Deep Networks for Image-to-Image Translation with Mux and Demux Layers and Deep Laplacian Pyramid networks with Denseblock for image super resolution –PIRM2018 factsheet–

August 14, 2018

1 Team details

- Team name
  BOE-SBG
- Main contact name, address, phone number, and email
  Liu Hanwen,
  No.9 Dize Road, BDA, Beijing, 100176, P.R. China
  +86 15201020615
  liuhanwen@boe.com.cn
- Team members and affiliations
  Liu Hanwen, BOE Technology Group Co., Ltd
  Pablo Navarrete Michelini, BOE Technology Group Co., Ltd
  Zhu Dan, BOE Technology Group Co., Ltd
  Hu Fengshuo, BOE Technology Group Co., Ltd
- Team website URL (if any)
  We do not have a website.
- User names and entries on the PIRM 2018 challenge (development/validation and testing phases)
  All the submission were submitted by Liu Hanwen(liuhanwen@boe.com.cn)
  In validation phase, we had totally submitted 5 models, the detail is as Table 1.

Table 1: All the submissions

Date	Track	PSNR	SSIM	Speed-up	RAM	ScoreA	ScoreB	ScoreC
07-11	TrackB	22.45	0.92	1.32	1.6G	10.444	12.094	10.734
07-12	TrackA	23.31	0.8807	0.14	3.0G	7.581	-5.139	1.152
07-17	TrackB	22.21	0.9111	1.24	1.6G	8.431	8.131	8.238
08-02	TrackB	21.99	0.9103	0.09	1.6G	5.159	5.307	4.304
08-03	TrackB	22.23	0.9132	2.09	1.6G	10.434	10.705	11.169

Table 2: All the submissions

Date	Track	Validation-Track
08-03	TrackA	all
08-03	TrackB	A
08-03	TrackB	B
08-03	TrackB	C

In test phase, we had totally submitted four models for different tracks. The detail is as Table 2.

- Best scoring entries of the team during development/validation phase

  The best entries in Track A during validation phase is the submission in 07-12 in table 1.

  The best entries in Track B during validation phase is the submission in 07-11 in table 1.

- Link to the codes/executables of the solution(s)

  The model for trackA(all):

  https://raw.githubusercontent.com/hanwenliu/hanwenliu.github.io/master/test/sr/model.pb

  The models for trackB:

  validation trackA:

  https://raw.githubusercontent.com/hanwenliu/hanwenliu.github.io/master/test/enhancement/A/model.pb

  validation trackB:

  https://raw.githubusercontent.com/hanwenliu/hanwenliu.github.io/master/test/enhancement/B/model.pb

  validation trackC:

  https://raw.githubusercontent.com/hanwenliu/hanwenliu.github.io/master/

test/enhancement/C/model.pb

2 Contribution details

- Title of the contribution

  Enhancement:

  Deep networks for image-to-image translation with Mux and Demux layers

  Super Resolution:

  Deep Laplacian Pyramid networks with Denseblock for image super resolution

- General method description

  In the enhancement networks, we use Mux layers and Demux layers for up-sampling and down-sampling. Compared with max/average pooling and strided convolutional layers, Mux and Demux layers could do the samplings more efficiently. In the down-sampling, Demux layers could translate features to be smaller without losing any information. So we could get the ouput with high efficiency and high perceptual quality.

  Among the Mux and Demux layers, we process the images with operations called denseblocks, which include several convolutional layers that were connected with dense skip connections. This idea was inspired by DenseNet in the reference 1.

  With the advantages of Mux, Demux and denseblocks, we could get high performance of this model.

  For the task of super resolution, We propose our SR network based on the Laplacian pyramid framework with Denseblock which was inspired by the reference 4. The architecture of the SR model is shown in figure5. In our model, we use convolution and transpose convolution layers for down-sampling and up-sampling. Then, we process the images with denseblocks, which followed by a convolution layer to keep the output feature numbers consistent with the input feature numbers of denseblock. Note that the parameters of denseblock, conv layer and up-sampling 2 times layer at lower levels are shared with higher levels. The output upsample image is then combined with the input image to produce a high-resolution output image.

- Description of the particularities of the solutions deployed for each of the challenge competitions and/or tracks

  For the enhancement, we use Demux and Mux layers that could perform the dowm-sampling and up-sampling with high efficiency and keeping all the information within the features.

  For the super resolution, we shared parameters of denseblocks and strided convolutional and transpose convolutional layers in different levels, which

could help to get high performance. By the way, because of the sharing parameters we could get a better 2x up-scaler, which could be combined for 4x, 8x super resolution tasks.

For both tasks, we use denseblocks coming from densenet(reference 1), which could reduce the number of parameters to get high performance.

For different validation tracks, we use the same architecture but different parameters.

- References

1. G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger. "Densely Connected Convolutional Networks". [arXiv:1608.06993]

2. R. Mechrez, I. Talmi, L. Zelnik-Manor. "The contextual loss for image transformation with non-aligned data". [arXiv: 1803.02077]

3. A. Mittal, R. Soundararajan, and A. C. Bovik, "Making acompletely blind image quality analyzer", In IEEE Signal Processing Letters, 2013.

4. W. S. Lai, J. B. Huang, N. Ahuja, M. H. Yang. "Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution". [arXiv:1704.03915]

5. P. Navarrete, L. Zhang, and J. He, "Upscaling with deep convolutional networks and muxout layers,"" in GPU Technology Conference 2016, San Jose, CA, USA, May 2016, Poster Session.

- Representative image / diagram of the method(s)

3 Global Method Description

- Total method complexity: all stages

  The model consists of convolutional layers and instance normalization layers. So the running time complexity should be linear to the size of the input.

- Which pre-trained or external methods / models have been used (for any stage, if any)

  To calculate the contextual loss, we use the pretrained VGG-19 network.

- Which additional data has been used in addition to the provided PIRM 2018 training and validation data (at any stage, if any)

  No additional data has been used.

- Training description

  For training, we reuse the code of dped. But we add more terms to the total losses.

  First is a weighted L1 loss, which we calculate the L1 loss of each channels of the image(R, G, and B), and then combine them together with different weights. We use the weights from Y channel conversation.

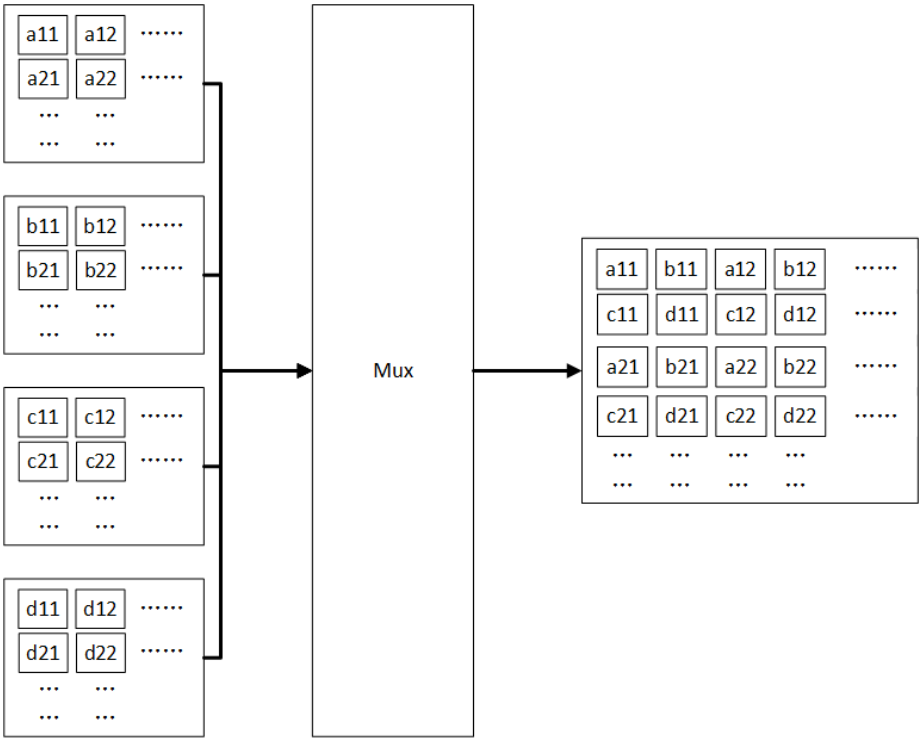

Figure 1: Mux layer that can be used as up-sampling layers in convolutional networks.

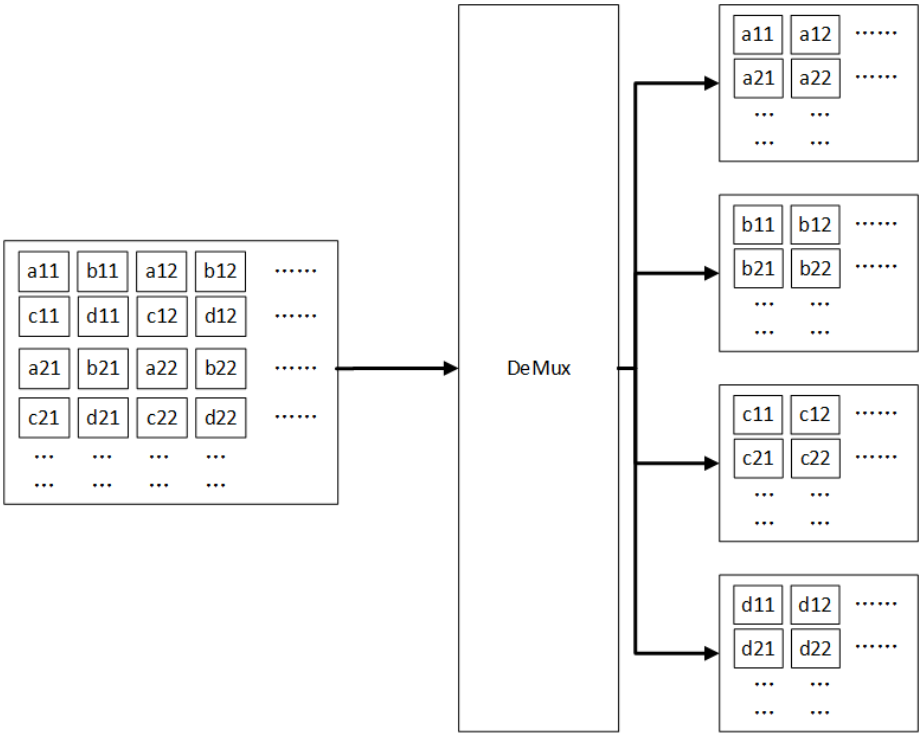

Figure 2: Demux layer that can be used as dowm-sampling layers in convolutional networks.

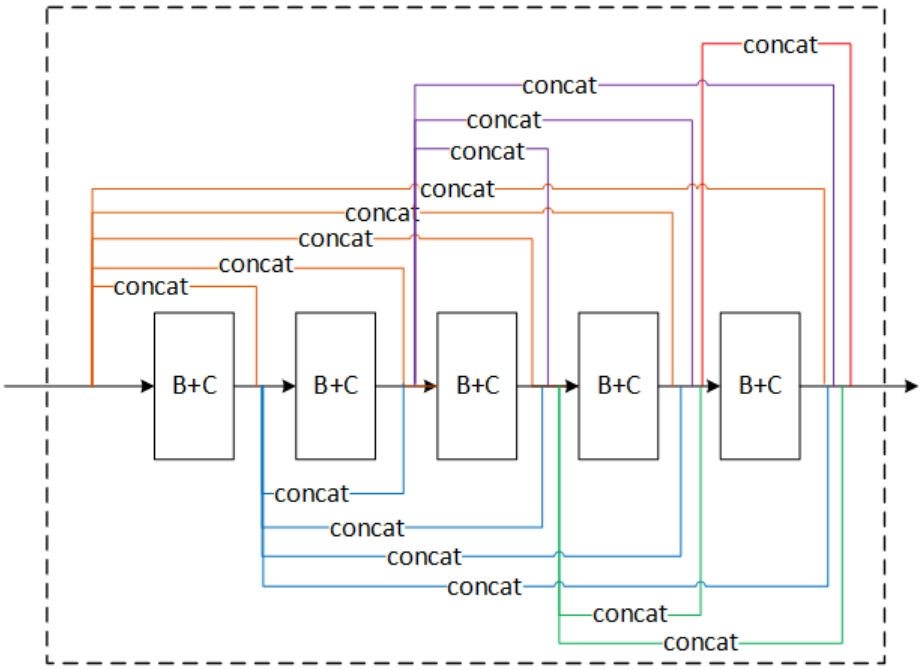

Figure 3: Denseblock we used in the model.

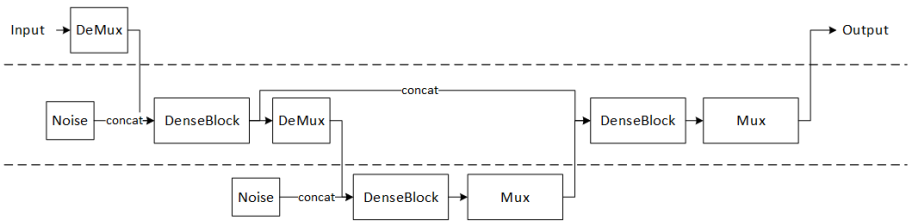

Figure 4: The architecture of the model of enhancement

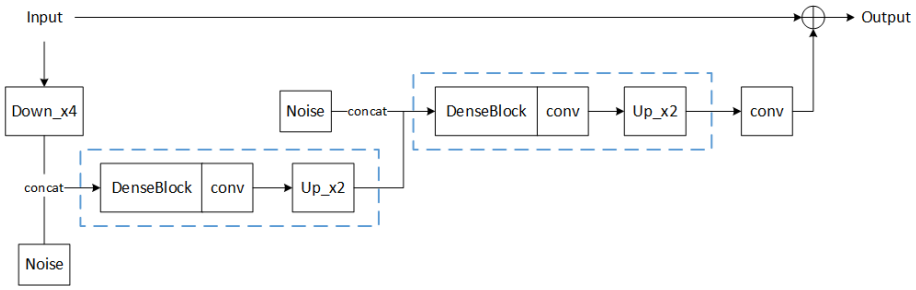

Figure 5: The architecture of the model of super resolution, blue means sharing parameters

The other loss is call contextual loss, which could try to give better perceptual quality to the output. We got the idea of contextual loss from the paper in reference [2].

During the training, we perform the validation every 1000 steps. In the validation, we calculate the PSNR, SSIM and NIQE to find the best model. NIQE is one index that could represent the perceptual quality, which was presented in the paper in reference [3].

Other configurations of the training is as belows: the batch size is 50, the learning rate is decreasing from 5e-4 to zero every 3000 steps, the dataset is exactly the traning data from dped.

- Testing description

  To test the model, we use the script got from the competition. To check the output images, we save the images by add a little code to it.

- Quantitative and qualitative advantages of the proposed solution

  Because of the Mux and Demux layers, we could reduce the size of images and features without lossing any information. So we could gain time and memory efficiency with high quality.

  With the contextual loss in the training and NIQE in the validation, output images could keep high perceptual quality.

- Results of the comparison to other approaches (if any)

  We have not compared our results to other approaches yet.

- Results on other standard benchmarks (if any)

  We do not have results on other standard benchmarks.

- Novelty degree of the solution and if it has been previously published

  We use the Mux and Demux layers as described before. Demux layer could reduce the size of images without lossing any information, so we could do

down-sampling without any convolutional layers, which could speed up a lot.

With the contextual loss and NIQE in validation, the output images could keep a good perceptual quality.

It has not been previously published.

4 Track A: Image Super-Resolution

Any particularities of the deployed solution for the Track A competitions (scores A, B, C) (if applicable)

1. Parameters of convolutional layers and transpose convolutional layers in different levels were shared, thus to get high performance, as shown in Figure5.

2. Denseblocks were used to process the images, as shown in Figure3, which were the key of densenet and could gain high performance.

3. For different validation tracks, we adjust the weights of different parts of the losses during the training.

5 Track B: Image Enhancement

Any particularities of the deployed solution for the Track B competitions (scores A, B, C) (if applicable)

1. Demux and Mux layers were used for down-sampling and up-sampling, as shown in Figure4.. As described before, they could do the downscaling and upscaling without loss any information. Thus we could get image with high quality and high performance.

2. Denseblocks were used to process the images, which were the key of densenet and could gain high performance, as shown in Figure3.

3. For different validation tracks, we adjust the weights of different parts of the losses during the training.

6 Ensembles and fusion strategies

- Describe in detail the use of ensembles and/or fusion strategies (if any).

  We use a single method for one track, and different parameters for different validation tracks.

  For different tracks(super resolution and enhancement), we use different architectures.

  The architecture of the enhancement model is shown in figure4

  The architecture of the super resolution model is shown in figure5

7 Technical details

- Language and implementation details (including platform, memory, parallelization requirements)

  Language: Python
  Framework: Tensorflow 1.8.0
  CUDA: 9.0
  CuDnn: 7.0
  Platform: Tesla P100 GPU(training), GTX 1060(test)
  Memory: No less than 1.6G
  Parallelization requirements: Standard environment for tensorflow 1.8.0.

- Human effort required for implementation, training and validation?

  For implementation, build the environment of suitable version of tensorflow(PC or Server) or tensorflow lite(android devices), and load the model file.

  For training, down load the dataset and adjust the hyper parameters.

  For validation, combine the script of calculating the scores and the script to calculate NIQE(representation of the perceptual quality), and select the model with highest scores and NIQE.

- What is the training/testing time? What is the runtime at test per image on CPU, GPU or smartphone?

  Training time is about 10 minutes per 1000 steps.

  Test time is about 800ms per image.

  The runtime at test per image on GPU is about 800ms per image. We did not test the model on CPU or phones.

- How many parameters are in the model?

  There are 48,744 parameters in each enhancement model.

  There are 57,860 parameters in each super resolution model.

- Is a generative adversarial component used in the model?

  Yes, we use the same configuration of DPED.

- How the perceptual quality is enforced on the solution?

  We use contextual loss and niqe in the validation.

- Comment the robustness and generality of the proposed solution(s)? Is it easy to deploy it for other settings/ image mapping tasks?

  It could be used for arbitrary image sizes.

  It could be easy to use for different applications, such as style transfer and Image-to-Image translation.

- Comment the efficiency of the proposed solution(s)?

  Because of the Demux and Mux layers and denseblocks described above, we could get high performance of the model. It takes about 872 ms to process one image in 1280x720.

- Results of the comparison to other methods, or results on standard benchmarks such as Set5, Set14, B100, Urban100.

  We have not compared to other methods.

8 Other details

- Planned submission of a solution(s) description paper at PIRM 2018 workshop.

  Yes. We would like to submit one paper at PIRM 2018 workshop.

- General comments and impressions of the PIRM2018 challenge.

  We have extensive experience on image generation and image processing. The subjective nature of perceptual image quality makes this problem very different. We are happy to know about this challenge that focuses on image perceptual quality. And what is more important is that this challenge is focus on applications, not only quantitative advantages.

- What do you expect from a new challenge in image restoration and enhancement?

  We hope there would be some challenges about super resolution and enhancement on videos.

- Other comments: encountered difficulties, fairness of the challenge, proposed subcategories, proposed evaluation method(s), etc.