PIRM Challenge on Perceptual Image Enhancement on Smartphones: Report

Andrey Ignatov[1(✉)], Radu Timofte[1], Thang Van Vu[2], Tung Minh Luu[2], Trung X Pham[2], Cao Van Nguyen[2], Yongwoo Kim[3], Jae-Seok Choi[3], Munchurl Kim[3], Jie Huang[4], Jiewen Ran[4], Chen Xing[4], Xingguang Zhou[4], Pengfei Zhu[4], Mingrui Geng[4], Yawei Li[1], Eirikur Agustsson[1], Shuhang Gu[1], Luc Van Gool[1], Etienne de Stoutz[12], Nikolay Kobyshev[12], Kehui Nie[5], Yan Zhao[5], Gen Li[6], Tong Tong[6], Qinquan Gao[5], Liu Hanwen[11], Pablo Navarrete Michelini[11], Zhu Dan[11], Hu Fengshuo[11], Zheng Hui[7], Xiumei Wang[7], Lirui Deng[8], Rang Meng[9], Jinghui Qin[13], Yukai Shi[13], Wushao Wen[13], Liang Lin[13], Ruicheng Feng[10], Shixiang Wu[10], Chao Dong[10], Yu Qiao[10], Subeesh Vasu[14], Nimisha Thekke Madam[14], Praveen Kandula[14], A. N. Rajagopalan[14], Jie Liu[15], and Cheolkon Jung[15]

[1] Computer Vision Lab, ETH Zurich, Zürich, Switzerland
[2] Department of Electrical Engineering, KAIST, Daejeon, Republic of Korea
[3] Video and Image Computing Lab, KAIST, Daejeon, Republic of Korea
[4] Meitu Imaging & Vision Lab, Xiamen, China
[5] Fuzhou University, Fuzhou, China
[6] Imperial Vision, Fuzhou, China
[7] Xidian University, Xi'an, China
[8] Tsinghua University, Beijing, China
[9] Zhejiang University, Hangzhou, China
[10] Shenzhen Institute of Advanced Technology, Shenzhen, China
[11] BOE Technology Group Co., Ltd., Beijing, China
[12] ETH Zurich, Zürich, Switzerland
[13] Sun Yat-sen University, Guangzhou, Switzerland
[14] Indian Institute of Technology Madras, Chennai, India
[15] School of Electronic Engineering, Xidian University, Xi'an, China

Abstract. This paper reviews the first challenge on efficient perceptual image enhancement with the focus on deploying deep learning models on smartphones. The challenge consisted of two tracks. In the first one, participants were solving the classical image super-resolution problem with a bicubic downscaling factor of 4. The second track was aimed at real-world photo enhancement, and the goal was to map low-quality photos from the iPhone 3GS device to the same photos captured with a DSLR camera.

A. Ignatov and R. Timofte ({andrey,radu.timofte}@vision.ee.ethz.ch, ETH Zurich) are the challenge organizers, while the other authors participated in the challenge. The Appendix contains the authors' teams and affiliations. PIRM 2018 Challenge webpage: http://ai-benchmark.org.

L. Leal-Taixé and S. Roth (Eds.): ECCV 2018 Workshops, LNCS 11133, pp. 315–333, 2019.
https://doi.org/10.1007/978-3-030-11021-5_20

The target metric used in this challenge combined the runtime, PSNR scores and solutions' perceptual results measured in the user study. To ensure the efficiency of the submitted models, we additionally measured their runtime and memory requirements on Android smartphones. The proposed solutions significantly improved baseline results defining the state-of-the-art for image enhancement on smartphones.

Keywords: Image enhancement · Image super-resolution
Challenge · Efficiency · Deep learning · Mobile
Android · Smartphones

1 Introduction

The majority of the current challenges related to AI and deep learning for image restoration and enhancement [3,4,6,28,32,35] are primarily targeting only one goal—high quantitative results measured by mean square error (MSE), peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), mean opinion score (MOS) and other similar metrics. As a result, the general recipe for achieving top results in these competitions is quite similar: more layers/filters, deeper architectures and longer training on dozens of GPUs. However, one question that might arise here is whether often marginal improvements in these scores are actually worth the tremendous computational complexity increase. Maybe it is possible to achieve very similar perceptual results by using much smaller and resource-efficient networks that can run on common portable hardware like smartphones or tablets. This question becomes of special interest due to the uprise of many machine learning and computer vision problems directly related to these devices, such as image classification [10,31], image enhancement [13,14], image super-resolution [8,34], object tracking [11,38], visual scene understanding [7,21], face detection and recognition [20,26], etc. A detailed description of the smartphones' hardware acceleration resources that can be potentially used for deep learning and mobile machine learning frameworks are given in [15].

The PIRM 2018 challenge on perceptual image enhancement on smartphones is the first step towards benchmarking resource-efficient architectures for computer vision and deep learning problems targeted at high perceptual results and deployment on mobile devices. It considers two classical computer vision problems—image super-resolution and enhancement, and introduces specific target performance metrics that are taking into account both networks' runtime, their quantitative and qualitative visual results. In the next sections we describe the challenge and the corresponding datasets, present and discuss the results and describe the proposed methods.

2 PIRM 2018 Challenge

The PIRM 2018 challenge on perceptual image enhancement on smartphones has the following phases:

i *development:* the participants get access to the data;
ii *validation:* the participants have the opportunity to validate their solutions on the server and compare the results on the validation leaderboard;
iii *test:* the participants submit their final results, models, and factsheets.

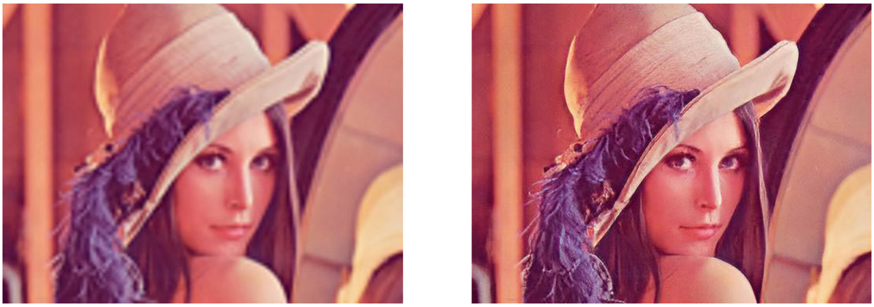

Fig. 1. A low-res image (left) and the same image super-resolved by SRGAN (right).

The PIRM 2018 challenge on perceptual image enhancement on smartphones consists of two different tracks described below.

2.1 Track A: Image Super-Resolution

The first track is targeting a conventional super-resolution problem, where the goal is to reconstruct the original image based on its bicubically downscaled version. To make the task more practical, we consider a downscaling factor of 4,

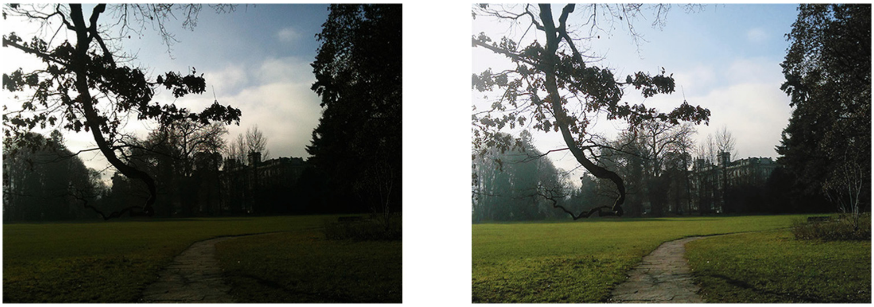

Fig. 2. The original iPhone 3GS photo (left) and the same image enhanced by the DPED network [13] (right).

some sample results for which obtained with SRGAN network [19] are shown in the Fig. 1. To train deep learning models, the participants used DIV2K dataset [1] with 800 diverse high-resolution train images crawled from the Internet.

2.2 Track B: Image Enhancement

The goal of the second track is to automatically improve the quality of photos captured with smartphones. In this task, we used DPED [13] dataset consisting of several thousands of images captured simultaneously with three smartphones and one high-end DSLR camera. Here we consider only a subtask of mapping photos from a very old iPhone 3GS device into the photos from Canon 70D DSLR. An example of the original and enhanced DPED test images are shown in the Fig. 2.

3 Scoring and Validation

The participants were required to submit their models as TensorFlow *.pb* files that were later run on the test images and validated based on three metrics:

- Their speed on HD-resolution (1280 × 720 pixels) images measured compared to the baseline SRCNN [8] network,
- PSNR metric measuring their fidelity score,
- MS-SSIM [37] metric measuring their perceptual score.

Though MS-SSIM scores are known to correlate better with human image quality perception than PSNR, they are still often not reflecting many aspects of real image quality. Therefore, during the final test phase we conducted a user study involving more than 2000 participants (using MTurk platform[1]) that were asked to rate the visual results of all submitted solutions, and the resulting Mean Opinion Scores (MOS) then replaced MS-SSIM results. For Track B methods, the participants in the user study were invited to select one of four quality levels (probably worse, probably better, definitely better, excellent) for each method result in comparison with the original input image. The expressed preferences were averaged per each test image and then per each method to obtain the final MOS.

The final score of each submission was calculated as a weighted sum of the previous scores:

$$\begin{aligned}\textbf{Total Score} = \alpha \cdot (\text{PSNR}_{\text{solution}} - \text{PSNR}_{\text{baseline}}) &+ \\ \beta \cdot (\text{MS-SSIM}_{\text{solution}} - \text{MS-SSIM}_{\text{baseline}}) &+ \\ \gamma \cdot \min(4, \text{Time}_{\text{baseline}} / \text{Time}_{\text{solution}}) &.\end{aligned} \tag{1}$$

To cover a broader range of possible targets, we have additionally introduced three validation tracks with different weight coefficients: the first one (score A) was favoring solutions with high quantitative results, the second one (score B)—with high perceptual results, and the third one (score C) was aimed at the best

[1] https://www.mturk.com/.

balance between the speed, visual and quantitative scores. Below are the exact coefficients for all tracks:

Image Super-Resolution:

- $\text{PSNR}_{\text{baseline}} = 26.5$, $\text{SSIM}_{\text{baseline}} = 0.94$,
- (α, β, γ): score A - $(4, 100, 1)$, score B - $(1, 400, 1)$, score C - $(2, 200, 1.5)$.

Image Enhancement:

- $\text{PSNR}_{\text{baseline}} = 21.0$, $\text{SSIM}_{\text{baseline}} = 0.90$,
- (α, β, γ): score A - $(4, 100, 2)$, score B - $(1, 400, 2)$, score C - $(2, 200, 2.9)$.

The implementation of the scoring scripts, pre-trained baseline models and submission requirements are also available in the challenge github repository[2].

4 Results

During the validation phase, we have obtained more than 100 submissions from more than 20 different teams. 12 teams entered in the final test phase and submitted their models, codes and factsheets; Tables 1 and 2 summarize their results.

Table 1. Track A (Image super-resolution), final challenge results.

Team	PSNR	MS-SSIM	CPU, ms	GPU, ms	Razer Phone, ms	Huawei P20, ms	RAM	Score A	Score B	Score C
TEAM_ALEX	28.21	0.9636	701	48	936	1335	1.5 GB	**13.21**	**15.15**	**14.14**
KAIST-VICLAB	28.14	0.9630	**343**	**34**	**812**	**985**	1.5 GB	12.86	14.83	13.87
CARN_CVL	28.19	0.9633	773	112	1101	1537	1.5 GB	13.08	15.02	14.04
IV SR+	28.13	0.9636	767	70	1198	1776	1.6 GB	12.88	15.05	13.97
Rainbow	28.13	0.9632	654	56	1414	1749	1.5 GB	12.84	14.92	13.91
Mt.Phoenix	28.14	0.9630	793	90	1492	1994	1.5 GB	12.86	14.83	13.87
SuperSR	28.18	0.9629	969	98	1731	2408	1.5 GB	12.35	14.17	12.94
BOE-SBG	27.79	0.9602	1231	88	1773	2420	1.5 GB	9.79	11.98	10.55
SRCNN (Baseline)	27.21	0.9552	3239	205	7801	11566	2.6 GB	5.33	7.77	5.93

4.1 Image Super-Resolution

First of all, we would like to note that all submitted solutions demonstrated high efficiency: they were generally three to eight times faster than SRCNN, and at the same time were providing radically better visual and quantitative results. Another interesting aspect is that according to the results of the user study, its participants were not able to distinguish between the visual results produced by different solutions, and MOS scores in all cases except for the baseline SRCNN model were almost identical. The reason for this is that neither of the submitted models were trained with a strong adversarial loss component: they were mainly

[2] https://github.com/aiff22/ai-challenge.

optimizing Euclidean, MS-SSIM and VGG-based losses. In this track, however, we still have two winners: the first one is the solution proposed by TEAM_ALEX that achieved the best scores in all three validation tracks, while the second winning solution from KAIST-VICLAB has demonstrated the best runtime on all platforms, including two Android smartphones (Razer Phone and Huawei P20) on which it was able to process HD-resolution images under 1 s.

4.2 Image Enhancement

Similarly to the previous task, all submissions here were able to significantly improve the runtime and PSNR scores of the baseline SRCNN [8,13] and DPED [13] approaches. Regarding the perceptual quality, in this case there is no clear story, mainly high PSNR scores did not guarantee the best visual results, and vice versa. Also, MS-SSIM does not predict well the perceptual quality captured by MOS. The winner of this track is Mt.Phoenix team that achieved top MOS scores, as well as the best A, B and C scores and the fastest runtime on CPU and GPU. On smartphones, this solution required around 1.5 and 2 s for enhancing one HD-resolution photo on the Razer Phone and Huawei P20, respectively.

4.3 Discussion

The PIRM 2018 challenge on perceptual image enhancement on smartphones promotes the efficiency in terms of runtime and memory as a critical measure for successful deployment of solutions on real applications and mobile devices. For both considered tasks (super resolution and enhancement) a diversity of proposed solutions surpassed the provided baseline methods and demonstrated a greatly improved efficiency compared to many conventional techniques [15]. We conclude that the challenge through the proposed solutions define the state-of-the-art for image enhancement on smartphones.

5 Proposed Methods

This section describes solutions submitted by all teams participating in the final stage of the PIRM 2018 challenge on perceptual image enhancement on smartphones.

5.1 TEAM_ALEX

For track A, TEAM_ALEX proposed a residual neural network with 20 residual blocks [36], though all computations in this CNN were mainly done on the images downscaled by a factor of 4 with two desubpixel blocks; in the last two layers they were upscaled back to their original resolution with two subpixel modules. The main idea of desubpixel downsampling is shown on the Fig. 3—this is a reversible downsampling done via rearranging the spatial features into

Table 2. Track B (Image enhancement), final results. The results are sorted according to the MOS scores. CNN model from Rainbow team was using *tf.image.adjust_contrast* operation not yet available in TensorFlow Mobile and was not able to run on Android.

Team	PSNR	MS-SSIM	MOS	CPU, ms	GPU,	Razer Phone, ms	Huawei P20, ms	RAM	Score A	Score B	Score C
Mt.Phoenix	21.99	0.9125	**2.6804**	**682**	**64**	1472	2187	1.4 GB	**14.72**	**20.06**	**19.11**
EdS	21.65	0.9048	2.6523	3241	253	5153	Out of memory	2.3 GB	7.18	12.94	9.36
BOE-SBG	21.99	0.9079	2.6283	1620	111	1802	2321	1.6 GB	10.39	14.61	12.62
MENet	22.22	0.9086	2.6108	1461	138	2279	3459	1.8 GB	11.62	14.77	13.47
Rainbow	21.85	0.9067	2.5583	828	111	-	-	1.6 GB	13.19	16.31	16.93
KAIST-VICLAB	21.56	0.8948	2.5123	2153	181	3200	4701	2.3 GB	6.84	9.84	8.65
SNPR	22.03	0.9042	2.4650	1448	81	1987	3061	1.6 GB	9.86	10.43	11.05
DPED (Baseline)	21.38	0.9034	2.4411	20462	1517	37003	Out of memory	3.7 GB	2.89	4.90	3.32
Geometry	21.79	0.9068	2.4324	833	83	1209	1843	1.6 GB	12.0	12.59	14.95
IV SR+	21.60	0.8957	2.4309	1375	125	1812	2508	1.6 GB	8.13	9.26	10.05
SRCNN (Baseline)	21.31	0.8929	2.2950	3274	204	6890	11593	2.6 GB	3.22	2.29	3.49
TEAM_ALEX	21.87	0.9036	2.1196	781	70	962	1436	1.6 GB	10.21	3.82	10.81

several channels to reduce spatial dimensions without losing information. The whole network was trained with a combination of MSE and VGG-based loses on patches of size 196 × 196px (image super-resolution) and 100 × 100px (image enhancement) for 2×10^5 and 2×10^6 iterations, respectively. The authors used Adam optimizer with β_1 set to 0.9 and a batch size of 8; training data was additionally augmented with random flips and rotations. The learning rate was initialized at $1e-4$ and halved when the network was 60% trained.

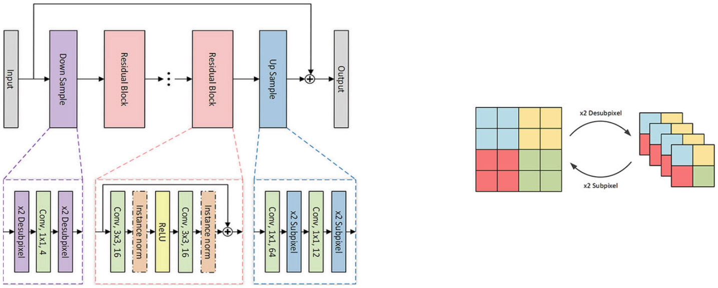

Fig. 3. Desubpixel block and the CNN architecture proposed by TEAM_ALEX.

5.2 KAIST-VICLAB

In track A, KAIST-VICLAB proposed a similar approach of using 4× image downscaling and residual learning, however their CNN (Fig. 4) consisted of only 8 convolutional layers. High visual and quantitative results were still obtained by using a slightly different training scheme: the authors applied a small amount of Gaussian blur to degrade the downscaled low-resolution training patches, while they improved construct and sharpness of the target high-resolution images. Furthermore, residual units, pixel shuffle [27], error feedback scheme [9] and xUnit [17] were integrated into network for faster learning and higher performance. The authors used 2,800 additional images from the BSDS300, Flickr500 and Flickr2K datasets for training, and augmented data with random flips and rotations. The network was trained for 2000 epochs on 128 × 128px patches with L1 loss only; the batch size was set to 4, the learning rate was $1e-4$.

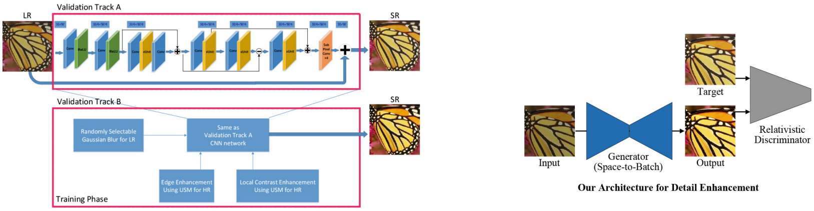
Validation Track A
LR
SR
H×W
H/4×W/4
Conv
lReLU
Conv
xUnit
Sub Pixel Conv ×4
Validation Track B
SR
Randomly Selectable Gaussian Blur for LR
Same as Validation Track A CNN network
Edge Enhancement Using USM for HR
Local Contrast Enhancement Using USM for HR
Training Phase
Target
Input
Generator (Space-to-Batch)
Output
Relativistic Discriminator
Our Architecture for Detail Enhancement

Fig. 4. Solutions proposed by KAIST-VICLAB for tracks A (left) and B (right).

For track B, KAIST-VICLAB presented an encode-decoder based architecture (Fig. 4), where spatial sizes are reduced with a space-to-batch technique: instead of using stride-2 convolutions, the feature maps obtained after each layer are divided into 4 smaller feature maps that are then concatenated along the batch dimension. The authors used an additional adversarial component, and for the discriminator they proposed relativistic RGAN [16] with twice as many parameters as in the generator. The network was trained similarly to track A, but with a combination of color and adversarial losses defined in [13].

5.3 Mt.Phoenix

For image super-resolution, the Mt.Phoenix authors used a deep residual CNN with two downsampling blocks performing image downscaling and two deconvolution blocks for its upscaling to the original size. Besides the standard residual blocks, additional skip connections between the input and middle layers were added to improve the performance of the network. CNN was trained on 500 × 500px patches using Adam optimizer with an initial learning rate of $5e-4$ and a decay of $5e-5$. The network was trained with L1 loss, no data augmentation was used.

In the second track, Mt.Phoenix proposed a U-net style architecture [25] (Fig. 5) and augmented it with global features calculated by applying average pooling to features from its bottleneck layer. Additionally, a global transform layer performing element-wise multiplication of the outputs from the second and last convolutional layers was proposed. The network was trained with a combination of L1, MS-SSIM, VGG, total variation and GAN losses using Adam optimizer with a constant learning rate of $5e-4$.

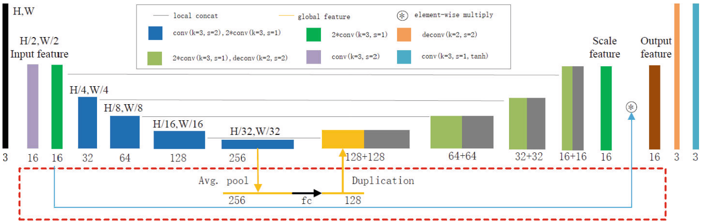

Fig. 5. U-net architecture for image enhancement proposed by Mt.Phoenix.

5.4 CARN_CVL

For image super-resolution, CARN_CVL proposed the convolutional anchored regression network (CARN) [22] (see Fig. 6) which has the capability to efficiently trade-off between speed and accuracy. Inspired by A+ [33,34] and ARN [2], CARN is formulated as a regression problem. The features are extracted from input raw images by convolutional layers. The regressors map features from low dimension to high dimension. Every regressor is uniquely associated with an anchor point so that by taking into account the similarity between the anchors and the extracted features, CARN can assemble the different regression results to form output features or the original image. In order to overcome the limitations of patch-based SR, all of the regressions and similarity comparisons between anchors and features are implemented by convolutional layers and encapsulated by a regression block. Furthermore, by stacking the regression block, the performance of the network increases steadily. CARN_CVL starts with the basic assumption of locally linear regression, derives the insights from it, and points out how to convert the architecture to convolutional layers in the proposed CARN.

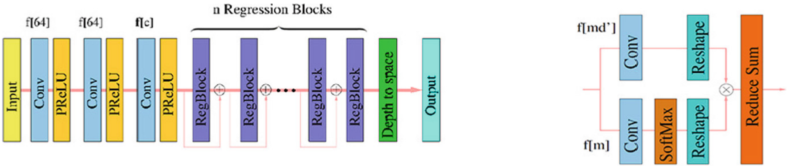

Fig. 6. CARN architecture and CARN Regression Block presented by CARN_CVL.

The challenge entry uses CARN with 5 regression blocks, 16 anchors/regressors per block, and a number of feature layers reduced to 2. In the two feature layers, the stride of the convolution operation is set to 2 because the bicubic interpolated image contains no high frequency information compared to the LR image but slows down the executation of the network. The number of inner channels is set as 8 for the upscaling factor 4.

5.5 EdS

EdS proposed a modification [30] of the original DPED ResNet architecture used for image enhancement (Fig. 7). The main difference in their network was the use of two 4 × 4 convolutional layers with stride 2 for going into lower dimensional space, and additional skip connections for faster training. The network was trained for 33K iterations using the same losses and setup as in [13].

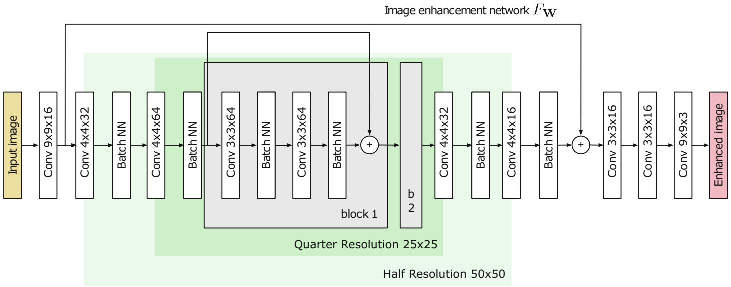

Fig. 7. A variation of the original DPED architecture proposed by EdS team.

5.6 IV SR+

The authors proposed a Fast Clique Convolutional Network (FCCN), which architecture was inspired by CliuqueNet [39] and MobileNet [10]. The proposed FCCN consists of feature extraction, fast clique block (FCB) and two deconvolution layers (Fig. 8). For feature extraction, two convolutional layers with 32 and 20 kernels are utilized. Then, to accelerate the FCCN architecture, these features are fed to FCB layers for extracting more informative convolutional features. The FCB layer consists of one input convolutional layer and four bidirectional densely connected convolutional layers with both depthwise and pointwise convolution. The network was trained using Adam optimizer and a batch size of 16 for 3M iterations with an initial learning rate of $1e-4$ halved after 2 million iterations.

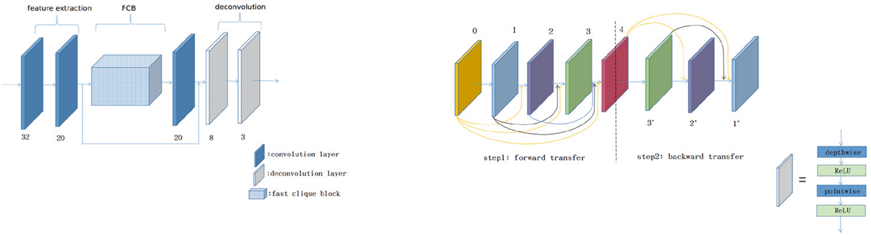

Fig. 8. FCCN and the corresponding Fast Clique Block (FCB) proposed by IV SR+.

5.7 BOE-SBG

The architecture of the network used for image super-resolution is presented in the Fig. 9 and is based on the Laplacian pyramid framework with a dense-block inspired by [18]. The parameters of denseblocks, strided and transposed convolutional layers are shared among different network levels to improve the performance. For image enhancement problem, the authors proposed a different architecture [23] (Fig. 9). First of all, it featured several Mux and Demux layers performing image up- and downscaling without information loss and that are basically a variant of (de)subpixel layers used in other approaches. This network was additionally trained with an extensive combination of various losses, including L1 loss for each image color channel, contextual, VGG, color, total variation and adversarial losses.

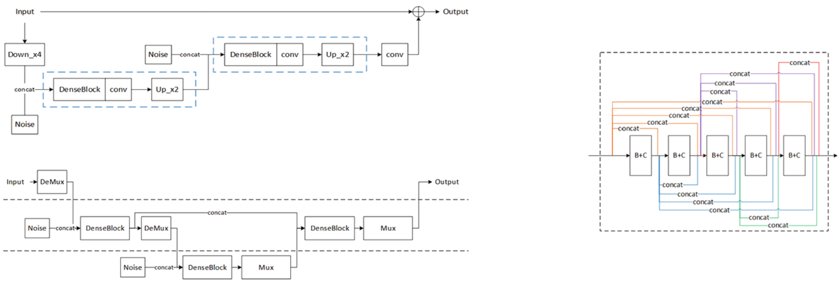

Fig. 9. Neural networks for image super-resolution (top), image enhancement (bottom) and the corresponding Denseblock (right) proposed by BOE-SBG team.

5.8 Rainbow

The CNN architecture used in the first track is shown in the Fig. 10. The network consists of two convolutional layers with stride 2, three convolutional layers with stride 1, cascaded residual blocks and a subpixel layer. The network was trained to minimize L1 and SSIM losses on 384 × 384px patches augmented with random flips and rotations. The learning rate was set to $5e-4$ and decreased by a factor of 5 every 1000 epochs.

A different approach [12] was used for image enhancement: the authors first trained a larger teacher generator and then used it to guide the training of the smaller student network (see Fig. 10). The latter was done by imposing additional knowledge distillation loss calculated as Euclidian distance between the corresponding normalized student's and teacher's feature maps. Besides this loss, the networks were trained with a combination of SSIM, VGG, L1, context, color and total variation losses using Adam optimizer with an initial learning rate of $5e-4$ decreased by a factor 10 for every 10^4 iterations.

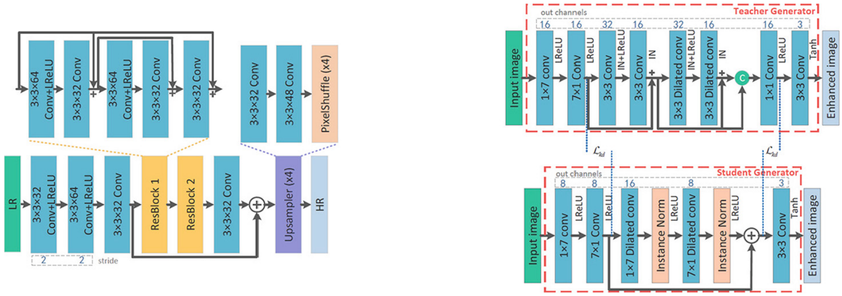

Fig. 10. CNN architectures proposed by Rainbow for tracks A (left) and B (right).

5.9 MENet

MENet team proposed a θ-inception Network depicted in the Fig. 11 for image enhancement problem. This CNN has a θ-inception block where the image is processed in parallel by convolutional and deconvolutional layers with strides 2 and 4 for multi-scale learning. Besides that, the size of the convolutional filters is different too: 3 and 5 in the first and the second case, respectively. At the end of this block, the corresponding two outputs are concatenated together with the output from the first convolutional layer and are passed to the last CNN layer. The network is trained using the same setup as in [13] with the following two differences: (1) two additional texture loss functions (local contrast normalization and gradient) are used and (2) after pre-training the network is additionally fine-tuned on the same dataset with Adam minimizer and a learning rate of $1e-4$.

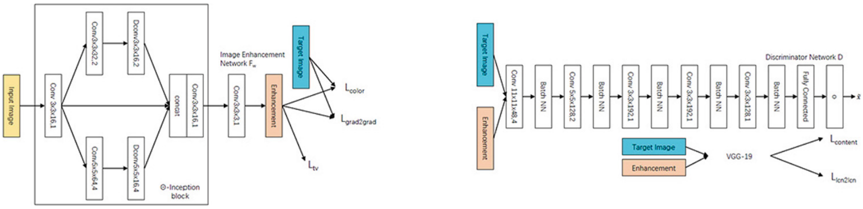

Fig. 11. θ-inception Network (generator and discriminator) presented by MENet team.

5.10 SuperSR

Figure 12 presents the CNN architecture used for image super-resolution problem. The network consists of one space-to-depth 4× downsampling layer followed by convolutional and residual layers with PReLU activation functions and one deconvolutional layer for image upscaling. The model was trained on 192×192px patches augmented with flips and rotations. Adam optimizer with a mini-batch size of 32 and a learning rate of $1e-3$ decayed by 10 every 1000 epochs was used for CNN training. After the initial pre-training with L2 loss, the training process was restarted with the same settings, while the loss function was replaced by a mixture of Charbonnier [5] loss and MS-SSIM losses.

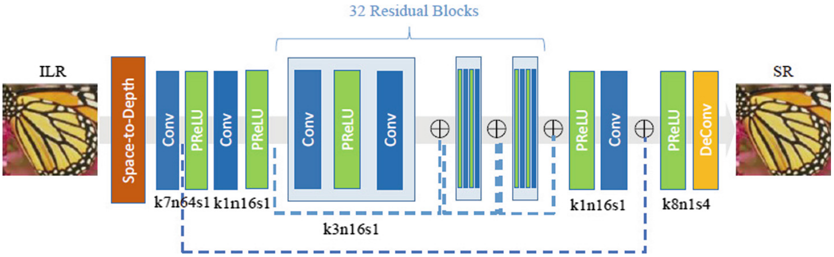

Fig. 12. Deep residual network proposed by SuperSR team.

5.11 SNPR

For image enhancement, SNPR derives three network architectures corresponding to different operating points. The generator networks ($G1$, $G2$, and $G3$) corresponding to the three different approaches and the common discriminator network D are shown in Fig. 13. *Conv(f, k, s)* refers to a convolution layer with f $k \times k$ filters performing convolution by a stride factor of s, ReLU is a Rectified Linear Unit, BN refers to batch-normalization, and *Pixel-Shuffler X2* refers to the pixel shuffler layer [27] which increases resolution by a factor of 2. The first three layers are meant to extract the features that are relevant for image enhancement. Feature extraction at low-image-dimension has the advantages of larger receptive field and much lower computational complexity [29]. To compensate for detrimental effects of spatial dimension reduction in features, the input image (which have full-resolution spatial features) is concatenated with the features extracted at low-dimensional space and then combined by the succeeding convolutional layers. Overall $G3$ achieves the best speed-up-ratio but with a lower performance as compared to DPED baseline [13], whereas $G1$ achieves the lowest speed-up-ratio while having comparable quality to that of DPED.

5.12 Geometry

The overall structure of the network [24] presented by Geometry team is shown in the Fig. 13. Each convolutional layer has 16 filters, and the network itself produces two outputs: one based on the features from the middle CNN layer, and one from the last layer. The intermediate output (Output OC) is used to compute SSIM loss, while the final one (Output OE) is used to compute the loss function consisting of adversarial, smooth, and style losses. During the training all losses are summed, and the network is trained as a whole using Adam optimizer with a learning rate of $5e-4$ decreased by a factor of 10 every 8000 iterations.

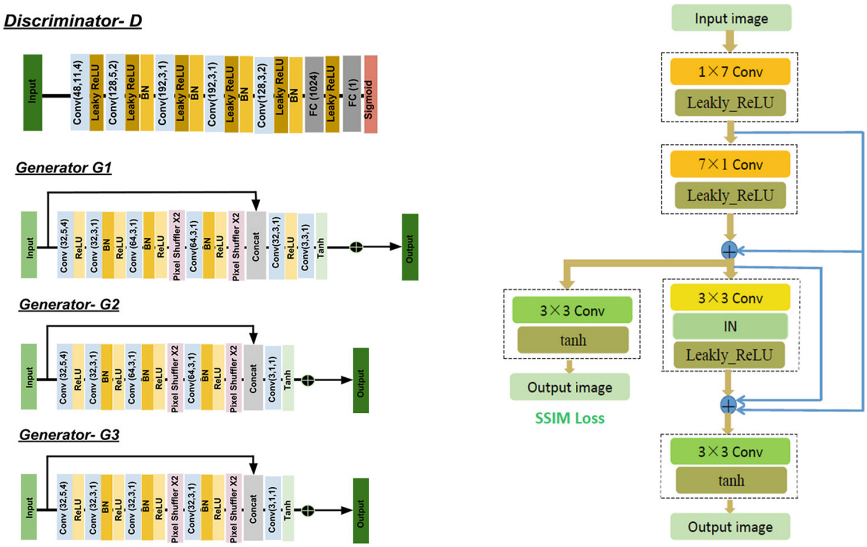

Fig. 13. Neural networks proposed by SNPR (left) and Geometry (right) teams.

Acknowledgements. We thank the PIRM2018 sponsors: ETH Zurich (Computer Vision Lab), Huawei Inc., MediaTek Inc., and Israel Institute of Technology.

Appendix 1: Teams and affiliations

PIRM 2018 Team

Title: PIRM Challenge on Perceptual Image Enhancement on Smartphones
Members: Andrey Ignatov – andrey@vision.ee.ethz.ch, Radu Timofte – radu.timofte@vision.ee.ethz.ch
Affiliations: Computer Vision Lab, ETH Zurich, Switzerland

TEAM_ALEX

Title: Fast and Efficient Image Quality Enhancement using Desubpixel Downsampling [36]
Members: Thang Vu – thangvubk@kaist.ac.kr, Tung Luu, Trung Pham, Cao Nguyen
Affiliations: Dept. of Electrical Engineering, KAIST, Republic of Korea

KAIST-VICLAB

Title-A: A Low-Complexity Convolutional Neural Network for Perceptual Super-Resolution using Randomly-Selected Degraded LR and Enhanced HR
Title-B: A Convolutional Neural Network for Detail Enhancement with the Relativistic Discriminator
Members: Yongwoo Kim – yongwoo.kim@kaist.ac.kr, Jae-Seok Choi, Munchurl Kim
Affiliations: Video and Image Computing Lab, KAIST, Republic of Korea

Mt.Phoenix

Title-A: Multi Level Super Resolution Net [25]
Title-B: Range Scaling Global U-Net for Perceptual Image Enhancement on Mobile Devices
Members: Pengfei Zhu – zpf2@meitu.com, Chen Xing, Xingguang Zhou, Jie Huang, Mingrui Geng, Jiewen Ran
Affiliations: Meitu Imaging & Vision Lab, China

CARN_CVL

Title: Convolutional Anchored Regression Network [22]
Members: Yawei Li – yawei.li@vision.ee.ethz.ch, Eirikur Agustsson, Shuhang Gu, Radu Timofte, Luc Van Gool
Affiliations: Computer Vision Lab, ETH Zurich, Switzerland

IV SR+

Title: An Efficient and Compact Mobile Image Super-resolution with Fast Clique Convolutional Network
Members: Kehui Nie[1] – n161120080@fzu.edu.cn, Yan Zhao[1], Gen Li[2], Tong Tong[2], Qinquan Gao[1]
Affiliations: [1] – Fuzhou University, China
[2] – Imperial Vision, China

Rainbow

Title: Perception-Preserving Convolutional Networks for Image Enhancement [12] on Smartphones
Members: Zheng Hui[1] – zheng_hui@aliyun.com, Xiumei Wang[1], Lirui Deng[2], Rang Meng[3]
Affiliations: [1] – Xidian University, China
[2] – Tsinghua University, China
[3] – Zhejiang University, China

SuperSR

Title: Enhanced FSRCNN for Image Super-Resolution
Members: Ruicheng Feng – jnjaby@gmail.com, Shixiang Wu, Chao Dong, Yu Qiao
Affiliations: Shenzhen Institute of Advanced Technology, China

BOE-SBG

Title-A: Deep Laplacian Pyramid Networks with Denseblock for Image Super-Resolution
Title-B: Deep Networks for Image-to-image Translation with Mux and Demux Layers [23]
Members: Liu Hanwen – liuhanwen@boe.com.cn, Pablo Navarrete Michelini, Zhu Dan, Hu Fengshuo
Affiliations: BOE Technology Group Co., Ltd, China

EdS

Title: Fast Perceptual Image Enhancement [30]
Members: Etienne de Stoutz – etienned@ethz.ch Nikolay Kobyshev
Affiliations: ETH Zurich, Switzerland

MENet

Title: Fast and Accurate DSLR-Quality Photo Enhancement Using θ-inception Network
Members: Jinghui Qin – qinjingh@mail2.sysu.edu.cn, Yukai Shi, Wushao Wen, Liang Lin
Affiliations: Sun Yat-sen University, China

SNPR

Title: Efficient Perceptual Image Enhancement Network for Smartphones
Members: Subeesh Vasu – subeeshvasu@gmail.com, Nimisha Thekke Madam, Praveen Kandula, A. N. Rajagopalan
Affiliations: Indian Institute of Technology Madras, India

Geometry

Title: Multiple Connected Residual Network for Image Enhancement on Smartphones [24]
Members: Jie Liu – jieliu543@gmail.com, Cheolkon Jung
Affiliations: School of Electronic Engineering, Xidian University, China

References

1. Agustsson, E., Timofte, R.: NTIRE 2017 challenge on single image super-resolution: dataset and study. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, vol. 3, p. 2 (2017)
2. Agustsson, E., Timofte, R., Van Gool, L.: Anchored regression networks applied to age estimation and super resolution. In: The IEEE International Conference on Computer Vision (ICCV), October 2017
3. Ancuti, C., Ancuti, C.O., Timofte, R.: NTIRE 2018 challenge on image dehazing: methods and results. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2018
4. Arad, B., Ben-Shahar, O., Timofte, R.: NTIRE 2018 challenge on spectral reconstruction from RGB images. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2018
5. Barron, J.T.: A more general robust loss function. arXiv preprint arXiv:1701.03077 (2017)
6. Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., Zelnik-Manor, L.: 2018 PIRM challenge on perceptual image super-resolution. In: European Conference on Computer Vision Workshops (2018)
7. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3223 (2016)
8. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 295–307 (2016)
9. Haris, M., Shakhnarovich, G., Ukita, N.: Deep backprojection networks for super-resolution. In: Conference on Computer Vision and Pattern Recognition (2018)
10. Howard, A.G., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
11. Huang, J., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: IEEE CVPR, vol. 4 (2017)
12. Hui, Z., Wang, X., Deng, L., Gao, X.: Perception-preserving convolutional networks for image enhancement on smartphones. In: European Conference on Computer Vision Workshops (2018)
13. Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., Van Gool, L.: DSLR-quality photos on mobile devices with deep convolutional networks. In: The IEEE International Conference on Computer Vision (ICCV) (2017)
14. Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., Van Gool, L.: WESPE: weakly supervised photo enhancer for digital cameras. arXiv preprint arXiv:1709.01118 (2017)
15. Ignatov, A., et al.: AI benchmark: Running deep neural networks on android smartphones. In: European Conference on Computer Vision Workshops (2018)
16. Jolicoeur-Martineau, A.: The relativistic discriminator: a key element missing from standard GAN. arXiv preprint arXiv:1807.00734 (2018)
17. Kligvasser, I., Shaham, T.R., Michaeli, T.: xUnit: learning a spatial activation function for efficient image restoration. arXiv preprint arXiv:1711.06445 (2017)
18. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep Laplacian pyramid networks for fast and accurate superresolution. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, p. 5 (2017)
19. Ledig, C., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR, vol. 2, p. 4 (2017)

20. Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G.: A convolutional neural network cascade for face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5325–5334 (2015)
21. Li, L.J., Socher, R., Fei-Fei, L.: Towards total scene understanding: classification, annotation and segmentation in an automatic framework. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 2036–2043. IEEE (2009)
22. Li, Y., Eirikur Agustsson, E., Gu, S., Timofte, R., Van Gool, L.: CARN: convolutional anchored regression network for fast and accurate single image super-resolution. In: European Conference on Computer Vision Workshops (2018)
23. Liu, H., Navarrete Michelini, P., Zhu, D.: Deep networks for image to image translation with Mux and Demux layers. In: European Conference on Computer Vision Workshops (2018)
24. Liu, J., Jung, C.: Multiple connected residual network for image enhancement on smartphones. In: European Conference on Computer Vision Workshops (2018)
25. Pengfei, Z., et al.: Range scaling global u-net for perceptual image enhancement on mobile devices. In: European Conference on Computer Vision Workshops (2018)
26. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
27. Shi, W., et al.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1874–1883 (2016)
28. Shoeiby, M., Robles-Kelly, A., Timofte, R., et al.: PIRM 2018 challenge on spectral image super-resolution: methods and results. In: European Conference on Computer Vision Workshops (2018)
29. Sim, H., Ki, S., Choi, J.S., Seo, S., Kim, S., Kim, M.: High-resolution image dehazing with respect to training losses and receptive field sizes. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2018
30. de Stoutz, E., Ignatov, A., Kobyshev, N., Timofte, R., Van Gool, L.: Fast perceptual image enhancement. In: European Conference on Computer Vision Workshops (2018)
31. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
32. Timofte, R., et al.: NTIRE 2017 challenge on single image super-resolution: methods and results. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1110–1121, July 2017. https://doi.org/10.1109/CVPRW.2017.149
33. Timofte, R., De Smet, V., Van Gool, L.: Anchored neighborhood regression for fast example-based super-resolution. In: The IEEE International Conference on Computer Vision (ICCV), December 2013
34. Timofte, R., De Smet, V., Van Gool, L.: A+: adjusted anchored neighborhood regression for fast super-resolution. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9006, pp. 111–126. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16817-3_8
35. Timofte, R., Gu, S., Wu, J., Van Gool, L.: NTIRE 2018 challenge on single image super-resolution: methods and results. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2018

36. Van Vu, T., Van Nguyen, C., Pham, T.X., Liu, T.M., Youu, C.D.: Fast and efficient image quality enhancement via desubpixel convolutional neural networks. In: European Conference on Computer Vision Workshops (2018)
37. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003, vol. 2, pp. 1398–1402, November 2003. https://doi.org/10.1109/ACSSC.2003.1292216
38. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
39. Yang, Y., Zhong, Z., Shen, T., Lin, Z.: Convolutional neural networks with alternately updated clique. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2413–2422 (2018)