	32.08 / 0.8985	31.27 / 0.9196
	CARN-M [28]	412K	37.53 / 0.9583	33.26 / 0.9141	31.92 / 0.8960	31.23 / 0.9193
	Ours	496K	37.79 / 0.9589	33.44 / 0.9146	32.10 / 0.8983	31.93 / 0.9254
4x	SRCNN [4]	57K	30.48 / 0.8628	27.49 / 0.7503	26.90 / 0.7101	24.52 / 0.7221
	VDSR [18]	666K	31.35 / 0.8838	28.01 / 0.7674	27.29 / 0.7251	25.18 / 0.7524
	LapSR [32]	813K	31.54 / 0.8850	28.19 / 0.7720	27.32 / 0.7280	25.21 / 0.7560
	DRRN [20]	298K	31.68 / 0.8888	28.21 / 0.7720	27.38 / 0.7284	25.44 / 0.7638
	MSLapSR [33]	220K	31.74 / 0.8890	28.26 / 0.7740	27.43 / 0.7310	25.51 / 0.7680
	MemNet [21]	677K	31.74 / 0.8893	28.26 / 0.7723	27.40 / 0.7281	25.50 / 0.7630
	IDN [27]	553K	31.82 / 0.8903	28.25 / 0.7730	27.41 / 0.7297	25.41 / 0.7632
	CARN-M [28]	412K	31.92 / 0.8903	28.42 / 0.7762	27.44 / 0.7304	25.62 / 0.7694
	Ours	510K	31.98 / 0.8906	28.41 / 0.7767	27.49 / 0.7323	25.98 / 0.7811

The figure shows that our model is capable to recover high frequency information from the low-resolution images. For images *image058* and *image059* of Urban100 dataset, our method accurately reconstructs the line patterns, while the other methods produce blurred images or images that contain artifacts. As shown on images *119082* and *37073* of BSD100, except for linear patterns, our model can better recover more complex objects, such as vehicles and text. In Fig. 6 we provide visual results of more 'natural' images of BSD100 dataset.

V. Conclusions

In this work, we propose a novel lightweight architecture that employs recursive dense blocks to progressively extract efficient features for HR image restoration. To enhance our model capacity, we exploit the benefits of coordinate information by adding a number of CoordCocv layers. Furthermore, we construct a more complex reconstruction sub-net, which takes into account the information of the bicubic image in order to produce a refined SR image at each level. The proposed method achieves competitive performance against other SoA approaches, on four benchmark datasets, in terms of PSNR and SSIM.

References

[1] Irani, Michal, and Shmuel Peleg. "Improving resolution by image registration." CVGIP: Graphical models and image processing 53.3 (1991): 231-239.

[2] Chang, Hong, Dit-Yan Yeung, and Yimin Xiong. "Super-resolution through neighbor embedding." Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Vol. 1. IEEE, 2004.

[3] Yang, Jianchao, et al. "Image super-resolution via sparse representation." IEEE transactions on image processing 19.11 (2010): 2861-2873.

[4] Dong, Chao, et al. "Learning a deep convolutional network for image super-resolution." European conference on computer vision. Springer, Cham, 2014.

[5] Dong, Chao, Chen Change Loy, and Xiaoou Tang. "Accelerating the super-resolution convolutional neural network." European Conference on Computer Vision. Springer, Cham, 2016.

[6] Lim, Bee, et al. "Enhanced deep residual networks for single image super-resolution." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2017.

[7] Liu, Rosanne, et al. "An intriguing failing of convolutional neural networks and the coordconv solution." Advances in Neural Information Processing Systems. 2018.

[8] Duchon, Claude E. "Lanczos filtering in one and two dimensions." Journal of applied meteorology 18.8 (1979): 1016-1022.

[9] Freedman, Gilad, and Raanan Fattal. "Image and video upscaling from local self-examples." ACM Transactions on Graphics (TOG) 30.2 (2011): 12.

[10] Glasner, Daniel, Shai Bagon, and Michal Irani. "Super-resolution from a single image." Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009.

[11] Huang, Jia-Bin, Abhishek Singh, and Narendra Ahuja. "Single image super-resolution from transformed self-exemplars." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[12] Yang, Chih-Yuan, Jia-Bin Huang, and Ming-Hsuan Yang. "Exploiting self-similarities for single frame super-resolution." Asian conference on computer vision. Springer, Berlin, Heidelberg, 2010.

[13] Bevilacqua, Marco, et al. "Low-complexity single-image super-resolution based on nonnegative neighbor embedding." (2012): 135-1.

978-1-7281-3401-7/18/$31.00 ©2019 IEEE

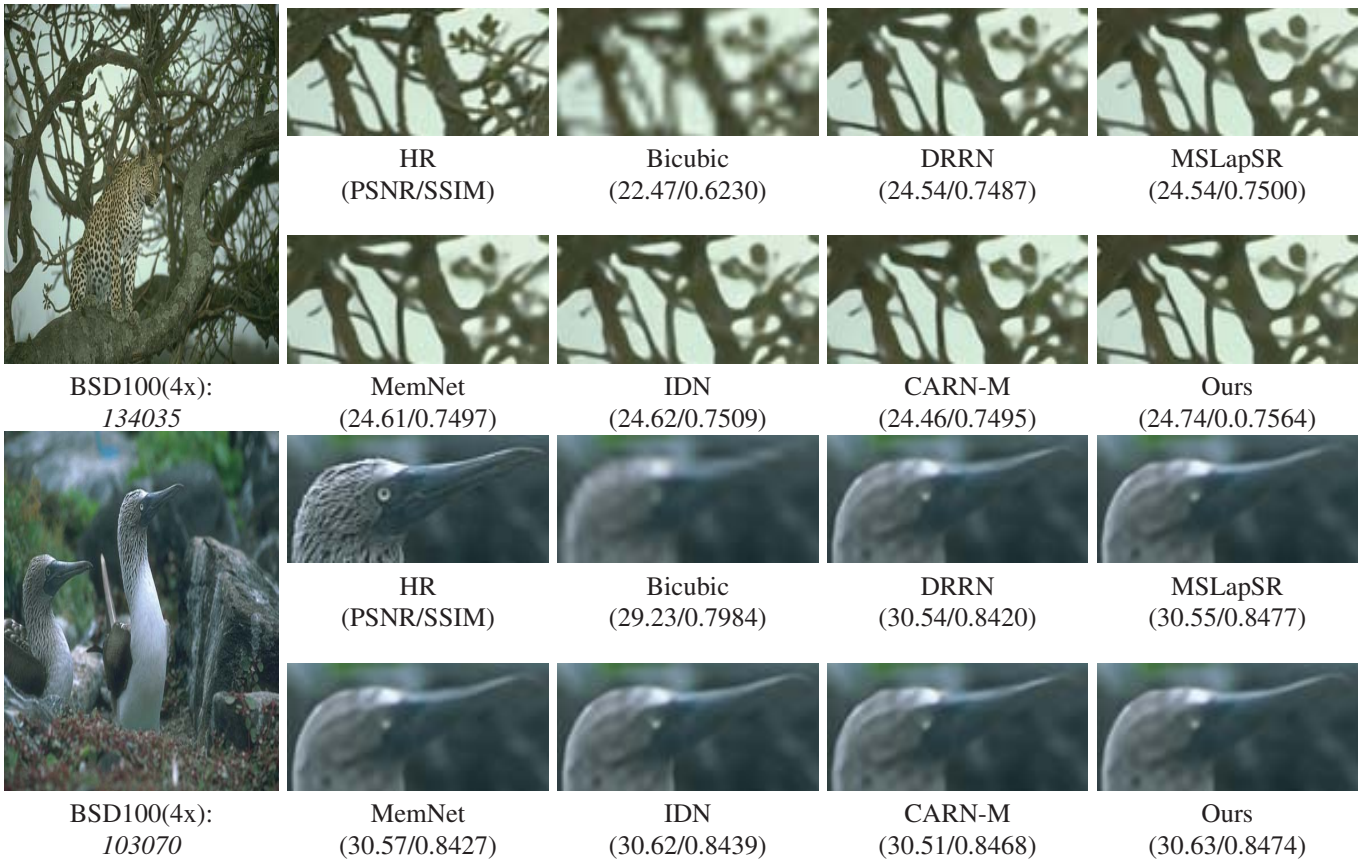

Fig. 6: Visual results of 'natural' images obtained by our model and other SoT methods on BDS100 dataset.

[14] Gao, Xinbo, et al. "Image super-resolution with sparse neighbor embedding." IEEE Transactions on Image Processing 21.7 (2012): 3194-3205.

[15] Zeyde, Roman, Michael Elad, and Matan Protter. "On single image scale-up using sparse-representations." International conference on curves and surfaces. Springer, Berlin, Heidelberg, 2010.

[16] Yang, Jianchao, et al. "Coupled dictionary training for image super-resolution." IEEE transactions on image processing 21.8 (2012): 3467-3478.

[17] Lu, Xiaoqiang, et al. "Geometry constrained sparse coding for single image super-resolution." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[18] Kim, Jiwon, Jung Kwon Lee, and Kyoung Mu Lee. "Accurate image super-resolution using very deep convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[19] Kim, Jiwon, Jung Kwon Lee, and Kyoung Mu Lee. "Deeply-recursive convolutional network for image super-resolution." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[20] Tai, Ying, Jian Yang, and Xiaoming Liu. "Image super-resolution via deep recursive residual network." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Vol. 1. No. 2. 2017.

[21] Tai, Ying, et al. "Memnet: A persistent memory network for image restoration." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[22] Shi, Wenzhe, et al. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[23] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[24] Tong, Tong, et al. "Image super-resolution using dense skip connections." Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017.

[25] Zhang, Yulun, et al. "Residual dense network for image super-resolution." The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018.

[26] Haris, Muhammad, Gregory Shakhnarovich, and Norimichi Ukita. "Deep back-projection networks for super-resolution." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[27] Hui, Zheng, Xiumei Wang, and Xinbo Gao. "Fast and Accurate Single Image Super-Resolution via Information Distillation Network." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[28] Ahn, Namhyuk, Byungkon Kang, and Kyung-Ah Sohn. "Fast, Accurate, and, Lightweight Super-Resolution with Cascading Residual Network." arXiv preprint arXiv:1803.08664 (2018).

[29] Zhang, Yulun, et al. "Image super-resolution using very deep residual channel attention networks." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[30] Kim, Jun-Hyuk, et al. "RAM: Residual Attention Module for Single Image Super-Resolution." arXiv preprint arXiv:1811.12043 (2018).

[31] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[32] Lai, Wei-Sheng, et al. "Deep laplacian pyramid networks for fast and accurate superresolution." IEEE Conference on Computer Vision and Pattern Recognition. Vol. 2. No. 3. 2017.

[33] Lai, Wei-Sheng, et al. "Fast and accurate image super-resolution with deep laplacian pyramid networks." IEEE transactions on pattern analysis and machine intelligence (2018).

[34] Wang, Yifan, et al. "A Fully Progressive Approach to Single-Image Super-Resolution." arXiv preprint arXiv:1804.02900 (2018).

[35] Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." European Conference on Computer Vision. Springer, Cham, 2016.

[36] Bruna, Joan, Pablo Sprechmann, and Yann LeCun. "Super-

resolution with deep convolutional sufficient statistics." arXiv preprint arXiv:1511.05666 (2015).
[37] Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." arXiv preprint (2017).
[38] Sajjadi, Mehdi SM, Bernhard Schlkopf, and Michael Hirsch. "Enhancenet: Single image super-resolution through automated texture synthesis." Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017.
[39] Wang, Xintao, et al. "Esrgan: Enhanced super-resolution generative adversarial networks." European Conference on Computer Vision. Springer, Cham, 2018.
[40] Timofte, Radu, et al. "Ntire 2017 challenge on single image super-resolution: Methods and results." Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on. IEEE, 2017.
[41] Martin, David, et al. "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics." Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. Vol. 2. IEEE, 2001.

978-1-7281-3401-7/18/$31.00 ©2019 IEEE