

## NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results

Radu Timofte      Shuhang Gu      Jiqing Wu      Luc Van Gool      Lei Zhang  
 Ming-Hsuan Yang      Muhammad Haris      Greg Shakhnarovich      Norimichi Ukita  
 Shijia Hu      Yijie Bei      Zheng Hui      Xiao Jiang      Yanan Gu      Jie Liu      Yifan Wang  
    Federico Perazzi      Brian McWilliams      Alexander Sorkine-Hornung  
 Olga Sorkine-Hornung      Christopher Schroers      Jiahui Yu      Yuchen Fan      Jianchao Yang  
 Ning Xu      Zhaowen Wang      Xinchao Wang      Thomas S. Huang      Xintao Wang  
 Ke Yu      Tak-Wai Hui      Chao Dong      Liang Lin      Chen Change Loy      Dongwon Park  
 Kwanyoung Kim      Se Young Chun      Kai Zhang      Pengju Liu      Wangmeng Zuo  
 Shi Guo      Jiye Liu      Jinchang Xu      Yijiao Liu      Fengye Xiong      Yuan Dong  
 Hongliang Bai      Alexandru Damian      Nikhil Ravi      Sachit Menon      Cynthia Rudin  
 Junghoon Seo      Taegyun Jeon      Jamyoun Koo      Seunghyun Jeon      Soo Ye Kim  
    Jae-Seok Choi      Sehwan Ki      Soomin Seo      Hyeonjun Sim      Saehun Kim  
 Munchurl Kim      Rong Chen      Kun Zeng      Jinkang Guo      Yanyun Qu      Cuihua Li  
 Namhyuk Ahn      Byungkon Kang      Kyung-Ah Sohn      Yuan Yuan      Jiawei Zhang  
 Jiahao Pang      Xiangyu Xu      Yan Zhao      Wei Deng      Sibt Ul Hussain      Muneeb Aadil  
 Rafia Rahim      Xiaowang Cai      Fang Huang      Yueshu Xu      Pablo Navarrete Micheli  
 Dan Zhu      Hanwen Liu      Jun-Hyuk Kim      Jong-Seok Lee      Yiwen Huang      Ming Qiu  
 Liting Jing      Jiehang Zeng      Ying Wang      Manoj Sharma      Rudrabha Mukhopadhyay  
    Avinash Upadhyay      Sriharsha Koundinya      Ankit Shukla      Santanu Chaudhury  
    Zhe Zhang      Yu Hen Hu      Lingzhi Fu

### Abstract

*This paper reviews the 2nd NTIRE challenge on single image super-resolution (restoration of rich details in a low resolution image) with focus on proposed solutions and results. The challenge had 4 tracks. Track 1 employed the standard bicubic downscaling setup, while Tracks 2, 3 and 4 had realistic unknown downgrading operators simulating camera image acquisition pipeline. The operators were learnable through provided pairs of low and high resolution train images. The tracks had 145, 114, 101, and 113 registered participants, resp., and 31 teams competed in the final testing phase. They gauge the state-of-the-art in single image super-resolution.*

### 1. Introduction

Example-based single image super-resolution (SR) targets the reconstruction of the lost high frequencies (rich

details) in an image with the help of a set of prior examples of paired low resolution (LR) and high resolution (HR) images. This problem is ill-posed, for each LR image the space of plausible corresponding HR images is huge and scales up quadratically with the magnification factor.

In the recent years the research literature largely focused on example-based single image super-resolution. The performance achieved by the top methods [38, 32, 7, 16, 20, 31, 21] continuously improved.

The NTIRE 2017 challenge [31, 1] was a step forward in benchmarking SR. It was the first challenge of its kind with tracks employing standard bicubic degradation and ‘unknown’ operators (blur and decimation) on the 1000 DIV2K resolution images from DIV2K [1] dataset.

The NTIRE 2018 challenge builds upon NTIRE 2017 and goes further. In comparison with the previous edition, NTIRE 2018: (1) uses the same DIV2K [1] dataset; (2) has only one bicubic downscaling track with magnification factor  $\times 8$ ; (3) promotes realistic settings emulating camera acquisition pipeline through three tracks with gradually increased difficulty.

R. Timofte (timofte@vision.ee.ethz.ch, ETH Zurich), S. Gu, L. Van Gool, L. Zhang and M.-H. Yang are the NTIRE 2018 organizers, while the other authors participated in the challenge.

Appendix A contains the authors’ teams and affiliations.

NTIRE webpage: <http://www.vision.ee.ethz.ch/ntire18/>

## 2. NTIRE 2018 Challenge

The objectives of the NTIRE 2018 challenge on example-based single-image super-resolution are: (i) to gauge and push the state-of-the-art in SR; (ii) to compare different solutions; and (iii) to promote realistic SR settings. **DIV2K Dataset** [1] employed by NTIRE 2017 SR challenge [31] is used also in our challenge. DIV2K has 1000 DIVERse 2K resolution RGB images with 800 for training, 100 for validation and 100 for testing purposes. The manually collected high quality images are diverse in contents.

### 2.1. Tracks

Access to data and submission of HR image results required registration on Codalab competition track.

**Track 1: Classic Bicubic  $\times 8$**  uses the bicubic downscaling (Matlab `imresize`, default settings), the most common setting from the recent SR literature, with factor  $\times 8$ . It is meant for easy deployment of recent proposed SR solutions.

**Track 2: Realistic Mild  $\times 4$  adverse conditions** assumes that the degradation operators emulating the image acquisition process from a digital camera can be estimated through training pairs of LR and HR images. The degradation operators are the same (use the same controlling parameters) within each image space and for all the images in train, validation, and test sets. As in reality, the motion blur and the Poisson noise are image dependent and can introduce pixel shifts and scaling. Each ground truth (GT) image from DIV2K is downgraded ( $\times 4$ ) to LR images.

**Track 3: Realistic Difficult  $\times 4$  adverse conditions** is similar to Track 2, only the degradation is stronger.

**Track 4: Realistic Wild  $\times 4$  adverse conditions** is similar to Tracks 2 and 3, the degradation operators are the same within an image space but *different* from one image to another. Some images are less degraded than other images. This setting is the closest to real ‘wild’ conditions. Due to increased complexity of the task 4 degraded LR images were generated for each HR train image.

**Challenge phases** (1) *Development phase*: the participants got pairs of LR and HR train images and the LR validation images of the DIV2K dataset; an online validation server with a leaderboard provided immediate feedback for the uploaded HR results to the LR validation images; (2) *Testing phase*: the participants got test LR images and were required to submit super-resolved HR image results, code, and a factsheet for their method. After the end of the challenge the final results were released to the participants.

**Evaluation protocol** The quantitative measures are Peak Signal-to-Noise Ratio (PSNR) measured in decibels [dB] and the Structural Similarity index (SSIM) [37], both full-reference measures computed between the HR result and the GT image. We report averages over sets of images.

<https://competitions.codalab.org>

As in [31] we ignore a boundary of  $6 + s$  image pixels ( $s$  is the zoom factor). Because of the pixel shifts and scalings, for Tracks 2, 3, and 4 we consider all the translations  $\in [-40, 40]$  on both axes, compute PSNR and SSIM and report the most favorable scores. Due to time complexity, for Tracks 2, 3, and 4 we computed PSNR and SSIM using a  $60 \times 60$ px centered image crop during validation phase and a  $800 \times 800$ px centered image crop for the final results.



Figure 1. Sample LR input images for Track 1,2,3, and 4, resp.

## 3. Challenge Results

From  $\sim 110$  registered participants on average per each track, 31 teams entered in the final phase and submitted results, codes/executables, and factsheets. Table 1 reports the final test results and rankings of the challenge, while in Table 2 the self-reported runtimes and major details are provided. The methods are briefly described in section 4 and the team members are listed in Appendix A.

**Architectures and main ideas** All the proposed methods, excepting TSSR of UW18, are deep learning based. The deep residual net (ResNet) architecture [10] and the dense net (DenseNet) architecture [11] are the basis for most of the proposed methods. For fast inference, thus train and test time benefits, most of the teams conduct the major SR operations in the LR space. Several teams, such as UIUC-IFP, BMITL-UNIST, Pixel\_Overflow, build their methods based on EDSR [21], the state-of-the-art approach and the winner of the previous NTIRE 2017 SR challenge [31, 1]; while, other teams, such as Toyota-TI, HIT-VPC, DRZ, PDN, proposed new architectures for SR.

**Restoration fidelity** The top 4 methods from ‘Classic Bicubic’ achieved similar PSNR scores (within 0.04dB). DeepSR entry, ranked 12th, is only 0.17dB behind the best PSNR score of Toyota-TI. On the realistic settings, Tracks 2,3, and 4, due to the existence of noise and motion blur, the training strategy and the network architecture plays are equally important. Although UIUC-IFP ranked 7th on ‘Classic Bicubic’, below DRZ and Duke Data Science, it adopted a pre-alignment step for the training phase and achieved the best performance on the realistic tracks 2 and 3, significantly better than DRZ and Duke Data Science. PDN ranked 1st on Track 4, however, without submitted results for the other tracks we cannot tell if their solution/architecture is better than that of UIUC-IFP.

**Ensembles and fusion** Most teams employ pseudo-ensembles [33]. The inputs are flipped/rotated and the HR results are aligned and averaged for enhanced prediction.

Table 1. NTIRE 2018 SR Challenge results and final rankings. Note that the ‘lpj008’ results are not ranked.

| (a) Track 1 Classic Bicubic $\times 8$ |                 |        |        | (b) Realistic Tracks 2, 3, & 4 $\times 8$ |                  |                        |        |                        |        |                        |        |
|--|-----------------|--------|--------|---|------------------|------------------------|--------|------------------------|--------|------------------------|--------|
| Team                                   | Author          | PSNR   | SSIM   | Team                                      | Author           | Track 2 Mild           |        | Track 3 Difficult      |        | Track 4 Wild           |        |
|  |                 |        |        |   |                  | PSNR                   | SSIM   | PSNR                   | SSIM   | PSNR                   | SSIM   |
| Toyota-TI                              | iim_lab         | 25.455 | 0.7088 | UIUC-IFP                                  | jhyume           | 23.631 <sub>(1)</sub>  | 0.6316 | 22.329 <sub>(1)</sub>  | 0.5721 | 23.080 <sub>(2)</sub>  | 0.6038 |
| Pixel_Overflow                         | McCourt_Hu      | 25.433 | 0.7067 | PDN                                       | xixihaha         |                        |        | 22.249                 | 0.5637 | 23.374 <sub>(1)</sub>  | 0.6122 |
| rainbow                                | zheng222        | 25.428 | 0.7055 | BMIPL_UNIST                               | BMIPL_UNIST      | 23.579 <sub>(2)</sub>  | 0.6269 | 22.074 <sub>(2)</sub>  | 0.5590 |                        |        |
| DRZ                                    | yifita          | 25.415 | 0.7068 | HIT-VPC*                                  | lpj008           |                        |        | 21.450 <sub>(9)</sub>  | 0.5339 | 22.795 <sub>(3)</sub>  | 0.5829 |
| Faceall_Xlabs                          | xjc_faceall     | 25.360 | 0.7031 | HIT-VPC                                   | cskzh            | 23.493 <sub>(3)</sub>  | 0.6174 | 21.899 <sub>(3)</sub>  | 0.5623 | 22.766 <sub>(4)</sub>  | 0.6023 |
| Duke Data Science                      | admian98        | 25.356 | 0.7037 | SIA                                       | mikigom          | 23.406 <sub>(5)</sub>  | 0.6275 | 21.689 <sub>(6)</sub>  | 0.5434 | 22.732 <sub>(6)</sub>  | 0.5844 |
| UIUC-IFP                               | jhyume          | 25.347 | 0.7023 | KAIST-VICLAB                              | jschoi           | 23.455 <sub>(4)</sub>  | 0.6175 | 21.592 <sub>(8)</sub>  | 0.5438 | 22.745 <sub>(5)</sub>  | 0.5881 |
| Haiyun_XMU                             | cr2018          | 25.338 | 0.7037 | DRZ                                       | yifita           | 23.397 <sub>(6)</sub>  | 0.6160 | 21.825 <sub>(4)</sub>  | 0.5573 | 22.707 <sub>(7)</sub>  | 0.5932 |
| BMIPL_UNIST                            | BMIPL_UNIST     | 25.331 | 0.7026 | srFans                                    | yyuan13          | 23.218 <sub>(9)</sub>  | 0.6222 | 21.658 <sub>(7)</sub>  | 0.5400 |                        |        |
| Ajou-LAMDA-Lab                         | nmhkahn         | 25.318 | 0.7023 | Duke Data Science                         | adamian98        | 23.374 <sub>(7)</sub>  | 0.6252 |                        |        |                        |        |
| SIA                                    | mikigom         | 25.290 | 0.7014 | bighead                                   |                  | 23.247 <sub>(8)</sub>  | 0.6165 |                        |        |                        |        |
| DeepSR                                 | enoch           | 25.288 | 0.7015 | ISP_Team                                  | hot_milk         | 23.098 <sub>(11)</sub> | 0.6167 | 21.779 <sub>(5)</sub>  | 0.5550 | 22.496 <sub>(8)</sub>  | 0.5867 |
|  | Mrobot0         | 25.175 | 0.6960 | BOE-SBG                                   | boe_sbg          | 23.123 <sub>(10)</sub> | 0.6008 | 21.443 <sub>(10)</sub> | 0.5275 | 22.352 <sub>(10)</sub> | 0.5612 |
| reveal.ai                              | muneebaadil     | 25.137 | 0.6942 | MCML                                      | ghgh3269         | 22.953 <sub>(12)</sub> | 0.6115 | 21.337 <sub>(11)</sub> | 0.5354 | 22.472 <sub>(9)</sub>  | 0.5842 |
| HIT-VPC                                | cskzh           | 25.088 | 0.6943 | DeepSR                                    | enoch            | 21.742 <sub>(15)</sub> | 0.5572 | 20.674 <sub>(16)</sub> | 0.5168 | 21.589 <sub>(12)</sub> | 0.5444 |
| MCML                                   | ghgh3269        | 24.875 | 0.7025 |   | jingliting       | 21.710 <sub>(16)</sub> | 0.5384 | 20.973 <sub>(12)</sub> | 0.5187 | 20.956 <sub>(14)</sub> | 0.5214 |
| BOE-SBG                                | boe_sbg         | 24.822 | 0.6817 | Haiyun_XMU                                | cr2018           | 21.519 <sub>(17)</sub> | 0.5313 | 20.866 <sub>(13)</sub> | 0.5072 | 21.367 <sub>(13)</sub> | 0.5321 |
| SRFun                                  | ccook           | 24.819 | 0.6829 | Ajou-LAMDA-Lab                            | nmhkahn          | 21.240 <sub>(18)</sub> | 0.5376 |                        |        |                        |        |
| KAIST-VICLAB                           | JSchoi          | 24.817 | 0.6810 | Juanluisgonzales                          | juanluisgonzales | 22.625 <sub>(13)</sub> | 0.5868 |                        |        |                        |        |
|  | zeweithe        | 24.773 | 0.6813 | APSARA                                    | mingqiu          |                        |        | 20.718 <sub>(15)</sub> | 0.4977 |                        |        |
|  | jingliting      | 24.714 | 0.6913 | NMH                                       | nmh              |                        |        | 20.645 <sub>(17)</sub> | 0.4890 |                        |        |
| CEERI                                  | harshakoundinya | 24.687 | 0.6719 |   | join16           | 20.453 <sub>(19)</sub> | 0.4928 |                        |        |                        |        |
| APSARA                                 | MingQiu         | 24.618 | 0.6817 | Baseline                                  | Bicubic          | 22.391 <sub>(14)</sub> | 0.5336 | 20.830 <sub>(14)</sub> | 0.4631 | 21.761 <sub>(11)</sub> | 0.4989 |
| UW18                                   | zzsmg           | 24.192 | 0.6531 |   |                  |                        |        |                        |        |                        |        |
| Baseline                               | Bicubic         | 23.703 | 0.6387 |   |                  |                        |        |                        |        |                        |        |

Table 2. Reported runtimes [s] per test image and details from the factsheets.

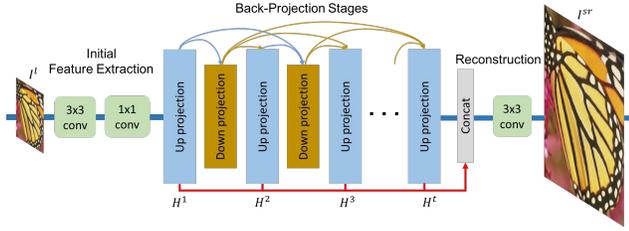
| Team              | runtime [s] |             | Platform                                     | CPU/GPU (at runtime)                   | Ensemble   |
|-------------------|-------------|-------------|--|--|--|
|                   | Track 1     | Track 2,3,4 |  |  |  |
| Ajou-LAMDA-Lab    | 13.84       | 13.84       | Pytorch                                      | GTX 1080Ti                             | flip/rotation ( $\times 8$ )                         |
| APSARA            | 30          | 30          | Tensorflow                                   | GTX 1080Ti                             | flip/rotation ( $\times 8$ )                         |
| BOE-SBG           | 0.15        | 1.11        | Pytorch                                      | Nvidia P100                            | -  |
| bighead           | -           | 1.5         | -  | -                                      | -  |
| BMIPL_UNIST       | 2.52        | 4.68        | Pytorch                                      | ?                                      | flip/rotation ( $\times 8$ )                         |
| CEERI             | 12.23       | -           | Tensorflow,Keras                             | GTX 1080                               | -  |
| DeepSR            | 9.89        | 1.83        | Tensorflow                                   | Titan X                                | flip/rotation ( $\times 8$ )                         |
| DRZ               | 11.65       | 2.91        | Pytorch                                      | Titan Xp                               | Track 1: flip/rotation ( $\times 8$ )                |
| Duke Data Science | 6.99        | 18          | ???  | Nvidia P100                            | flip/rotation ( $\times 8$ )                         |
| Faceall_Xlabs     | 7.31        | -           | Pytorch                                      | GTX 1080                               | flip/rotation ( $\times 4$ )                         |
| Haiyun_XMU        | 14.52       | 2.14        | Pytorch                                      | Track 1: Titan X Track 2,3,4: GTX 1080 | Track 1: flip/rotation ( $\times 8$ )                |
| HIT-VPC           | 0.26        | 0.2         | Matconvnet                                   | GTX 1080Ti                             | -  |
| ISP_Team          | -           | 2.1         | Tensorflow                                   | Titan X                                | -  |
| jingliting        | 1.27        | 0.72        | ???  | ???                                    | -  |
| join16            | -           | 4.12        | ???  | GTX 1080                               | -  |
| juanluisgonzales  | -           | 0.02        | ???  | ???                                    | -  |
| KAIST-VICLAB      | 0.44        | 1.60        | Track 1: Matconvnet, Track 2,3,4: Tensorflow | Titan Xp                               | Track 1: - Track 2,3,4: flip/rotation ( $\times 8$ ) |
| MCML              | 5.95        | 1.08        | Tensorflow                                   | GTX 1080                               | Track 1: flip/rotation ( $\times 8$ )                |
| Mrobot0           | 10          | -           | ???  | ???                                    | -  |
| NMH               | -           | 3.31        | ???  | ???                                    | -  |
| PDN               | -           | 13.07       | Pytorch                                      | 4 Titan Xp                             | Ensemble two variations of the proposed methods      |
| Pixel_Overflow    | 20          | -           | Tensorflow                                   | Nvidia P100                            | -  |
| rainbow           | 6.75        | -           | Pytorch                                      | GTX 1080Ti                             | flip/rotation ( $\times 8$ )                         |
| reveal.ai         | 92.95       | -           | Pytorch                                      | Tesla K80                              | flip/rotation ( $\times 8$ )                         |
| SIA               | 396.0       | 396.0       | Tensorflow                                   | CPU                                    | flip/rotation ( $\times 8$ )                         |
| srFans            | -           | 0.10        | Pytorch                                      | Tesla K80                              | -  |
| SRFun             | 1           | -           | Tensorflow                                   | GTX 1080Ti                             | -  |
| Toyota-TI         | 35          | -           | Pytorch                                      | Titan X                                | flip/rotation ( $\times 8$ )                         |
| UIUC-IFP          | 5.03        | 7.28        | Pytorch                                      | P100                                   | flip/rotation ( $\times 8$ )                         |
| UW18              | 300         | -           | Matlab                                       | Intel Core i7-6700K CPU @ 4.00GHz      | -  |
| zeweithe          | 1.02        | -           | ???  | ???                                    | -  |

**Runtime / efficiency** BOE-SBG reported the lowest runtime, 0.15s to super-resolve  $\times 8$  one LR image on GPU, but ranked 17th on ‘Classic Bicubic’ 0.63dB lower than the best ranked method of Toyota-TI. Among the top 4 methods on ‘Classic Bicubic’ track, rainbow achieved the best trade-off between efficiency and performance. On a GTX 1080Ti GPU, it takes 6.75s for rainbow, while 35s are necessary for Toyota-TI per LR image to generate the HR image, including self-ensemble for both methods.

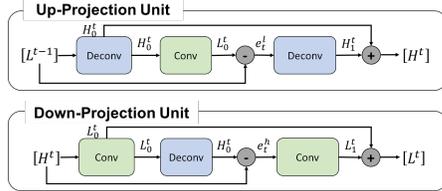
**Train data** Data augmentation by scaling (only Track 1), flipping, and rotation [33] is another commonly used technique. Only a couple of teams, including Pixel\_Overflow, used extra data for training. Pixel\_Overflow used images from [www.pexels.com](http://www.pexels.com), which is also the source of

many DIV2K images. HIT-VPC used Track 1 images to estimate downgrading operators on Tracks 3 and 4, thus their ‘lpj008’ entry in Table 1 is just for reference and not ranked in the challenge.

**Conclusions** By analyzing the settings, the proposed methods and their results we can conclude: (i) The proposed methods improve the state-of-the-art in SR. (ii) The top solutions are consistent across the realistic tracks, yet the top methods in ‘Classic Bicubic’ are not the top methods of the realistic tracks – domain specific knowledge (pre-alignment of train images) was critical. (iii) As expected, the realistic tracks are more challenging than the bicubic, reflected by the relatively lower PSNR (up to 2dB for the winners) of the results even if we compare  $\times 8$  with  $\times 4$ . (iv) SSIM is more



(a) DBPN architecture



(b) the up- and down-projection units in DBPN  
Figure 2. Toyota-TI’s DBPN network structure.

correlated with PSNR on ‘Classic Bicubic’ than on realistic tracks. (v) High magnification factors and realistic settings pose the extra problem of (subpixel) alignment between HR results and ground truth. (vi) Other ranking measures are necessary (such as perceptual ones). (vii) Further realistic challenges could introduce non-uniform degradations.

## 4. Challenge Methods and Teams

**4.1. Toyota-TI team** proposed a deep back-projection networks (DBPN) [9] (see Fig. 2) which uses error feedbacks from the up- and down-scaling steps to guide the network to achieve optimal result. Unlike the previous methods which predict the SR image in feed-forward manner, DBPN adopts mutually connected up- and down-sampling stages to generate LR as well as HR features, and accumulate both up- and down-projection errors to predicting the final SR results. A group of LR features are firstly extracted from the input LR image. Then, back-projection stages are utilized to alternatively generate LR and HR feature maps  $L^t$  and  $H^t$ , which further improved by dense connection where the input for each projection unit is the concatenation of the outputs from all previous units. At last, all the HR feature maps are utilized to reconstruct the final SR estimation  $I^{sr} = f_{Rec}([H^1, H^2, \dots, H^t])$ .

The structure of the newly introduced up-projection and down-projection units are shown in Fig. 2(b). To deal with classic bicubic  $\times 8$  downsampling SR problem, DBPN uses  $12 \times 12$  convolutional layer with eight striding and two padding in the projection units, and 19 projection units (10 up- and 9 down-projection units) have been adopted for generating the SR result.

The network is trained on images from DIV2K with augmentation [33]. At training phase, the input patch size is set to  $40 \times 40$  and the mini-batch size to 18. The model is trained with L1 loss using ADAM optimizer [18] with

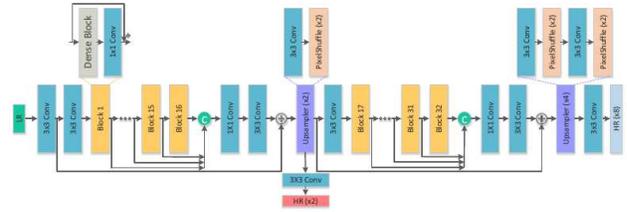


Figure 3. rainbow’s network architecture.

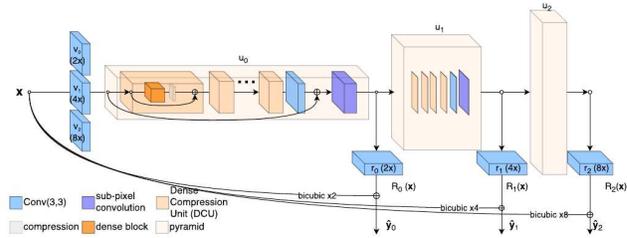


Figure 4. DRZ’s asymmetric pyramidal architecture with DCU.

learning rate  $1 \times 10^4$  and decrease by a factor of 10 for every  $5 \times 10^5$  iterations for total  $10^6$  iterations. In the testing phase, the authors adopt the self-ensemble strategy [33] to further improve the SR results.

**4.2. Pixel.Overflow team** [4] utilized the same network structure as EDSR [21]. To get better SR performance, external training data is adopted in the training phase. Pixel.Overflow uses Sobel filter to extract output and target image edges to emphasize loss on the edges and details.

**4.3. rainbow team** proposed a method based on EDSR [21] and SRDenseNet [11, 34] (Fig. 3). They employed a pyramid architecture to gradually generate the HR image. In order to trade-off the performance and the inference time, they adopted a two-step enlargement strategy. They trained the network with L1 loss and fine-tuned with L2 loss.

**4.4. DRZ team** proposed an asymmetric pyramidal structure for image SR [36] (see Fig. 4). Each level of the pyramid consists of a cascade of dense compression units (DCUs), and a sub-pixel convolution layer is utilized to generate the residual map to reconstruct the HR image. DCU consists of a smaller, modified densely connected block [11] followed by  $1 \times 1$  convolution. Compared with the original densely connected block proposed for classification, the batch normalization (BN) layer has been removed in DCU.

In the training phase, curriculum learning [5] strategy has been adopted to achieve better SR performance and shorter training time. Specifically, DRZ firstly trains the  $2 \times$  portion of the network and then gradually blend a new level of pyramid to reduce the impact on the previously trained layers. Curriculum learning adds an average of 0.07dB PSNR on the validation set of DIV2K for  $2 \times / 4 \times / 8 \times$  scales compared to 0.03dB using normal multiscale training.

**4.5. UIUC-IFP team** proposed a wide activation SR network (WDSR, see Fig. 5), which is a deep residual SR

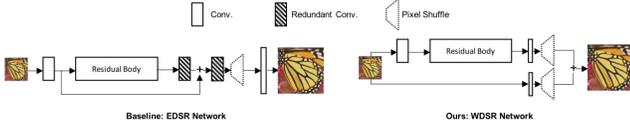
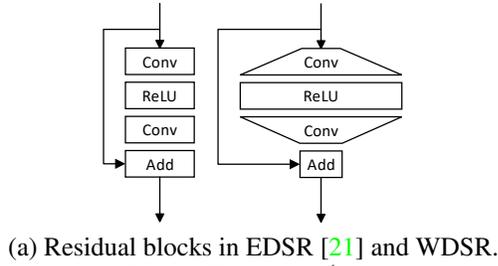


Figure 5. UIUC-IFP’s WDSR unit architecture.

network (two-layer residual blocks) similar to the baseline EDSR [21]. To improve the SR performance, WDSR modify the original EDSR in three aspects. Firstly, in comparison with EDSR, WDSR reduces the width of identity mapping pathway and increases the width of feature maps before the ReLU function in each residual block (see Fig. 5(a)). Their experiments showed that WDSR is extremely effective for improving accuracy. Secondly, UIUC-IFP follows recent works [8, 21, 31] which remove the BN layer in the residual blocks and adopts weight normalization in their WDSR approach, although the introducing of weight normalization in training SR networks may not help that much, it enables the authors to use higher learning rate to train the network. Thirdly, WDSR removes some convolution layers used in EDSR and directly generate the shuffled SR estimation (see Fig. 5(b)), such a strategy is able to improve the processing speed while not affect accuracy of SR network.

For Track 1, UIUC-IFP utilized similar training parameters as EDSR, the only difference is that weight normalization enables UIUC-IFP to increase the learning rate  $10\times$  to 0.001. After training with L1 loss, the model is finetuned with PSNR loss, directly. The finetune step leads to around 0.03dB PSNR improvement on the DIV2K validation set. For Tracks 2, 3 and 4, UIUC-IFP utilized a pre-align step to alleviate the random shift effects between the LR and HR images. Specifically, the HR images are shifted up to 40 pixels, and then bicubic downscaled HR images are compared with given realistic LR images to find coarse aligned HR images for each LR image.

In the testing phase, a self-ensemble inference strategy has been adopted to improve SR performance [33].

**4.6. PDN team** proposed the PolyDenseNet (PDN) (see Fig. 6) for image SR. The basic building block of PDN is PolyDense Block (PDB), which is motivated by PolyNet [42] and DenseNet [11]. Each PDB contains three 5-layer dense block and use three parameters  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  to combine the dense block outputs  $D_1$ ,  $D_2$  and  $D_3$  to

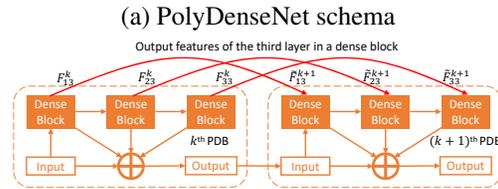
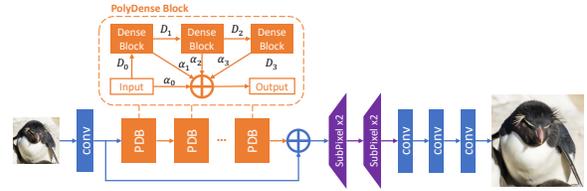
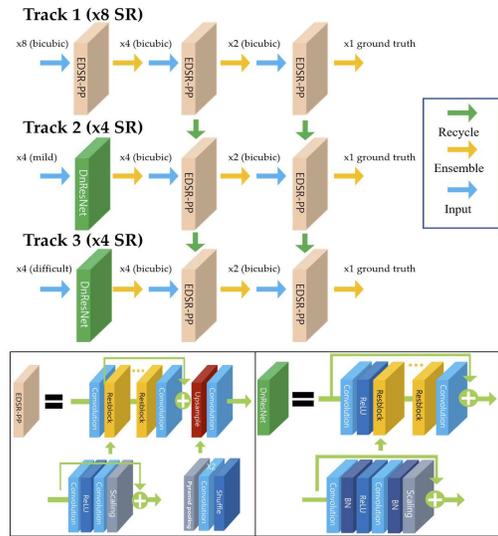
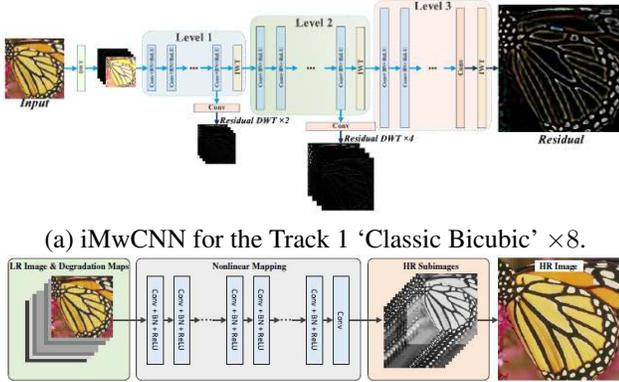


Figure 6. PDN’s PolyDenseNet, a variant of PolyDenseBlock.

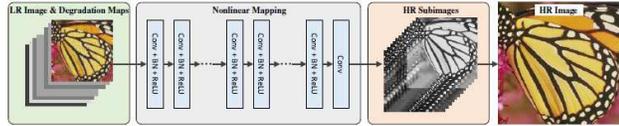


get the output. PDN team investigated also a PDN variant by building skip connections between adjacent PDBs (see Fig. 6(b)). The results by the two variants are ensemble at test time. In the training phase, the authors upsample the LR images and calculate the best shifting parameters w.r.t. ground truth based on PSNR. For brightness scaling, the authors adjust the pixel mean of LR images by the mean of its corresponding ground-truth image.

**4.7. BMIPL\_UNIST team** decomposed the original problems of NTIRE 2018 challenge into subproblems (SR at various scales and denoising / deblurring) and proposed an efficient module-based single image SR network [27] (EM-BSR, see Fig. 7). For an individual module network on SR, they proposed EDSR-PP which integrated pyramid pooling into the upsampling layer of EDSR [21] for better utilizing both the global and local context information. For a module network on denoising / deblurring, they proposed a residual convolution network (DnResNet) which replaced convolution blocks of DnCNN [40] by residual blocks with BN and



(a) iMwCNN for the Track 1 ‘Classic Bicubic’  $\times 8$ .



(b) SRMD for the realistic settings: Track 2, 3, and 4.

Figure 8. HIT-VPC’s solutions.

scaling. A pre-processing step aligning the input and target images have been adopted in the training process of DnResNet, which is reportedly critical for good performance.

**4.8. HIT-VPC team** utilized different strategies for solving the bicubic and realistic experimental settings. For Track 1 ‘Classic bicubic’  $\times 8$ , HIT-VPC proposed an inverse multi-level wavelet convolutional neural network (iMwCNN). As shown in Fig. 8(a), iMwCNN is designed as pyramid structure with multi-level wavelet packet transform (WPT) [22]. The input LR image is firstly bicubic interpolated by a scale factor 2, and the DWT coefficients of the interpolated image are the network input. To get a scale factor of 8, 3-level networks have been adopted for estimating the inverse DWT coefficients. Between each level of networks, a fixed inverse wavelet transform is adopted to transform the coefficients back to the image space. Each level of network contains 8 convolutional layers, and feature map number for the three levels are set as 256, 256 and 128, respectively. In the training phase, the loss is defined on each scale of estimations.

For the realistic settings (Tracks 2, 3, and 4), HIT-VPC built upon their recently proposed super-resolution network for multiple degradations (SRMD) [39]. As illustrated in Fig. 8(b), SRMD takes the parameterized degradation map as well as LR image as the network inputs, and utilizes 20 convolution + BN + ReLU blocks to estimate the HR subimages. In order to apply SRMD to tracks 2, 3 and 4, the blur kernel of which is unknown, HIT-VPC centers the blur kernels based on the largest values to align the LR image and HR image, and calculates the mean (aligned) degradation maps for each track. Then, the mean degradation maps for each track is used for super-resolve images from the corresponding tracks.

HIT-VPC\* (‘lpj008’, not ranked) submitted additional results of a single SRMD model for Tracks 3 and 4. They used Track 1 images for degradation operator estimation and showed the advantages of non-blind SRMD: (i) it can handle Tracks 3 and 4 in a single model while (ii) produc-

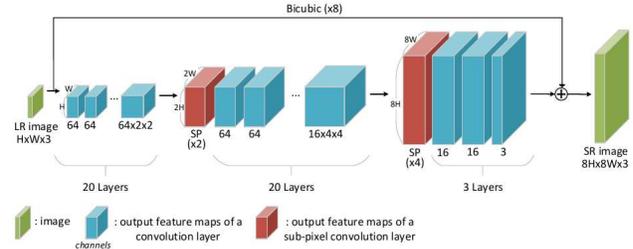


Figure 9. KAIST-VICLAB’s proposed network architecture.

ing better results with accurate blur kernel than the blind SRMD.

**4.9. Faceall\_Xlabs team**’s architecture is based on EDSR [21]. The filter number for each convolution layer has been changed to 256, and 80 residual blocks were used.

**4.10. Duke Data Science team** [4] also adopted different strategies for the bicubic and realistic settings. For the bicubic setting, they utilized EDSR [21] with a different training strategy. Warm restarts and cosine annealing approach has been introduced to allow the network to jump out the local minima.

For the realistic settings, the authors firstly trained a DnCNN [40] and an EDSR [21] for denoising and SR separately, and then finetuned the two networks in tandem.

**4.11. SIA team** reproduced EDSR [21] and used Charbonnier loss instead of L1 loss, as suggested in [19]. To take full advantage of convolution operation for the full image, the CPU was used at test time. For Tracks 2, 3, and 4, the train pairs were first aligned based on PSNR.

**4.12. KAIST-VICLAB team** [15] designed, for Track 1, a 43-layer CNN (see Fig. 9) for progressively upscaling the input RGB image to the final target resolution. Two sub-pixel convolution layers are inserted after the 20-th and the 40-th convolution layer to enlarge the feature maps by 2 and 4, resp. The network is trained in a coarse-to-fine manner: first for  $2\times$  upscaling, then for  $4\times$ , and finally for  $8\times$ .

For Tracks 2, 3 and 4, KAIST-VICLAB developed a solution comprised from: 1) Four  $5\times 5$ -sized filters are learned between LR and HR training sub-images, which are applied to create noise-reduced and luminance-corrected intermediate images of HR sizes. 2) Aligned HR training images are generated by aligning original HR training images with the intermediate images. 3) A 58-layered CNN with 2M parameters is trained using the noisy LR training images and the newly aligned HR training images. Specifically, residual learning, residual units and two subpixel convolution layers are adopted in the network.

**4.13. Haiyun\_XMU team** [6] adopted different strategies for the bicubic and realistic settings. For Track 1 an EDSR [21]-based model with 50 residual blocks were trained to reconstruct the HR image. While, for the realistic settings, Haiyun\_XMU design the persistent memory

block [30] in two ways and then embed it into EDSR [21]. First, for MemEDSR, the authors replace the body part of EDSR with a memory block with 4 residual blocks, and each memory module links to the gate unit, which adaptively selects the features needed to store. Second, for IRMem, the authors design a memory block with an IR-CNN [41] block, which delete all the BN layer and add residual factor assigned 0.1, and embed this memory block into EDSR [21]. The MemEDSR is adopted in Tracks 2 and 3, while the IRMem is adopted in Track 4.

**4.14. Ajou-LAMDA-Lab team** proposed progressive CARN [3] which apply progressive training [14] based on the CARN [2]. Specifically, the structure of the CARN module is shown in Fig. 10 (b), the local and global cascading modules are expected to extract multi-level representations. Upon this, three-stage of CARN modules are progressively trained to reconstruct the  $\times 8$  HR image as depicted in Fig. 10 (a). In the training process, extra CARN module is added after at the end of the stage and replace the previous reconstruction layers with the one that produces the image in double resolution. Further, learning rate of pre-trained modules is decayed ten times to stabilize overall training.

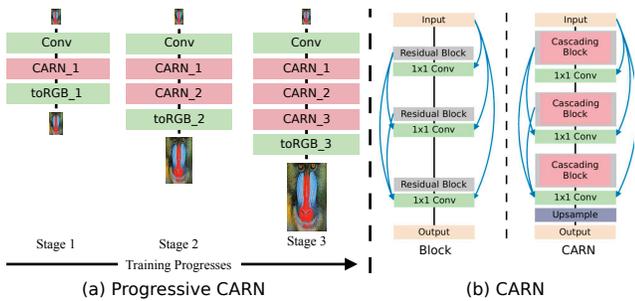


Figure 10. Ajou-LAMDA-Lab's networks.

**4.15. srFans team** improved EDSR [21] in two aspects: (i) the number of residual blocks have been changed to 30 for generating sharper details, and (ii) the first convolution layer in each res-block has been changed to dilated convolution with size 2. srFans adopted a pre-processing step to align the LR and HR image pairs before training.

**4.16. DeepSR team** proposed a deeply progressive memory network (DPMN). Specifically, the authors utilized convolution, summation, concatenation and deconvolution layers to build the DPMN block, and used 5 DPMN blocks to deal with the SR problem (see Fig. 11). Each convolution layer is followed by a leaky rectified unit (LReLU).

**4.17. reveal.ai team** proposed a network structure based on the DenseNet [11] (see Fig. 12. reveal.ai also introduced dense connections across dense blocks. Furthermore, each layer inside the dense-block is changed to a convolution + ReLU + convolution layer, which is similar to EDSR [21].

**4.18. ISP\_Team** used blur maps to improve the SR results in realistic conditions. Specifically, after estimating the blur

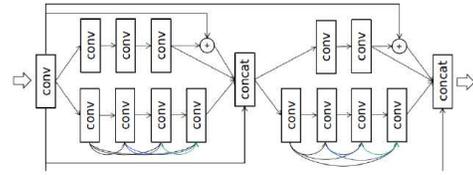


Figure 11. DeepSR's proposed DPMN unit.

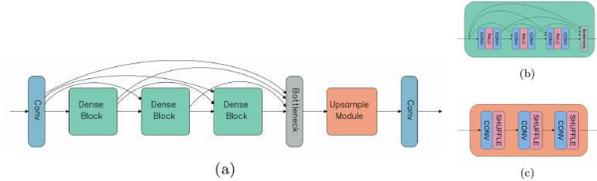


Figure 12. reveal.ai's network architecture.

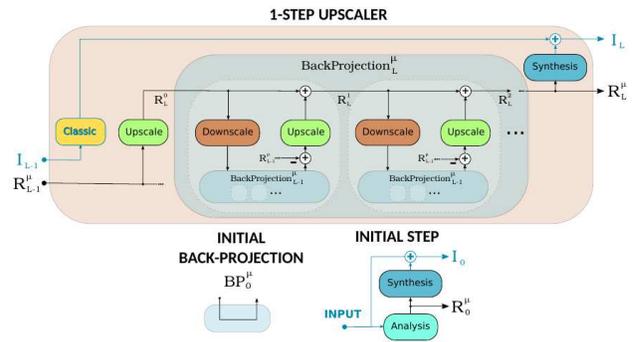


Figure 13. BOE team's Multigrid Back-projection Recursion.

map of testing image, ISP\_Team puts the blur map as well as the LR image into the SR-net to reconstruct the HR image.

**4.19. BOE-SBG team** developed a multi-scale SR system (see Fig. 13), for zooming factor 8 and 4, 3 and 2 levels of  $\times 2$  network are utilized to progressively upscale a LR image. In each scale of upsampling network, the authors adopted a multi-grid version of iterative back-projections [13, 35] in the latent space (e.g. features within a network) to further improve the SR performance. The authors utilized the learned upscale and downscale layers to improve the SR estimation with the LR residuals. In the upscale and downscale steps, the stride convolution/deconvolution operations and the newly proposed Muxout and TransposedMuxout [26, 24, 25] layers have been adopted for track 1, 3 and 2, 4, respectively.

**4.20. MCML team** proposed a network architecture [17] based on MDSR [21]. To improve the performance of MDSR [21], they proposed enhanced upscaling module (EUM) shown in Fig. 14. Compared with the original upscaling layer in MDSR, which uses only one convolution layer without an activation function to increase the number of features, they introduce four residual modules and concatenate the outputs of the modules to increase the number of feature maps. The proposed EUM has the advantages that it can handle nonlinear operations and exploit skip con-

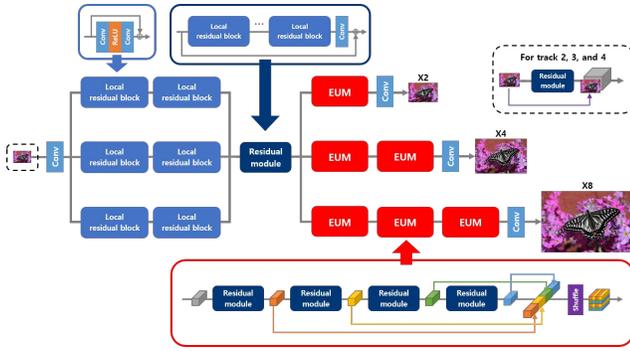


Figure 14. The network architecture used by MCML.

nections. They proposed a novel deep residual network for super-resolution (EUSR) by utilizing the EUM and multi-scale learning ( $\times 2$ ,  $\times 4$ , and  $\times 8$ ), whose structure is illustrated in Fig. 14.

For track 1, they used 48 residual blocks and 2 residual blocks in each residual module for feature extraction and upscaling, respectively. They used a single EUSR for the other three tracks, which exploit the information of multiple degradation processes (mild, difficult, and wild) instead of the multiple scales. They used 64 residual blocks and 2 residual blocks in each residual module for feature extraction and upscaling, respectively. They also used three additional feature maps as input, which are obtained by a residual module consisting of three residual blocks. The self-ensemble strategy [33] is adopted for track 1 at testing.

**4.21. SRFun team** divided the  $8\times$  SR problem into  $3 \times 2\times$  SR problems [12]. A modified version of ResNet and DenseNet deep autoencoder has been utilized to estimate the residual between target HR image and a pre-defined up-sampled image.

**4.22. APSARA team** used the LR image to estimate the wavelet coefficients of HR image. Since each sub-band map of HR wavelet coefficients are with the same size of LR image, the proposed network (see Fig. 15) do not need deconvolution or subpixel layers. The network uses the first 32 residual blocks and a  $1 \times 1$  convolution layer to compute the detail coefficients of HR image and utilize another 32 residual blocks to compute the approximation coefficients of HR image. Then, the two component are combined together to generate the final HR reconstruction.

**4.23. CEERI team** proposed an improved residual based gradual upscaling network (IRGUN) [29]. The IRGUN has a series of up-scaling and enhancement blocks (UEB) connected end-to-end and fine-tuned together to give a gradual magnification and enhancement. The up-scaling network is a 6 layer architecture, which contains 3 convolutional layers followed by 3 de-convolutional layers. While the enhancement network contains 10 layer residual enhancement network (RED-10) [23]. The authors repeated the UEB until they reach the required SR factor.

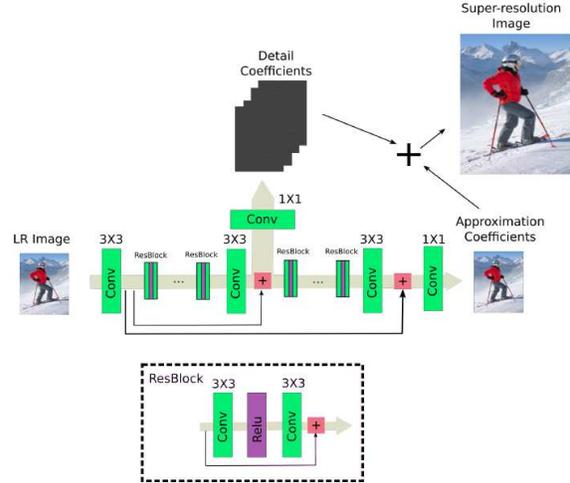


Figure 15. APSARA's WaveletSR architecture.

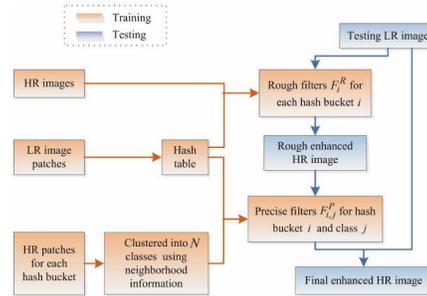


Figure 16. UW18's TSSR diagram.

Furthermore, the authors firstly perform the SR operation only on the Y channel of the YCbCr format of input image, and then combine the super-resolved Y channel with the bicubic enlarged CbCr channel. After transforming the image back to the RGB space, a 10 layers network has been adopted for further enhance the output image.

**4.24. UW18 team** proposed a two-step SR (TSSR) approach (see Fig. 16) which utilizes two successive resolution enhancement operation to super-resolve the input LR image. In the first step, a set of rough filters, which is based on the hash mechanism proposed in RAISR [28], is utilized to generate a coarse SR result. Then, a set of refined resolution enhancement filters are applied to yield the final HR patches.

**4.25 NMH team** improved EDSR [21] by expanding the number of feature maps before the last convolution layer to 512 and introducing a  $1 \times 1$  convolution layer to generate the HR reconstruction.

## Acknowledgements

We thank the NTIRE 2018 sponsors: Alibaba Group, NVIDIA Corp., SenseTime Group Ltd., Huawei Technologies Co. Ltd., Code Ocean, Google Inc., Disney Research, Amazon.com, Inc., and ETH Zurich.

## A. Teams and affiliations

### NTIRE2018 team

**Title:** NTIRE 2019 Challenge on example-based single image super-resolution

**Members:** Radu Timofte<sup>1,2</sup>

([radu.timofte@vision.ee.ethz.ch](mailto:radu.timofte@vision.ee.ethz.ch)), Shuhang Gu<sup>1</sup>, Jiqing Wu<sup>1</sup>, Luc Van Gool<sup>1,3</sup>, Lei Zhang<sup>4</sup>, Ming-Hsuan Yang<sup>5</sup>

**Affiliations:**

<sup>1</sup> Computer Vision Lab, ETH Zurich, Switzerland

<sup>2</sup> Merantix, Germany

<sup>3</sup> ESAT, KU Leuven, Belgium

<sup>4</sup> The Hong Kong Polytechnic University, China

<sup>5</sup> University of California at Merced, US

### Ajou-LAMDALab

**Title:** Image Super-resolution via Progressive Cascading Residual Network

**Members:** Namhyuk Ahn

([aa0dfg@ajou.ac.kr](mailto:aa0dfg@ajou.ac.kr)), Byungkon Kang, Kyung-Ah Sohn

**Affiliation:**

Ajou University, Republic of Korea

### APSARA

**Title:** Single Image Super Resolution based on Wavelet

**Members:** Ming Qiu ([mingqiu@xmu.edu.cn](mailto:mingqiu@xmu.edu.cn)), Liting Jing, Jiehang Zeng, Ying Wang

**Affiliation:**

Software School of Xiamen University, China

### BMIPL\_UNIST

**Title:** Efficient Module based Single Image Super Resolution for Multiple Problems

**Members:** Dongwon Park

([dong1@unist.ac.kr](mailto:dong1@unist.ac.kr)), Kwanyoung Kim, Se Young Chun

**Affiliation:**

School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology (UNIST), Republic of Korea

### BOE-SBG

**Title:** Multi-Grid Back-Projection Networks for Image SuperResolution

**Members:** Pablo Navarrete Michelini

([pnavarre@boe.com.cn](mailto:pnavarre@boe.com.cn)), Dan Zhu, Hanwen Liu

**Affiliation:**

BOE Technology Group Co., Ltd.

### CEERI

**Title:** Improved residual based gradual upscaling network(IRGUN)

**Members:** Manoj Sharma, Rudrabha Mukhopadhyay, Avinash Upadhyay, Sriharsha Koundinya, Ankit Shukla, Santanu Chaudhury

**Affiliation:**

CSIR-CEERI, India

### DeepSR

**Title:** Deeply Progressive Memory Network for Image Restoration

**Members:** Yan Zhao

([N161127032@fzu.edu.cn](mailto:N161127032@fzu.edu.cn)), Wei Deng

**Affiliation:**

Fuzhou University, China

### DRZ

**Title:** A Fully Progressive Approach to Single-Image Super-Resolution

**Members:** Yifan Wang<sup>1,2</sup>

([yifan.wang@disneyresearch.com](mailto:yifan.wang@disneyresearch.com)), Federico Perazzi<sup>1</sup>, Brian McWilliams<sup>1</sup>, Alexander Sorkine-Hornung<sup>3</sup>, Olga Sorkine-Hornung<sup>2</sup>, Christopher Schroers

**Affiliation:**

<sup>1</sup>Disney Research,

<sup>2</sup>ETH Zurich, Switzerland

<sup>3</sup>Facebook Oculus

### Duke Data Science

**Title:** CosEDSR

**Members:** Alexandru Damian

([ad315@duke.edu](mailto:ad315@duke.edu)), Nikhil Ravi, Sachit Menon, Cynthia Rudin

**Affiliation:**

Duke University, US

### Faceall\_Xlabs

**Members:** Jinchang Xu

([xjc1@bupt.edu.cn](mailto:xjc1@bupt.edu.cn)), Yijiao Liu, Fengye Xiong, Yuan Dong, Hongliang Bai

**Affiliation:**

Beijing University of Posts and Telecommunications, China

Beijing Faceall Technology Co., Ltd, China

## Haiyun\_xmu

**Title:** Persist Memory Residual Network for Super Resolution

**Members:** Rong Chen

(*chenrong\_mail@qq.com*), Kun Zeng, Jinkang Guo, Yanyun Qu, Cuihua Li

**Affiliation:**

Xiamen University, China

## HIT-VPC

**Title:** Learning a Single Convolutional Super-Resolution Network for Multiple Degradations

**Members:** Kai Zhang

(*cskai Zhang@gmail.com*), Pengju Liu, Wangmeng Zuo, Shi Guo, Jiye Liu

**Affiliation:**

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

## ISP\_Team

**Members:** Xiaowang Cai

(*460571930@qq.com*), Fang Huang, Yueshu Xu

**Affiliation:**

## JuanLuisGonzales

**Title:** Receptive Field ESPCN

**Members:** Juan Luis Gonzalez

(*juanluisgb@kaist.ac.kr*)

**Affiliation:**

KAIST, Republic of Korea

## KAIST-VICLAB

**Title:** Track 1: A Progressive Super-resolution Method using Convolutional Neural Networks

Track 2, 3, 4: Convolutional Neural Networks for Alignment and Super-Resolution

**Members:** Soo Ye Kim

(*sooyekim@kaist.ac.kr*), Jae-Seok Choi, Sehwan Ki, Soomin Seo, Hyeonjun Sim, Saehun Kim, Munchurl Kim

**Affiliation:**

Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea

## MCML

**Title:** Deep residual network using enhanced upscale modules for super-resolution

**Members:** Jun-Hyuk Kim

(*junhyuk.kim@yonsei.ac.kr*), Jong-Seok Lee

**Affiliation:**

Yonsei University, Republic of Korea

## NMH

**Members:** Lingzhi Fu

(*lexuszhi1990@gmail.com*)

## PDN

**Title:** Deep Poly-Dense Network for Image Super-Resolution

**Members:** Xintao Wang<sup>1</sup>

(*wx016@ie.cuhk.edu.hk*), Ke Yu<sup>1</sup>, Tak-Wai Hui<sup>1</sup>, Chao Dong<sup>2</sup>, Liang Lin<sup>2</sup>, Chen Change Loy<sup>1</sup>

**Affiliation:**

<sup>1</sup>The Chinese University of Hong Kong,

<sup>2</sup>SenseTime Research

## Pixel\_Overflow

**Title:** x8 super-resolution, explored the effect of edge loss using Sobel filter

**Members:** Shijia Hu

(*sh395@duke.edu*), Yijie Bei

**Affiliation:** Duke University, US

## rainbow

**Title:** Image Super-Resolution via Deep Pyramidal Residual Network (DPRN)

**Members:** Zheng Hui

(*zheng\_hui@stu.xidian.edu.cn*), Xiao Jiang, Yanan Gu, Jie Liu

**Affiliation:**

School of Electronic Engineering, Xidian University, China

## reveal.ai

**Title:** Denseption for Single-Image Super Resolution

**Members:** Sibt Ul Hussain

(*sibtul.hussain@nu.edu.pk*), Muneeb Aadil, Rafia Rahim

**Affiliation:**

National University of Computer & Emerging Sciences, Islamabad, Pakistan

## SIA

**Title:** Reproducing EDSR

**Members:** Junghoon Seo

(*sjh@satreci.com*), Taegyun Jeon, Jamyoun Koo, Seunghyun Jeon

**Affiliation:**  
R&D Center, Satrec Initiative

## srFans

**Title:**Dilated Deeper Residual Network for Super Resolution

**Members:***Yuan Yuan*

(*yyuan13@ualberta.ca*), Jiawei Zhang, Jiahao Pang , Xiangyu Xu

**Affiliation:**

Electrical and Computer Engineering, University of Alberta, Canada

## SRFun

**Title:** Densely Connected High Order Residual Network for Single Frame Image Super Resolution

**Members:***Yiwen Huang*

(*nickgray0@gmail.com*)

**Affiliation:**

Wenhua College, China

## Toyota-TI

**Title:** Deep Back-Projection Networks

**Members:***Muhammad Haris*<sup>1</sup>

(*mharis@toyota-ti.ac.jp*), Greg Shakhnarovich<sup>2</sup>, Norimichi Ukita<sup>1</sup>

**Affiliation:**

<sup>1</sup> Toyota Technological Institute

<sup>2</sup> TTI-Chicago, US

## UIUC-IFP

**Title:** Wide Activation and Weight Normalization for Accurate Image SuperResolution

**Members:***Jiahui Yu*<sup>1</sup>

(*jyu79@illinois.edu*), Yuchen Fan<sup>1</sup>, Jianchao Yang<sup>2</sup>, Ning Xu<sup>2</sup>, Zhaowen Wang<sup>3</sup>, Xinchao Wang<sup>4</sup>, Thomas S. Huang<sup>1</sup>

**Affiliation:**

<sup>1</sup>University of Illinois at Urbana-Champaign, US

<sup>2</sup>Snap Inc.,

<sup>3</sup>Adobe Research,

<sup>4</sup>Stevens Institute of Technology, US

## UW18

**Title:**TSSR(Two-Step Super Resolution)

**Members:***Zhe Zhang*<sup>1,2</sup>

(*zhangzsmg@gmail.com*), Yu Hen Hu<sup>1</sup>

**Affiliation:**

<sup>1</sup>Dept. Electrical and Computer Engineering, University of

Wisconsin-Madison, Madison, US

<sup>2</sup>MOE KLINNS Lab, Institute of Integrated Automation, School of Electronic and Information Engineering, Xian Jiaotong University, China

## References

- [1] E. Agustsson and R. Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 1, 2
- [2] N. Ahn, B. Kang, and K.-A. Sohn. Fast, accurate, and, lightweight super-resolution with cascading residual network. *arXiv preprint arXiv:1803.08664*, 2018. 7
- [3] N. Ahn, B. Kang, and K.-A. Sohn. Image super-resolution via progressive cascading residual network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. 7
- [4] Y. Bei, A. Damian, S. Hu, S. Menon, N. Ravi, and C. Rudin. New techniques for preserving global structure and denoising with low information loss in single-image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 4, 6
- [5] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009. 4
- [6] R. Chen, Y. Qu, K. Zeng, J. Guo, L. Cui-hua, and Y. Xie. Persistent memory residual network for single image super resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. 6
- [7] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199. Springer, 2014. 1
- [8] Y. Fan, H. Shi, J. Yu, D. Liu, W. Han, H. Yu, Z. Wang, X. Wang, and T. S. Huang. Balanced two-stage residual networks for image super-resolution. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1157–1164. IEEE, 2017. 5
- [9] M. Haris, G. Shakhnarovich, and N. Ukita. Deep back-projection networks for super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4
- [10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2
- [11] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017. 2, 4, 5, 7
- [12] Y. Huang and M. Qin. Densely connected high order residual network for single frame image super resolution. *arXiv preprint arXiv:1804.05902*, 2018. 8
- [13] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 53(3):231–239, 1991. 7

- [14] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations (ICLR)*, 2018. 7
- [15] S. Ki, H. Sim, S. Y. Kim, J.-S. Choi, S. Kim, and M. Kim. Fully end-to-end learning based conditional boundary equilibrium gan with receptive field sizes enlarged for single ultra-high resolution image dehazing. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018. 6
- [16] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1
- [17] J.-H. Kim and J.-S. Lee. Deep residual network with enhanced upscaling module for super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. 7
- [18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4
- [19] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 624–632, 2017. 6
- [20] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016. 1
- [21] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 1, 2, 4, 5, 6, 7, 8
- [22] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989. 6
- [23] X. Mao, C. Shen, and Y. Yang. Image restoration using very deep fully convolutional encoder-decoder networks with symmetric skip connections. *CoRR*, abs/1603.09056, 2016. 8
- [24] P. Navarrete and H. Liu. Convolutional networks with muxout layers as multi-rate systems for image upscaling. *CoRR*, abs/1705.07772, 2017. 7
- [25] P. Navarrete and H. Liu. Upscaling beyond super-resolution using a novel deep learning system. *GPU Technology Conference*, March 2017. 7
- [26] P. Navarrete, L. Zhang, and J. He. Upscaling with deep convolutional networks and muxout layers. In *GPU Technology Conference 2016*, Poster Session, San Jose, CA, USA, May 2016. 7
- [27] D. Park, K. Kim, and S. Y. Chun. Efficient module based single image super resolution for multiple problems. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. 5
- [28] Y. Romano, J. Isidoro, and P. Milanfar. Rair: rapid and accurate image super resolution. *IEEE Transactions on Computational Imaging*, 3(1):110–125, 2017. 8
- [29] M. Sharma, R. Mukhopadhyay, A. Upadhyay, S. Koundinya, A. Shukla, and S. Chaudhury. Irgun : Improved residue based gradual up-scaling network for single image super resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. 8
- [30] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4539–4547, 2017. 7
- [31] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 1, 2, 5
- [32] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In D. Cremers, I. Reid, H. Saito, and M.-H. Yang, editors, *12th Asian Conference on Computer Vision*, 2014. 1
- [33] R. Timofte, R. Rothe, and L. Van Gool. Seven ways to improve example-based single image super resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 3, 4, 5, 8
- [34] T. Tong, G. Li, X. Liu, and Q. Gao. Image super-resolution using dense skip connections. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4809–4817. IEEE, 2017. 4
- [35] U. Trottenberg and A. Schuller. *Multigrid*. Academic Press, Inc., Orlando, FL, USA, 2001. 7
- [36] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers. A fully progressive approach to single-image super-resolution. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018. 4
- [37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. 2
- [38] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. 1
- [39] K. Zhang, W. Zuo, and Z. Lei. Learning a single convolutional super-resolution network for multiple degradations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 6
- [40] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep cnn for image denoising. *arXiv preprint arXiv:1608.03981*, 2016. 5, 6
- [41] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. *arXiv preprint*, 2017. 7
- [42] X. Zhang, Z. Li, C. C. Loy, and D. Lin. Polynet: A pursuit of structural diversity in very deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3900–3908. IEEE, 2017. 5