Deep Networks for Image-to-Image Translation with Mux and Demux Layers

Hanwen Liu[1], Pablo Navarrete Michelini[1], and Dan Zhu[1]

BOE Technology Group Co., LTD., No.9 Dize Road, BDA, Beijing, 100176, P.R.China

Abstract. Image processing methods using deep convolutional networks have achieved great successes on quantitative and qualitative assessments in many tasks, such as super–resolution, style transfer and and enhancement. Most of these solutions use many layers, many filters and complex architectures. It is difficult to implement them on mobile devices, e.g. smart phones, because of the limited resources. Many applications need to deploy these methods on mobile devices. But it is difficult because of limited resources. In this paper we present a lightweight end–to–end deep learning approach for image enhancement. To improve the performance, we present mux layer and demux layers, which could perform up–sampling and down–sampling by shuffling the pixels without losing any information of feature maps. For further higher performance, dense-blocks are used in the models. To ensure the consistency of the output and input, we use weighted L1 loss to increase PSNR. To improve image quality, we use adversarial loss, contextual loss and perceptual loss as parts of the objective functions during training. And NIQE is used for validation to get the best parameters for perceptual quality. Experiments show that, compared to the state–of–the–art, our method could improve both the quantitative and qualitative assessments, as well as the performance. With this system, we get the third place in PIRM Enhancement–On–Smartphones Challenge 2018(PIRM–EoS Challenge 2018).

Keywords: Mux layer · Demux layer · Image enhancement · Deep learning

1 Introduction

In recent years, embedded cameras in mobile devices have been improved rapidly, which has brought mobile photographs to a substantially new level. However, because of some limits, such as small size, compact lenses and the lack of specific hardware, the quality of mobile photographs is still falling behind DSLR cameras.

Because of high–aperture optics and larger sensors, DSLR cameras could capture photographs with higher quality, color rendition and less noise. These physical differences between DSLR cameras and mobile devices lead to a great gap, making DSLR cameras quality unattainable for compact mobile devices.

There have been many methods that could enhance the mobile photographs, but most of them could only adjust global parameters, such brightness or contrast. They are usually based on some pre-defined rules to adjust these parameters. However, image quality is very related to textures and image semantics, which are difficult for classic method to improve.

Mobile devices face two big difficulties to take photographs with similar quality of DSLR cameras. One is to find algorithms that could improve photographs with not only global parameters, but also semantic and perceptual qualities. The second problem is to implement these algorithms on mobile devices, which means that these methods should be lightweight.

First, image processing methods based on deep convolutional networks usually could achieve these targets to improve the quality of images. Several solutions for different sub–tasks have solved the first problem about image quality. These methods solve image–to–image translation, targeting at translating images from one domain to another. To ensure the high perceptual quality of outputs, many of them use some particular metrics to measure image quality and put them as part of the objective functions. The sub–tasks include image super resolution, image deblurring, image dehazing and denoising. However, methods based deep learning usually cost large number of resources, such as CPU, GPU and memory, which makes it difficult to implement on mobile devices.

Second, to solve the implementation problem, recent deep learning architectures have been used. This includes: MobileNet[15], ShuffleNet[16], MeNet[31] and DPED[3]. All these architectures target devices with limited resources.

The remainder of this paper is structured as follows. In Section 2 we introduce some related works to our research. Section 3 explains the main contributions of this paper. Section 4 presents our method in detail, include the architecture and loss functions. Section 5 shows experiment results and analysis. Finally, Section 6 concludes this paper.

2 Related Work

Image super resolution aims at restoring an original image form its downscale version. In [2], they used a CNN and MSE loss to learn how to map low resolution images to high resolution. This is the first deep learning solutions for single image super resolution. Later work proposed deeper and more complex architectures, such as[4], [5], [6]. Recently, photo–realistic results with high perceptual quality have been possible to achieve by using a pre–trained VGG network for loss function[1] and adversarial networks[7]. They are known to be efficient at recovering plausible high–frequency components, that look more realistic at the cost of losing distortion values [29].

Image deblurring and dehazing aim at removing artificially added haze or blur from the images. Usually, MSE loss is used as a loss function and the proposed CNN architectures consist of three to fifteen convolutional layers[8] [9][10], or are bi–channel CNNs[11].

Image denoising similarly targets removal of noise and artifacts from the images. In[12] the authors presented weighted MSE together with a three–layer CNN, while in[13] it was shown that an eight–layer residual CNN performs better when using a standard MSE.

Image enhancement DPED network[3] presented a novel approach for the photo enhancement task based on learning a mapping between photos from mobile devices and DSLR camera. The model is trained in an end–to–end fashion without any additional supervision or manually adjusted features. Authors of DPED used a multi–term loss function composed of color, texture and content terms, allowing an efficient image quality estimation.

Improving performance of deep learning models is always an important direction in the field of deep learning. Many compact networks are designed for mobile or embedded applications. SqueezeNet[14] proposed fire modules, where 1×1 convolutional layer is first applied to squeeze the width of the network, followed by a layer mixing 3×3 and 1×1 convolutional kernels to reduce parameter. MobileNet[15] exploited depthwise separable convolutions as its building unit, which decompose a standard convolution into a combination of a depthwise convolution and a pointwise convolution. ShuffleNet[16] used depthwise convolutions and pointwise group convolutions into the bottleneck unit[17], and proposed the channel shuffle operation to enable inter–group information exchange. These networks do not use model compression techniques and so they can be trained without using large models and the training procedure is very fast.

Improving image quality Contextual loss[19] was proposed to measure the similarity between the feature distributions of two images. Contextual loss identifies similar patches between two images, making it better as a perceptual quality target. Another metric to measure perceptual image quality is the Natural Image Quality Evaluator(NIQE) [21]. NIQE is a completely blind image quality index based on a collection of statistical features that are known to follow a multivariate Gaussian for natural images. It can thus quantify how natural or real image looks without any reference image, providing a perceptual quality index similar to a human evaluations such as MOS.

3 Contributions

To solve the problems mentioned before, there are two research targets. One is to present a novel model to keep high performance, and the other is to propose methods to ensure the high quality of processed images.

To achieve these targets, we make the following contributions:

1. Novel layers with shuffling pixels for up–sampling and down–sampling, which we call Mux and Demux layers, respectively. Mux layers divide input features into groups, with each group consisting on four input features. By rearranging pixels of every group of four features, the output of the Mux layer doubles the width and height. Thus, it can be used as up–sampling layer in CNN. Demux layer converts input features into 4 times the number of features with half the

width and half the height. So Demux layer can be used as down–sampling layer. Because of the shuffling pixels, the outputs keep all the informations from the inputs, as opposed to to standard pooling and unpooling layers.

2. For high performance, we present new CNN architectures. In the new models, Mux layer, Demux layer and DenseNet[18] were combined together. Because no information is lost, input images could be down–sampled directly instead of pre–processed with convolutional layers. Feature maps processed by convolutional layers are always in low resolution, thus leading to high efficiency. Besides, DenseNet is also designed for performance, which used densely skip connections to reduce the parameters of each convolutional layer.

3. Our loss function adds a weighted L1 cost and a contextual loss to the total loss function used in DPED method, which can improve the perceptual quality of output images. For validation, we use the NIQE index to find the model with the best perceptual quality.

4 Method

Image enhancement is a sub–task of image–to–image translation, which would translate low quality images to images with high quality. So our target is to learn a mapping from domain X to domain Y, given training dataset $\{x_i\}_{i=1}^{N}$ where $x_i \in X$ and $\{y_i\}_{i=1}^{M}$ where $y_i \in Y$. We denote the data distribution as $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$. In order to get high frequency details, we add noise to the input images. So our model is target to learn the mapping from the observed images x and noises z to y: $G : \{x, z\} \rightarrow y$. In addition, we introduce the discriminator D that is trained to distinguish between images $\{y\}$ and produced images $\{G(x)\}$. Finally, the generator G is trained to produce outputs that cannot be distinguished from "real" images by D. This diagram of the system is shown in Figure 5.

To train the model, we use a multi–term loss function which composed of adversarial loss, weighted loss, perceptual loss, contextual loss, color loss and total variation loss.

4.1 Network Architecture

Motivation. Our network architecture is motivated by the well–known design of multi–rate system in digital signal processing [22][23][24]. In multi–rate systems one is interested to analyze an image at different low resolutions without losing information. For one dimensional signals we can take odd and even samples (demuxing) into different filters. If the filters satisfy the so–called Vetterli and Vaidyanathan conditions [23] then the original signal can be recovered. Perfect reconstructions is achieved with a system that filters in low resolution and recombine them into a high–resolution image (muxing). This principle also applies when we replace filters by convolutional networks since these can be interpreted as generalized adaptive filters [25]. In prior work we have applied this idea to design image super–resolution systems[26] [27][28] using a so–called MuxOut layer. The later considers only the synthesis stage of multi–rate systems.

Here, we move one step forward to include both the analysis and synthesis stages of multi–rate systems. For image enhancement we need the "perfect reconstruction" property of multi–rate systems to guarantee that we can recover the original content. We know that for linear convolutions perfect reconstruction is possible by the Vetterli and Vaidyanathan conditions. When using convolutional networks the filters are obtained during the training process, with a loss function that can impose the perfect reconstruction target. Our design based on multi–rate principles guarantee that at least one local minima exists to recover the original content. In image enhancement "perfect reconstruction" is not our final target since we want to modify the input image, but it guarantees that the architecture would be able to keep all the information needed to solve the problem when processing at lower resolutions.

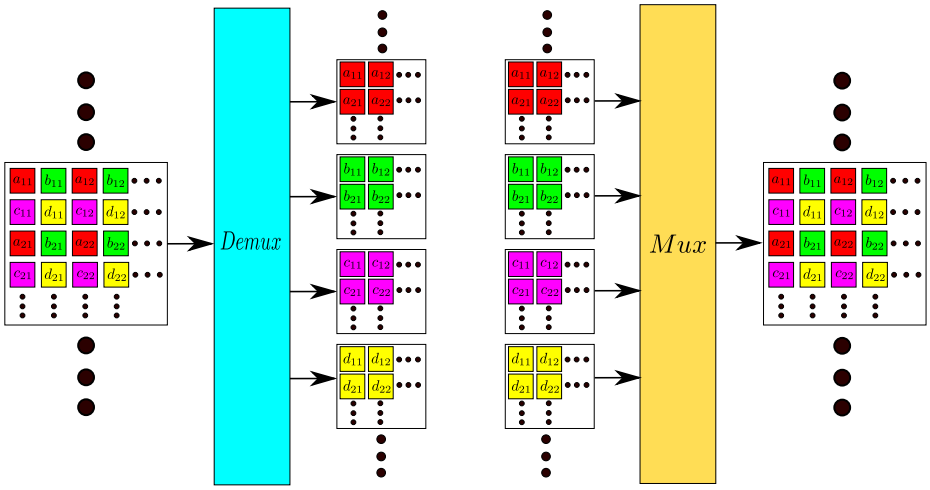

Fig. 1. Demux layer and Mux layer.

Mux Layer. [6] proposed an efficient sub–pixel layer called pixel shuffling, which is also based on the theory of multi–rate filters. The pixel shuffling layer is used for up–sampling in the end of convolutional networks to produce three channels of R, G, and B.

Mux layer is very similar to the pixel shuffling layer, but it can also process feature maps and produce a varied number of feature maps. So it can be used in the middle of convolutional networks. Another property of Mux layer is that it could be used together with Demux layer for perfect reconstruction. As shown in Figure 1, it is the architecture of Mux layer. Mux layer could be used as up–sampling layers in convolutional neural networks, which is based on the theory of multi–filter. Mux would divide input feature maps into small groups, with

each group consisting of four feature maps. By arranging pixels of every four feature maps according to the aforementioned rules, output feature maps would be 2 times higher and 2 time wider than the inputs. And the number of feature maps is one fourth of the number of inputs.

The architecture of Demux layer is shown in Figure 1. Demux layer could be used as a down–sampling layer in convolutional neural networks which is the inverse operation of Mux layer. So Demux layer would convert input feature maps into 4 times the number of feature maps, and the size of outputs is half the height and half the width of the inputs. It is known that, standard down–sampling layers are irreversible, e.g. max pooling, average pooling and strided convolutional layers, that drop or change the value of pixels. So, convolution operations always precede these layers to process high–resolution features with full information. From Figure 1 it could be seen that Demux would not change the value of any pixel, just shuffling the position of them according to some rules. So Demux performs down–sampling without losing any information in the input. Input images could be downscaled directly and be processed in lower resolution. This is one of the keys to improve the performance of deep learning models.

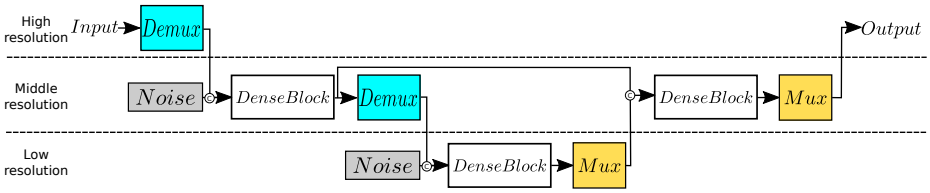

Fig. 2. Normal configuration of generator.

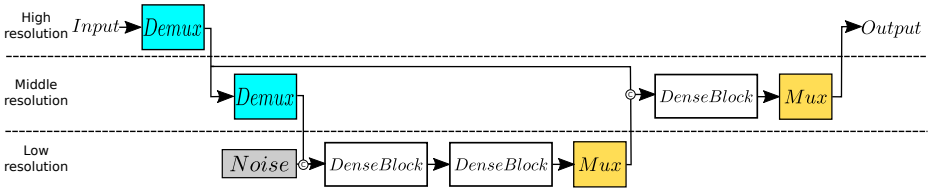

Fig. 3. High–performance configuration of generator.

Generator. With the configuration of Figure 2, the generator would process images in three scales. Input RGB images would be down–sampled by Demux layer, and then processed by a denseblock, which is based on densenet[18]. Then

another Demux and denseblock are used to down–sample the features and process them. After processing in the third scale, Mux layer is used for up–sampling followed by another denseblock. At the end of this model, feature maps are processed by another Mux layer to get the output RGB image. To generate high frequency details and get high perceptual quality, noise channels are added to feature maps after every Demux layers. And we use instance normalization layers after every 3×3 convolutional layers to adjust global parameters of the images and feature maps.

Another configuration of generator is shown in Figure 3. In this configuration, input image is processed by Demux layer two times at the very beginning. So the feature maps would go directly to the third scale. Most of the convolutional operations would process feature maps in very small size, thus to reduce computation complexity and improve performance.

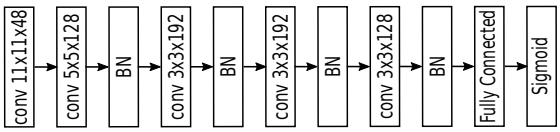

Fig. 4. Model of discriminator.

Discriminator. Figure 4 shows the architecture of the discriminator, which is the same as in DPED[3] system. The discriminator consists of five convolutional layers, each followed by a LeakyReLU activation function and batch normalization. The first, second and fifth convolutional layers are strided with a step size of 4, 2, and 2, respectively. A sigmoid activation function is applied to the output of the last fully connected layer containing 1024 neurons to output the probability that the input image belongs to the target domain.

4.2 Loss Function

Adversarial Loss. For image enhancement, important differences between input images and target images are the brightness, contrast, and texture qualities. We build a generative adversarial network to learn the mapping between input and target domains. The discriminator observes both generated images (fake) and target images (real), and its goal is to predict whether the input image is real or not. It is trained to minimize a cross–entropy loss function. The adversarial loss for generator is defined as a standard GAN[30]:

$$\mathcal{L}_{\text{adv}} = -\sum_{i} \log D(G(x, z), y) \tag{1}$$

where G and D denote the generator and discriminator networks, respectively.

Weighted L1 Loss. In order to get close to the DSLR image we applied a weighted L1 loss during training. We calculate the L1 loss of each channel (R, G, and B) between outputs and reference DSLR images, and then combine them together with different weights, which are taken from the YUV color conversion. The formula for weighted L1 loss is:

$$\mathcal{L}_{\text{l1}} = 0.299\times\|r(G(x))-r(y)\|_1+0.587\times\|g(G(x))-g(y)\|_1+0.114\times\|b(G(x))-b(y)\|_1 \ , \tag{2}$$

where $r()$, $g()$ and $b()$ are the operations to get the R, G, and B channels from one image, respectively.

Perceptual Loss. Inspired by [1], [7] and [3], we also define our perceptual loss based on the feature maps extracted by the pre–trained VGG network to measure high–level perceptual and semantic differences between images. Let $\phi_j(x)$ be the outputs of the j–th layer of the pre–trained VGG network ϕ when processing image x. If j–th is a convolutional layer, $\phi_j(x)$ would be a feature map of shape $C_j \times H_j \times W_j$. Then the perceptual loss of the generator is the Euclidean distance between feature representations:

$$\mathcal{L}_{\text{perc}} = \frac{1}{C_j H_j W_j}[||\phi_j(y) - \phi_j(G(x,z))||^2] \ . \tag{3}$$

As demonstrated in [1], images reconstructed from the high–layers features tend to preserve the content and overall spatial structure, and not the color, texture, and exact shapes. Using perceptual loss for our image transformation network encourages the output images to be perceptually similar to the input images, but does not force them to match exactly.

Contextual Loss. Authors of [19] proposed a novel contextual loss that could be effective for many image transformation tasks. The contextual loss is defined as below:

$$CX(x,y) = \frac{1}{N}\sum_j \max_i CX_{ij} \ , \tag{4}$$

where CX_{ij} is the similarity between features x_i and y_j, which is defined as:

$$w_{ij} = \exp\frac{1-\frac{d_{ij}}{\min_k d_{i,k}+\epsilon}}{h} \tag{5}$$

$$CX_{ij} = \frac{w_{ij}}{\sum_k w_{ik}} \tag{6}$$

where d_{ij} is the Cosine distance between x_i and y_j, ϵ is a fixed value of 0.00001, $h > 0$ is a band–width parameters.

The final contextual loss function is as:

$$\mathcal{L}_{\text{ctx}} = -\log(CX(\phi_j(G(x,z)), \phi_j(y))) \tag{7}$$

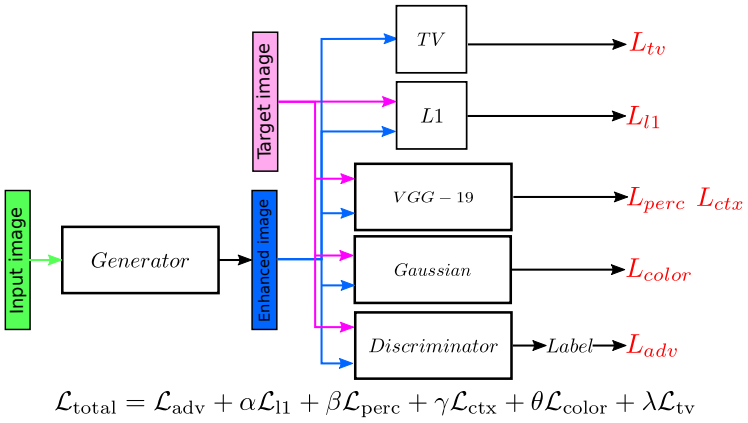

Fig. 5. Training system of proposed method. The coefficients of different part of the total loss is shown in Table 1.

Color Loss. To measure the color difference between the enhanced and target images, authors of [3] proposed applying a Gaussian blur and computing Euclidean distance between the obtained representations. The formula of color loss can be written as:

$$\mathcal{L}_{\text{color}} = \|GB(G(x)) - GB(y)\|_2^2 \tag{8}$$

where $GB()$ is the function of Gaussian blur.

Total Variation Loss. In addition to previous losses, we also add a total variation loss[20] to enforce spatial smoothness of the produced images. The formula of total variation loss is as:

$$\mathcal{L}_{\text{tv}} = \frac{1}{CHW}\|\nabla_x G(x,z) + \nabla_y G(x,z)\| \tag{9}$$

where C, H and W are the dimensions of the generated image $G(x,z)$.

Total Loss. The full loss function of our generator is defined as weighted sum of previews losses with different coefficients:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{adv}} + \alpha\mathcal{L}_{\text{l1}} + \beta\mathcal{L}_{\text{perc}} + \gamma\mathcal{L}_{\text{ctx}} + \theta\mathcal{L}_{\text{color}} + \lambda\mathcal{L}_{\text{tv}} \tag{10}$$

4.3 Training strategy

The training procedure is shown in Figure 5. We use the same dataset as DPED method. The coefficients of different parts of the loss function are presented in Table 1:

Table 1. Coefficients of parts of total loss in Formula10 during the experiments.

Parameter	α	β	γ	θ	λ
Value	5000	10	10	0.5	2000

During the training, we perform validation to avoid over–fitting. For distortion quality, we use PSNR and SSIM between the enhanced images and DSLR images to evaluate the model.

Besides the distortion quality and loss function, we also use NIQE as one of the metrics to evaluate the models. For the PIRM-EoS the evaluation considers different aspects. The evaluation formula measures a distortion index (PSNR), perceptual quality (MS-SSIM during validation and MOS for final evaluation), and running time on different devices. From this, MOS score is the most unpredictable during training and validation because we cannot reproduce the opinion of people. To approximate the MOS scores we use a linear mapping of the NIQE index that is know to be linearly correlated to MOS values. Thus, we could estimate the scores that would be calculated in the test phase of the challenge.

5 Experiments

We use the generator in Figure 2 and Figure 3 for image enhancement. Using the same test datasets as DPED[3], we compare our method against several solutions that are very relevant to the task. To evaluate the performance, we calculate the running–time of each method while processing images of size 1280×720.

5.1 Baselines

Apple Photo Enhancer(APE) is a commercial product known to generate one of the best visual results, while the algorithm is unpublished. The method is triggered using automatic Enhance function from the Photos APP. It performs image improvement without taking any parameters.

SRCNN[2] is a fundamental baseline super–resolution method, thus addressing a task related to end–to–end image–to–image translation. Hence we chose it as a baseline to compare to. The method relies on a standard three–layer CNN and MSE loss function and maps from low resolution / corrupted images to the restored image.

Johnson et al.[1] could get high quality outputs in photo–realistic super resolution and style transferring tasks. The method is based on a deep residual network that is trained to minimize a VGG–based loss function.

DPED.[3] is one of the state of the art in image enhancement. A similar loss function to our method is used for the training to minimize the adversarial loss, perceptual loss, color loss and total variation loss.

ResNet.[17] is an important baseline in the field of classification and image processing. Denseblocks are widely used in deep convolutional networks. In the experiments, we use two configurations of ResNet, ResNet–8–32 and ResNet–12–64.

5.2 Evaluations

Table 2. PSNR and SSIM compared with different methods on DPED test images

Method	PSNR	SSIM
APE	17.28	0.8631
SRCNN[2].	19.27	0.8992
Johnson et al[1].	20.32	0.9161
resnet–12–64[17]	22.43	0.9203
resnet–8–32[17]	22.57	0.9163
DPED[3]	20.08	0.9201
ours normal	**22.98**	**0.9235**
ours high–performance	**22.73**	**0.9206**

Distortion evaluation. We compared the PSNR and SSIM of enhanced images with APE, SRCNN, Johnson et al., ResNet, DPED and our method on the task of mapping photographs from iphone 3GS to DSLR(Canon) images, which is part of the DPED test dataset. The results are shown in Table 2. From these results, we see that our method is the best in terms of both PSNR and SSIM in this experiment.

Table 3. NIQE scores compared with different methods, lower NIQE value means better perceptual quality.

Method	SRCNN	ResNet–12–64	ResNet8–32	DPED	ours normal	ours high–performance
NIQE value	6.55	9.37	7.61	6.82	**6.00**	**5.90**

Perceptual evaluation. We calculate the NIQE scores of enhanced images of different methods. The results are shown in Table 3. Results show that our method could get the lowest NIQE values, which means that the images produced by our method have better perceptual quality. We believe that the key factor for this improvement is the contextual loss used in training and NIQE value used in validation.

The produced images of different methods could be seen in Figure 6. From the output images, it could be seen that the results of our method and DPED

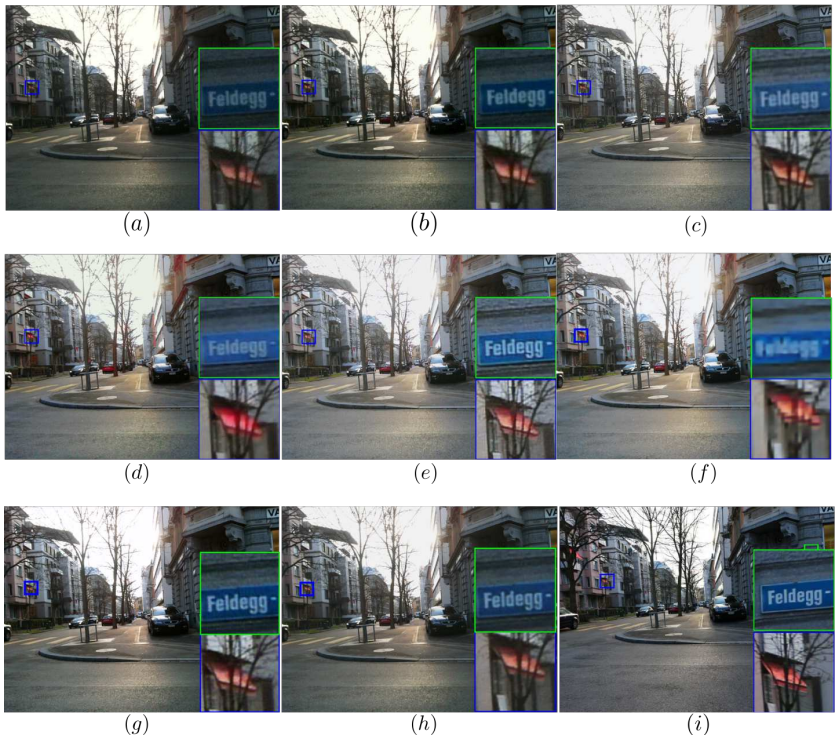

Fig. 6. Results comparison of different methods: (a)original iPhone photo,(b)APE, (c)Dong et al[2]. (d)Johnson et al[1].(e)DPED[3], (f)resnet–8–32, (g)our model in normal configuration, (h)our model in high–performance configuration, (i)DSLR image.

are closer to the DSLR image than other methods. We observe, for example, that DPED increases the brightness excessively in this experiment. The brightness is visibly stronger than DSLR image. Our method could balance the contrast and artificial details better than other methods.

Performance evaluation. We run this experiment to measure the processing time of our method and other methods with input image resolution 1280×720. Table 4 shows that our method shows a significant advantage in processing speed. It is the Mux and Demux layers that bring this advantage, because the convolutional operations do now need to process feature maps in large size.

Ablation study. We did the ablation study of the normal configuration generator by removing every part of loss function to find the effect of each loss. Results are shown in Table 5. We can get the highest PSNR and SSIM with full losses. When contextual loss was removed the NIQE increased obviously, which means

Table 4. Running–time performance of different methods.

Method	SRCNN	ResNet–12–64	ResNet8–32	DPED	ours normal	ours high–performance
running time(ms)	2044	12539	3928	16072	**1148**	**806**

the contextual loss help to improve the perceptual quality of outputs. The lowest PSNR was got without L1 loss, which means that the L1 loss was effective to improve distortion. When remove adversarial loss, L1 loss or perceptual loss, the NIQE was better than system with full losses, but with worse distortion. For the PIRM challenge, our system could get balance of perceptual and distortion. More visual results about ablation study are in Figure 7

Table 5. Ablation study of losses and noise

Condition	PSNR	SSIM	NIQE
no contextual loss	22.44	0.9114	6.43
no adversarial loss	22.64	0.9173	5.64
no L1 loss	21.79	0.9121	5.08
no color loss	22.17	0.9166	6.07
no perceptual loss	22.72	0.9158	5.32
no total variation loss	22.53	0.9171	6.13
full losses and noise	22.98	0.9235	6.00
no noise	22.84	0.9207	6.87

We also do another by removing noise with full losses to prove the effect of noise. Result was in the bottom of Table 5. It can be seen that without noise, PSNR and SSIM became a little worse but NIQE got worse a lot, which indicated that noise could help to improve the perceptual quality obviously. Visual results about removing noise are in Figure 7

5.3 Results of the challenge

We took part in track B of the PIRM–EoS Challenge 2018 with the normal configuration of generator as shown in Figure 2. The official results of this challenge is shown in Figure 8. We achieved the third place according to MOS scores. And the 4th, 4th and 5th places according to scoreA, scoreB and scoreC, respectively. We believe that the high ranking in perceptual quality is due to the use of contextual loss and NIQE index during training and validation stages, which help us to improve MOS scores.

5.4 Limitations

Although our method can perform the enhancement for low quality photos, it can not be used for any kind of bad images. Several typical failure cases are

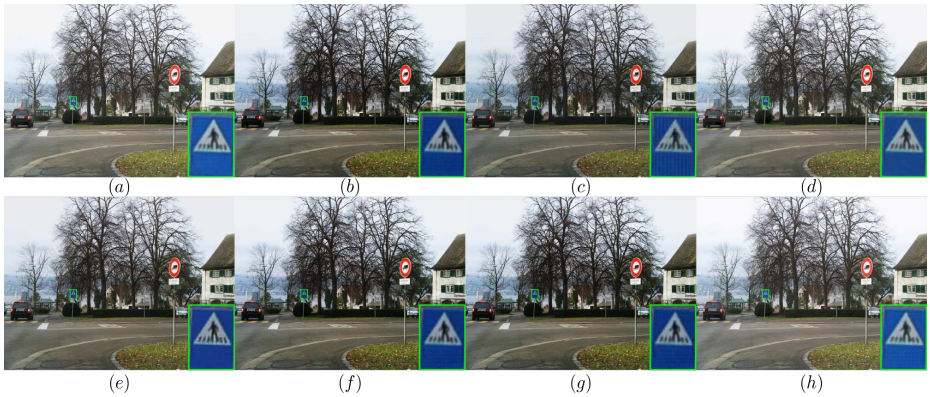

Fig. 7. Results of ablation study: (a)full losses and noise,(b)no L1 loss, (c)no perceptual loss, (d)no color loss, (e)no adversarial loss, (f)no contextual loss, (g)no total variation loss, (h)no noise.

Track B, Final

Team	PSNR	MS-SSIM	MOS	CPU, ms	GPU, ms	Razer Phone, ms	Huawei P20, ms	RAM	Score A	Score B	Score C
MLPhoenix [1st]	21.99	0.9125	2.6804	682	64	1472	2187	1.4GB	14.72	20.06	19.11
EdS	21.65	0.9048	2.6523	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	-	-	1.6GB	13.19	16.31	16.93
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	2.465	1448	81	1987	3061	1.6GB	9.86	10.43	11.05
DPED-Baseline	21.38	0.9034	2.4411	20462	1517	37003	Out of memory	3.7GB	2.89	4.9	3.32
Geometry	21.79	0.9068	2.4324	833	83	1209	1843	1.6GB	12.0	12.59	14.95
IV SR+	21.6	0.8957	2.4309	1375	125	1812	2508	1.6GB	8.13	9.26	10.05
SRCNN-Baseline	21.31	0.8929	2.295	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

Fig. 8. Results of the trackB of PIRM–EoS Challenge 2018. Because of the contextual loss and NIQE, we are in the third place according to MOS scores, which is better than most of the players.

shown in Figure 9. On translation tasks whose input are photos with low–light, our method often succeeds. We have also explored the enhancement with high–light images, with little success. For example, images with overexposure can not be translated to high-quality images. One reason is that the network could not adjust the brightness and contrast depend on the inputs with this training system. Another reason is the dataset of DPED we used during training only contains low–light images. Adding more self–adaption and extending training data are important works in the future.

References

1. J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super–Resolution. Springer International Publishing, Cham, 694–711 (2016)

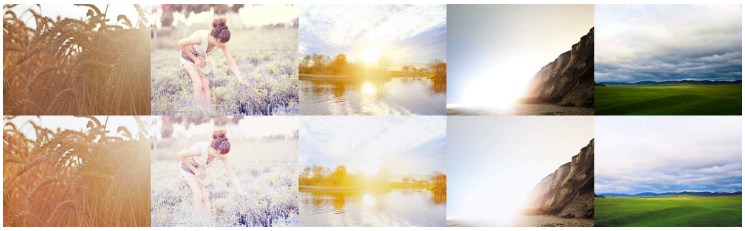

Fig. 9. Failure examples of our methods.Top: input images. Bottom: output images.

2. C. Dong, C. C. Loy, K. He, and X. Tang. Learning a Deep Convolutional Network for Image Super–Resolution. Springer International Publishing, Cham, 184–199 (2014)
3. A. Ignatov, N. Kobyshev, K. Vanhoey, R. Timofte and L. v. Gool. DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks. Proceedings of the IEEE international conference on computer vision(2017)
4. J. Kim, J. K. Lee and K. M. Lee. Accurate image super resolution using very deep convolutional networks. IEEE Conference on Computer Vision and Pattern Recognition, 1646–1654(2016)
5. X. Mao, C. Shen, and Y. B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. Advances in neural information processing system 29, 2802–2810(2016)
6. W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert and Z. Wang. Real–time single image and video super–resolution using an efficient sub–pixel convolutional nerual networks. IEEE Conference on Computer Vision and Pattern Recognition,(2016)
7. C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang and W. Shi. Photo–realistic single image super–resolution using a generative adversarial network. IEEE Conference on Computer Vision and Pattern Recognition,(2017)
8. B. Cai, X. Xu, K. Jia, C. Qing and D. Tao. Dehazenet: An end–to–end system for single image haze removal. IEEE Transactions on Image Processing, 25(11):5287–5298, (2016)
9. M. Hradis, J. Kotera, P. Zemcik and F. Sroubek. Convolutional neural networks for direct text deblurring. Proceedings of BMVV 2015. The British Machine Vision Association and Society for Pattern Recognition. (2015)
10. Z. Ling, G. Fan, Y. Wang, and X. Lu. Learning deep transmission network for single image dehazing. IEEE Conference on Computer Vision and Pattern Recognition,(2016)
11. W. Ren. S. Liu, H. Zhang, J. Pan, X. Cao, and M. H. Yang. Single image dehazing via multi-scale convolutional neural networks. Springer International Publishing, Cham 154–169, (2016)
12. X. Zhang and R. Wu. Fast depth image denoising and enhancement using a deep convolutional network. IEEE International Conference on Acoustics, Speech and Singnal Processing, 2499–2503, (2016)
13. P. Svoboda, M. Hradis, D. Barina, and P. Zemcik. Compression artifacts removal using convolutional neural networks. CoRR, sbs,1605.00366 (2016)

14. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360,(2016).
15. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, (2017).
16. X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. arXiv preprint arXiv:1707.01083, (2017)
17. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition,770778, (2016).
18. G. Huang, Z. Liu, and L. V. D. Maaten. Densely connected convolutional networks. IEEE Conference on Computer Vision and Pattern Recognition,(2017)
19. R. Mechrez, I. Talmi, and L. Z. Manor. The contextual loss for image transformation with non aligned data. arXiv preprint arXiv:1803.02077, (2018)
20. H. A. Aly and E. Dubois. Image up-sampling using totalvariation regularization with a new observation model. IEEE Transactions on Image Processing, 14(10):16471659, Oct 2005
21. A. Mittal, R. Soundararajan, A. C. Bovik. Making a "completely blind" image quality analyzer. IEEE Signal Processing Letters, 20(3), 209-212, (2013)
22. J.G. Proakis and D.G. Manolakis. Digital Signal Processing. Prentice Hall international editions. Pearson Prentice Hall, (2007)
23. P. P. Vaidyanathan. Multirate Systems And Filter Banks, Prentice Hall, Englewood Cliffs, (1993)
24. S. Mallat. A Wavelet Tour of Signal Processing, Academic Press, (1998).
25. P. Navarrete and H. Liu. Convolutional networks with muxout layers as multi-rate systems for image upscaling. CoRR, abs/1705.07772, (2017)
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)
27. P. Navarrete and H. Liu. Upscaling beyond superresolution using a novel deep learning system. GPU Technology Conference, March (2017)
28. R. Timofte, S. Gu, J. Wu, L. Van Gool, L. Zhang, M.-H. Yang, M. Harris, G. Shakhnarovich, N. Ukita, et al. 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)
29. Blau, Y., Michaeli, T.: The perception-distortion tradeoff. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
30. I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al. Generative Adversarial Nets. NIPS, (2014).
31. Z. Qin, Z. Zhang, S. Zhang, H. Yu, and Y. Peng. Merging and evolution: improving convolutional neural networks for mobile applications. arXiv preprint arXiv:0803.09127, (2018)
32. H. Liu, P. Navarrete and D. Zhu. Arsty-GAN: a style transfer system with improved quality, diversity and performance. International Conference on Pattern Recognition,(2018)