PIRM Challenge on Perceptual Image Enhancement on Smartphones: Report

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

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 downsampling 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. 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.

1 Introduction

The majority of the current challenges related to AI and deep learning for image restoration and enhancement [32, 35, 3, 4, 6, 28] 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 [31, 10], image enhancement [13, 14], image super-resolution [8, 34], object tracking [38, 11], visual scene understanding [21, 7], face detection and recognition [20, 26], etc.

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 [15]. It considers two classical computer vision problems — image super-resolution and enhancement, and introduces 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.

* 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
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, some sample results for which obtained with SRGAN network [19] are shown in the figure 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 figure 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:

$$\text{Total Score} = \alpha \cdot (\text{PSNR}_{\text{solution}} - \text{PSNR}_{\text{baseline}}) + \beta \cdot (\text{MS-SSIM}_{\text{solution}} - \text{MS-SSIM}_{\text{baseline}}) + \gamma \cdot \min(4, \frac{\text{Time}_{\text{baseline}}}{\text{Time}_{\text{solution}}})$$

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

**Image super-resolution:**
- $\text{PSNR}_{\text{baseline}} = 26.5, \text{SSIM}_{\text{baseline}} = 0.94,$
- $(\alpha, \beta, \gamma)$: score A - (4, 100, 1), score B - (1, 400, 1), score C - (2, 200, 1.5).

**Image enhancement:**
- $\text{PSNR}_{\text{baseline}} = 21.0, \text{SSIM}_{\text{baseline}} = 0.90,$
- $(\alpha, \beta, \gamma)$: 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.

1https://www.mturk.com/

2https://github.com/aiff22/ai-challenge
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           | 1.5GB| 13.21   | 15.15   | 14.14   |
| KAIST-VICLAB  | 28.14 | 0.9630  | 343     | 34      | 812            | 985            | 1.5GB| 12.86   | 14.83   | 13.87   |
| CARN_CVL     | 28.19 | 0.9633  | 773     | 112     | 1101           | 1537           | 1.5GB| 13.08   | 14.97   | 13.91   |
| IV SR+        | 28.13 | 0.9636  | 767     | 70      | 1198           | 1776           | 1.6GB| 12.84   | 14.92   | 13.97   |
| Rainflow      | 28.19 | 0.9633  | 773     | 112     | 1101           | 1537           | 1.5GB| 13.08   | 14.97   | 13.91   |
| SuperSR       | 28.13 | 0.9632  | 654     | 56      | 1414           | 1749           | 1.5GB| 12.84   | 14.92   | 13.97   |
| BOE-SBG       | 27.79 | 0.9602  | 1231    | 88      | 1773           | 2420           | 1.5GB| 9.79    | 11.98   | 10.55   |
| SRCNN (Baseline) | 27.21 | 0.9552  | 3239    | 205     | 7801           | 11566          | 2.6GB| 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 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 second.

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 seconds 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 greatly improved the efficiency. We conclude that the challenge through the proposed solutions define the state-of-the-art for image enhancement on smartphones.

5 Proposed methods

5.1 TEAM_ALEX

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

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 figure 3 — this is a reversible downsampling done via rearranging the spatial features into several channels to reduce spatial dimensions without losing information. The whole network was trained with a combination of MSE and VGG-based losses on patches of size \(196 \times 196\)px (image super-resolution) and \(100 \times 100\)px (image enhancement) for \(2 \times 10^5\) and \(2 \times 10^6\) iterations, respectively. The authors used Adam optimizer with \(\beta_1\) 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 percent trained.

5.2 KAIST-VICLAB

In track A, KAIST-VICLAB proposed a similar approach of using \(4 \times\) image downsampling and residual learning, however...
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, ms | Razer Phone, ms | Huawei P20, ms | RAM | Score A | Score B | Score C |
|---------------|------|---------|------|---------|---------|-----------------|----------------|-----|---------|---------|---------|
| Mt.Phoenix    | 21.99| 0.9125  | 2.6884| 682     | 64      | 1472            | 2187           | 1.4GB| 14.72   | 20.06   | 19.11   |
| EUS           | 21.65| 0.9048  | 2.6753| 3241    | 253     | 5153            | (out of memory) | 2.3GB| 7.18    | 12.94   | 9.36    |
| BOE-SBG       | 21.99| 0.9079  | 2.6283| 1620    | 111     | 1802            | 2321           | 1.6GB| 10.39   | 14.61   | 12.62   |
| MEnet         | 22.22| 0.9086  | 2.6108| 1461    | 138     | 2279            | 3459           | 1.8GB| 11.62   | 14.77   | 13.47   |
| Rainbow       | 21.85| 0.9067  | 2.5583| 828     | 111     | -               | -              | 2.3GB| 7.18    | 12.94   | 9.36    |
| KAIST-VICLAB  | 21.56| 0.8948  | 2.5123| 2153    | 181     | 3200            | 4701           | 2.3GB| 6.84    | 9.84    | 8.65    |
| SNPR          | 22.03| 0.9042  | 2.4650| 1448    | 81      | 1987            | 3061           | 1.6GB| 9.86    | 10.43   | 11.05   |
| DPED (Baseline)| 21.38| 0.9034  | 2.4411| 20462   | 1517    | 37903           | (out of memory) | 3.7GB| 2.89    | 4.90    | 3.32    |
| Geometry      | 21.79| 0.9068  | 2.4324| 833     | 83      | 1209            | 1843           | 1.6GB| 12.0    | 12.59   | 14.95   |
| IV SR+        | 21.60| 0.8957  | 2.4309| 1375    | 125     | 1812            | 2508           | 1.6GB| 8.13    | 9.26    | 10.05   |
| SRCNN (Baseline)| 21.31| 0.8929  | 2.2950| 3274    | 204     | 6890            | 11593          | 2.6GB| 3.22    | 2.29    | 3.49    |
| TEAM_ALEX     | 21.87| 0.9036  | 2.1196| 781     | 70      | 962             | 1436           | 1.6GB| 10.21   | 3.82    | 10.81   |

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 upsampling 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 $5e^{-4}$.

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+ [34, 33] and ARN [2], CARN is formulated as a regression problem. The features are extracted from input raw images by convolutional...
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 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.

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 execution 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 $4 \times 4$ 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].

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.

5.7 BOE-SBG

The architecture of the network used for image super-resolution is presented in the figure 9 and is based on the Laplacian pyramid framework with a denseblock 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.

5.8 Rainbow

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

Figure 10: CNN architectures proposed by Rainbow for tracks A (left) and B (right).
The CNN architecture used in the first track is shown in the figure 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 \(5 \times 10^{-4}\) 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 \(5 \times 10^{-4}\) decreased by a factor 10 for every \(10^4\) iterations.

5.9 MENet

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

MENet team proposed a θ-inception Network depicted in the figure 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 optimizer and a learning rate of \(1 \times 10^{-4}\).

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 \(1 \times 10^{-3}\) 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.

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 \(f \times k \times k\) 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 figure 14. 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 $5 \times 10^{-4}$ decreased by a factor of 10 every 8000 iterations.

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 Down-sampling [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 on Smartphones [12]
Members: Zheng Hui 1 – zheng_hui@aliyun.com, Xiumei Wang 3, 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  
**Members:** Liu Hanwen – liuhanwen@boe.com.cn, Pablo Navarrette 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 $\theta$-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

[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) (Oct 2017) 4

[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) 1

[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) 1

[5] Barron, J.T.: A more general robust loss function. arXiv preprint arXiv:1701.03077 (2017) 6

[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) 1

[7] Cords, M., Omran, M., Ramos, S., Rhelfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016) 1

[8] Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence 38(2), 295–307 (2016) 1, 2, 3

[9] Haris, M., Shakhnarovich, G., Ukit, N.: Deep backprojection networks for super-resolution. In: Conference on Computer Vision and Pattern Recognition (2018) 4

[10] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017) 1, 5

[11] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: IEEE CVPR. vol. 4 (2017) 1

[12] Hui, Z., Wang, X., Deng, L., Gan, X.: Perception-preserving convolutional networks for image enhancement on smartphones. In: European Conference on Computer Vision Workshops (2018) 6, 7

[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 Int. Conf. on Computer Vision (ICCV) (2017) 1, 2, 3, 4, 5, 6

[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) 1

[15] Ignatov, A., Timofte, R., Szczepaniak, P., Chou, W., Wang, K., Wu, M., Hartley, T., Van Gool, L.: Ai benchmark: Running deep neural networks on android smartphones. In: European Conference on Computer Vision Workshops (2018) 1

[16] Jolicoeur-Martineau, A.: The relativistic discriminator: a key element missing from standard gan. arXiv preprint arXiv:1807.00734 (2018) 4

[17] Kligvasser, I., Shaham, T.R., Michaeli, T.: xunit: Learning a spatial activation function for efficient image restoration. arXiv preprint arXiv:1711.06445 (2017) 4

[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) 5

[19] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. vol. 2, p. 4 (2017) 2
