NTIRE 2020 Demoireing Challenge Factsheet -C3Net: Demoireing Network Attentive in Channel, Color and Concatenation-

Sangmin Kim, Hyungjoon Nam, Jisu Kim, Jechang Jeong

March 24, 2020

1 Team details

- Team name
  - Reboot
- Team leader name
  - Sangmin Kim
- Team leader address, phone number, and email
  - Address: Room 607, R&D Building,
  222, Wangsimni-ro, Seongdong-gu, Seoul, Korea, 04763
  - Phone number: +82-10-7295-1909
  - Email address: ksmh1652@gmail.com
- Rest of the team members
  - Hyungjoon Nam, Jisu Kim, Jechang Jeong
- Team website URL (if any): None
- Affiliation
  - Image Communication & Signal Processing Laboratory,
  Hanyang University, Seoul, Korea
- Affiliation of the team and/or team members with NTIRE2020 sponsors (check the workshop website): None
- User names and entries on the NTIRE2020 Codalab competitions (development/validation and testing phases)
  - sangmin_kim, Nam, kevin_k, jimmy_jeong
- Best scoring entries of the team during development/validation phase
  - Track 1: PSNR 41.30 / SSIM 0.99
  - Track 2: PSNR 40.55 / SSIM 0.99

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

- Link to the restoration results of all frames
  - Track 1: Single Image
  - Track 2: Burst

2 Contribution details

- Title of the contribution
  - C3Net: Demoireing Network Attentive in Channel, Color and Concatenation

- General method description

  - Demoireing Network Attentive in Channel, Color and Concatenation (C3Net) is inspired by Residual Non-local Attention Network (RNAN) [1] and Deep Iterative Down-Up CNN for Image Denoising (DIDN) [2].

  - The entire network consists of n Attention Via Concatenation Blocks (AVCBlocks) in global feature fusion used in Residual Dense Network for Image Super-Resolution (RDN) [3].

  - An AVCBlock consists of two branches: trunk branch and mask branch. In trunk branch, there are r residual blocks (ResBlocks) in parallel to retain the original values of input in diverse ways. In mask branch, there are a attentive block (AttBlock) for guiding the values from trunk branch to demoire. Outputs from two branches are concatenated and the number of channels of concatenated features halves for entering next block.
  - A ResBlock is similar as one used in Local Excitation Network for Restoring a JPEG-Compressed Image [4] which consists of one convolutional layer, one PReLU layer, and one convolutional layer again. We added channel attention [5] with ReLU to cope with demoireing problems related to colors.

  - An AttBlock is a U-net with Resblocks, convolutional layers with stride 2 for downscaling, and pixel shuffle with kernel size 2 for upscaling features. The scaling layers and usage of U-net as a block benchmarked DIDN.

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

  - In Track 2: Burst, we added global maxpooling [6] following each AVCBlock. The algorithm give feature maps which have maximum among 7 images and replicate and concatenate 7 times to match the dimension. Please refer to Figure 6.

- The proposed C3Net concatenates the output of AVCBlock and the output of AVCBlock and global maxpooling layer and the number of channels of concatenated features halves for entering next block.

- References

[1] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu, "Residual Non-local Attention Networks for Image Restoration," *arXiv preprint arXiv:1903.10082*, 2019.
[2] Songhyun Yu, Bumjun Park, and Jechang Jeong, "Deep Iterative Down-Up CNN for Image Denoising," *CVPRW 2019*, pp.0-0, 2019.
[3] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu, "Residual Dense Networks for Image Super-Resolution, *CVPR 2018*, pp. 2472-2481, 2018.
[4] Songhyun Yu, Bumjun Park, and Jechang Jeong, "Local Excitation Network for Restoring a JPEG-Compressed Image," *IEEE Access*, pp.138032-138042, 2019.
[5] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu, "Image Super-Resolution Using Very Deep Residual Channel Attention Networks," *ECCV 2018*, 2018.
[6] Miika Aittala and Frédo Durand, "Burst Image Deblurring Using Permutation Invariant Convolutional Neural Networks," *ECCV 2018*, 2018.

- Representative image / diagram of the method(s)

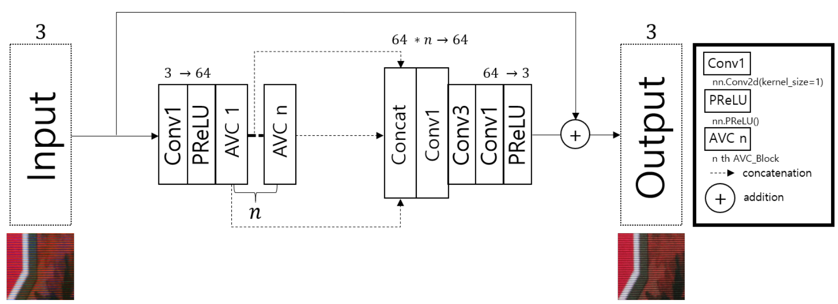

Figure 1. The structure of the proposed C3Net

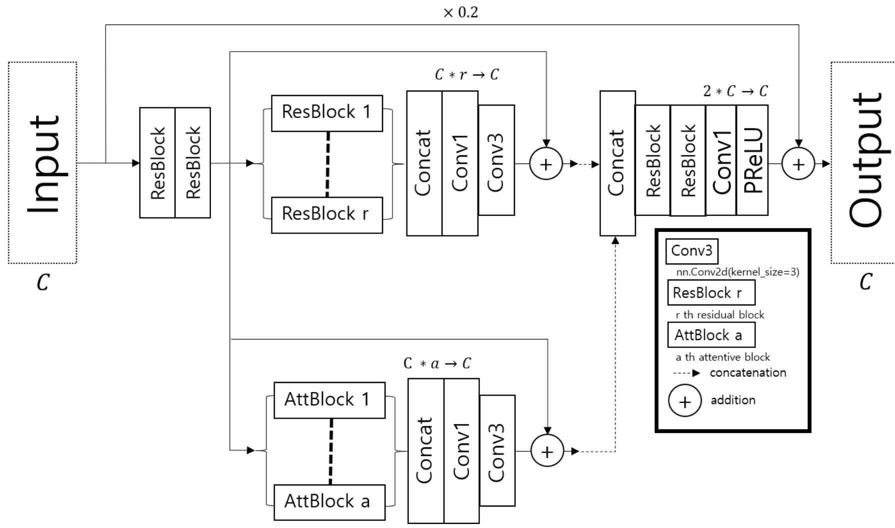

Figure 2. The structure of Attention-Via-Concat Block (AVCBlock)

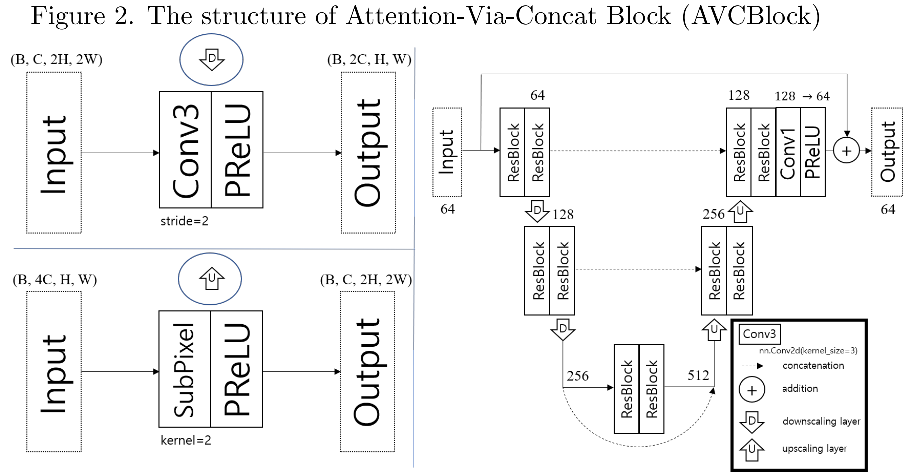

Figure 3. The structure of downscaling layer (upper left), upscaling layer (lower left), and attentive block (AttBlock, right)

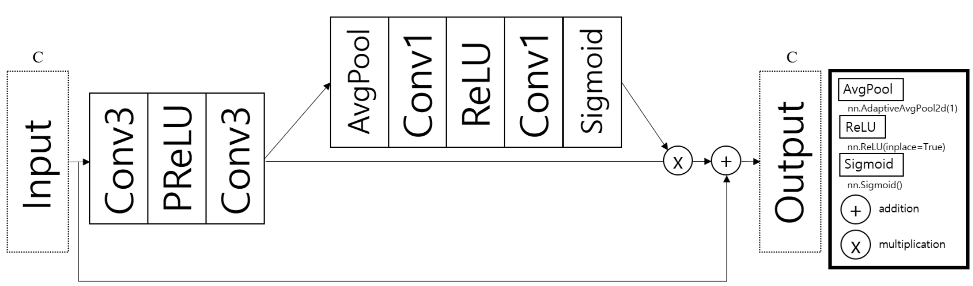

Figure 4. The structure of residual block (ResBlock)

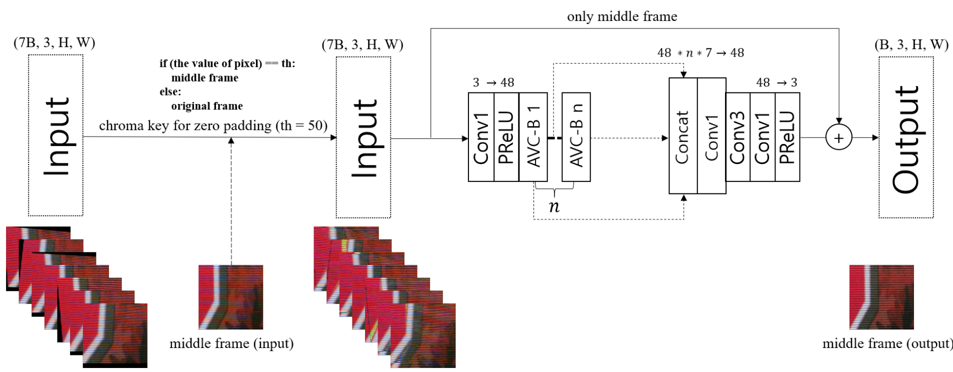

Figure 5. The structure of the proposed C3Net-Burst

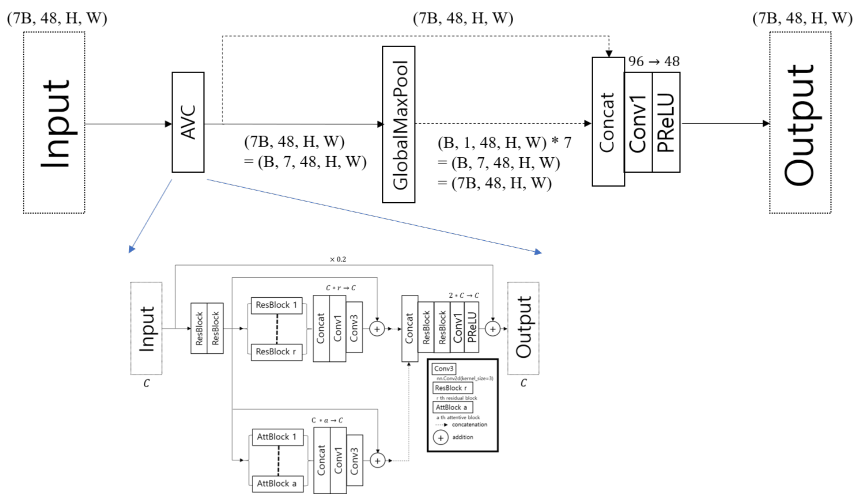

Figure 6. The structure of the proposed AVC-Burst Block (AVC-B Block)

3 Global Method Description

- Total method complexity: all stages
  (Model size)
  - Track 1: Single Image 627MB
  - Track 2: Burst 81MB

  (Model parameters)
  - Track 1: Single Image 164,182,745 parameters
  - Track 2: Burst 21,105,078 parameters

  (Runtime)
  - Track 1: Single Image 0.87 seconds per image
  - Track 2: Burst 0.22 seconds per image

- Which pre-trained or external methods / models have been used (for any stage, if any): None
- Which additional data has been used in addition to the provided NTIRE training and validation data (at any stage, if any): None
- Training description
  - We set the batch size of the training patches to 1 for maximizing the model size and have better results, and train our proposed network for about 200 epochs in Track 1: Single Image whose number of iterations is 9000 per epoch and 82 epochs in Track 2: Burst whose number of iterations is 9999 per epoch.
  - We utilize the ADAM optimizer with an initial learning rate of 0.0001 and momentum 0.9.
  - At first, the learning rate is divided by 2 for every 30 epochs.
  - When the loss starts converging, the learning rate is divided by 2 for every 10 epochs.
  - For the activation function, we used the parametric rectified linear unit (PReLU) except in channel attention.
  - We use L1 Loss or L1 (RGB) + L1 (UV) for the loss function.
  - In Track 1: Single Image, We set the number of increased channels (C) to 64, the number of AVCBlocks (n) to 22, the number of ResBlocks in trunk branch (r) to 2 and the number of AttBlocks in mask branch (a) to 1.
  - In Track 2: Burst, We set the number of increased channels (C) to 48, the number of AVCBlocks (n) to 5, the number of ResBlocks in trunk branch (r) to 2 and the number of AttBlocks in mask branch (a) to 1.
  - In Track 2: Burst, we use chroma key for padding some training input images whose pixels are zero irregularly, except middle images whose name is "*_3.png". The pixel value of the threshold in all R, G, and B channels is 50.
- Testing description
  - In Track 2: Burst, we use chroma key for padding some validation/testing input images whose pixels are zero irregularly, except middle images whose name is "*_3.png". The pixel value of the threshold in all R, G, and B channels is 50.
- Quantitative and qualitative advantages of the proposed solution
  - PSNR scores are greater than 41. Furthermore, the PSNR result above 41 was possible for a few entries among over 40 entire entries during development/validation phase. Runtime is also reasonable, running at about less than 1 second per test image which is faster than that of entries in first and second place during development/validation phase.

- Results of the comparison to other approaches (if any)
  - When we used L1 (RGB) + L1 (UV) loss, named L1 color loss, we can get results about 0.1 more PSNR than those by using L1 Loss only.

- Results on other benchmarks (if any): None

- Novelty degree of the solution and if it has been previously published
  - The proposed C3Net used attention algorithm via concatenation and showed good results.
  - When pre-processing the input images in Track 2: Burst, we produced inputs with good quality by chroma key and the variation of pre-processing is so easy that we only change the threshold value.
  - By combining L1 (UV) loss with conventional L1 Loss, the proposed C3Net showed good results.
  - Not yet published.

- It is OK if the proposed solution is based on other works (papers, reports, Internet sources (links), etc). It is ethically wrong and a misconduct if you are not properly giving credits and hide this information.: Understood

4 Competition particularities

- Applied channel attention and L1 (RGB) + L1 (UV), namely L1 color loss to solve moire patterns in colored stripes.

- In Track 2: Burst, we used chroma key for pre-processing because of randomly zero pixels in burst images.

- In Track 2: Burst, C3Net used less blocks to endure large inputs comprised of burst images.

- In Track 2: Burst, C3Net used global maxpooling to have feature maps with good quality from a set of 7 burst images.

5 Ensembles and fusion strategies

- Describe in detail the use of ensembles and/or fusion strategies (if any).
  trained flip (np.flipud np.fliplr, $\times 4$) and used self-ensemble.

- What was the benefit over the single method?
  Without it, the PSNR result is 40.99.
  With it, it is 41.30.

- What were the baseline and the fused methods?
  None

6 Technical details

- Language and implementation details (including platform, memory, parallelization requirements)
  - Python 3.6.9, Pytorch 1.4.0
  - OOM issue might be occurred if GPU memory is lesser than 11GB.

- Human effort required for implementation, training and validation?
  - Pre-processed images using chroma key (in case of Track 2: Burst), implemented models, training environment, and test environment, as well as performed many experiments comparing loss functions, model sizes, etc.

- Training/testing time? Runtime at test per image.
  - Training Time: 10 days
  - Testing Time: 0.87 seconds per image. (Track 1)
  - Testing Time: 0.22 seconds per image. (Track 2)

- Comment the robustness and generality of the proposed solution(s)? Is it easy to deploy it for other sets of downscaling operators?
  - Provided that GPU memory is more than 11GB, the proposed C3Net ensures its robustness and generality.
  - Yes. All thing you have to do for using C3Net fully is giving inputs and ground truth with good quality.

- Comment the efficiency of the proposed solution(s)?
  - Provided that GPU memory is more than 11GB, the proposed C3Net ensures demoired image in less than 1 second per image.

7 Other details

- Planned submission of a solution(s) description paper at NTIRE2020 workshop.
  - I'll improve my validation/testing results and submit my paper soon.

- General comments and impressions of the NTIRE2020 challenge.
  - It's a lot challenging for me trying to make better results under extreme conditions in NTIRE2020.

- What do you expect from a new challenge in image/video restoration and enhancement?
  - Definitely.

- Other comments: encountered difficulties, fairness of the challenge, proposed subcategories, proposed evaluation method(s), etc.
  - The datasets in Track 2: Burst are abnormal, so training was too hard for me.