CSANet: High Speed Channel Spatial Attention Network for Mobile ISP

Ming-Chun Hsyu[1] mchsyu@itri.org.tw
Chih-Wei Liu[1,2] cwliu531@gmail.com
Chao-Hung Chen[1] ZHChen@itri.org.tw
Chao-Wei Chen[1] shoiradon086@gmail.com
Wen-Chia Tsai[1] itriA90147@itri.org.tw

[1]Industrial Technology Research Institute, Hsinchu, Taiwan, R.O.C
[2]National Yang Ming Chiao Tung University, Hsinchu, Taiwan, R.O.C

Abstract

The Image Signal Processor (ISP) is a customized device to restore RGB images from the pixel signals of CMOS image sensor. In order to realize this function, a series of processing units are leveraged to tackle different artifacts, such as color shifts, signal noise, moire effects, and so on, that are introduced from the photo-capturing devices. However, tuning each processing unit is highly complicated and requires a lot of experience and effort from image experts. In this paper, a novel network architecture, CSANet, with emphases on inference speed and high PSNR is proposed for end-to-end learned ISP task. The proposed CSANet applies a double attention module employing both channel and spatial attentions. Particularly, its spatial attention is simplified to a light-weighted dilated depth-wise convolution and still performs as well as others. ***As proof of performance, CSANet won 2^{nd} place in the Mobile AI 2021 Learned Smartphone ISP Challenge with 1^{st} place PSNR score****.*

1. Introduction

In conventional camera pipelines, no matter smartphones or DSLR cameras, complex and confidential hardware processes are employed to perform image signal processing, a specialized digital signal processor for reconstructing RGB images from raw Bayer images. The ISP pipeline consists of highly complicated DSP steps, e.g., denoising, white balancing, exposure correction, demosaicing, color transform, gamma encoding, and so on. Each step of the ISP pipeline is performed with individual task-specific loss function and hence, the residual error will be accumulated. In order to enhance the quality of RGB images from raw Bayer images, tedious parameter tuning process, usually handcrafted heuristics-based approaches, should be applied. A small change in parameter configuration might lead to different reconstructed RGB images.

Nowadays, smartphones have become a part of a person's daily life. How to make the photo quality of the mobile phone camera, e.g. Huawei P20 mobile camera, as close as possible to the professional one, Canon 5D DSLR, will be the customer's concern. It is known that a well-designed and adjusted ISP can bring competitive quality to the images taken by smartphones. However, applying the conventional ISP pipeline, there might always a big gap between the mobile phone and the professional cameras because each module in the ISP pipeline can neither control the output of the other modules nor recover the signal loss of previous ones.

With the advent of deep learning and the continuous improvements in memory and computational hardware, several research fields including computer vision, graphics, and computational photography have been making much progress. The idea that using a convolutional neural network (CNN) to replace the hardware-based ISP, namely PyNET [12], is supported by the fact that CNN can compensate for the information loss of input images, which is more reliable than the traditional ISP and can effectively break through the hardware limitation [4, 12, 2, 31].

We consider not only the PSNR quality of the RAW-to-RGB image but also the computation time and the total number of model parameters. A novel network architecture, namely CSANet, was proposed. The CSANet emphasized both inference speed and high PSNR. Our proposed method inferred at most 90.8 ms per image and achieved image quality over 23.7dB in the MAI 2021 Learned Smartphone ISP Challenge [8] in the final testing phase.

2. Related Works

2.1. PyNet

With PyNET network [12], it is possible that the low-quality images recorded by compact camera sensors, available in portable mobile devices, can be enhanced and restored. One of the biggest challenges in the RAW-to-RGB mapping task is to get high-quality real data that can be used for training deep CNN models. The AIM challenge on the learned ISP pipeline promoted a novel direction of research aiming at replacing the current tedious and expensive hand-crafted ISP solutions with data-driven learned ones capable to surpass them in terms of image quality. For this purpose, the participants were asked to map the smartphone camera RAW images to the higher quality images captured with a high-end DSLR camera. The AIM 2020 challenge[11], for example, employed the ZRR dataset containing paired and aligned photos produced by the Huawei P20 smartphone and Canon 5D Mark IV DSLR camera.

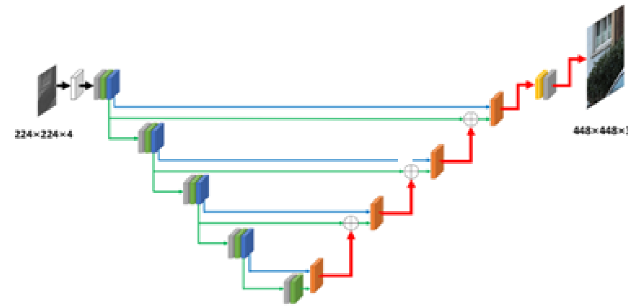

Figure 1. The main network architecture of the AWNet.

2.2. AWNet

Many of the proposed solutions significantly improve the original RAW images in perceptual quality. For example, the MW-ISPNet, a U-Net based multi-level wavelet ISP network, takes advantage of the MWCNN [16] and RCAN [29] architectures. In each U-Net level of MW-ISPNet, a residual group (RG) composed of 20 residual channel attention blocks (RCAB) is embedded. The standard down-sampling and up-sampling operations are replaced with a discrete wavelet transform based (DWT) decomposition to minimize the information loss in these layers. Meanwhile, the AWNet proposed by the team MacAI [2], utilized the attention mechanism and wavelet transform.

The AWNet is consisting of three blocks: lateral block, up-sampling, and down-sampling blocks, as shown in Fig. 1. The lateral block consists of several residual dense blocks (RDB) and a global context block (GCB). Same as the previous team, the authors used the discrete wavelet transform (DWT) instead of the pooling layers to preserve the low-frequency information, though they additionally used the standard downscaling convolutional and pixel shuffle layers in parallel with the DWT layers to get a richer set of learned features.

As shown in Fig. 1, the architecture of such a network generally consists of three parts. The first part, including the input, is to reduce the size of the image and simultaneously extract the feature information. The second part of the network is usually made up of a sequence of similar processing units which means to knead the information from feature maps of previous blocks or layers via different attention mechanisms or channel aggregations. The third part of the network is to transform the features maps into the desired output image resolution. This network design concept can also be found in these papers [23, 14, 15, 29].

2.3. Attention Modules

Although applying the multi-level encoder-decoder structure can capture different scales of objects and fuse semantic features, it cannot leverage the relationship between objects or stuff in a global view, which is also essential to construct the image. To address this problem, some attention blocks or modules were introduced in the proposed network [27, 3, 25, 21]. Attention modules can model long-range dependencies and have been widely applied in many tasks. The work in [20] is the first to propose the self-attention mechanism to draw global dependencies of inputs and apply it in machine translation. For the image vision field, Zhang et al. [28] introduced a self-attention mechanism to learn a better image generator. Meanwhile, the work in [22] applied a self-attention module to explore the effectiveness of the non-local operation in space-time dimensions for videos and images. Moreover, a dual attention network (DANet) [3] was proposed for natural scene image segmentation.

The DANet [3] applied a self-attention mechanism to capture feature dependencies in both the spatial and channel dimensions, respectively. Two parallel attention modules, one is the position attention module and the other is the channel attention module, were proposed. The position attention module would capture the spatial dependencies between any two positions of the feature maps, while the channel attention module would capture the channel dependencies between any two-channel maps. Finally, the outputs of these two attention modules were fused to further enhance the feature representations.

2.4. AI Model Acceleration on Mobile Devices

Neural networks of high accuracy usually require massive memory bandwidth and computational resource when running, which makes them difficult to be deployed on embedded systems because of limited hardware resources. Therefore, in order to run such applications on mobile devices, the designed network model needs to optimize inference cost in advance before deploying it to mobile devices. The optimization techniques include pruning [30], a process of removing weight connections to increase inference speed and decrease the model size; quantization [13], performing computation and storing tensors in lower bit-widths rather than in floating points; and binarization [17], using binary

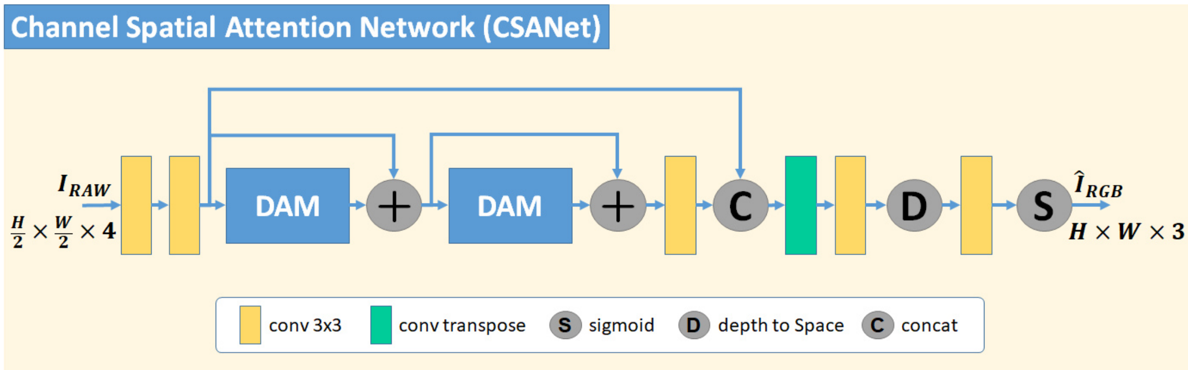

Figure 2. Our proposed Channel Spatial Attention Network (CSANet).

weights to replace floating ones to largely save the storage space and computation power. In the process of optimization, the network needs to be iteratively fine-tuned to minimize the accuracy loss as much as possible. Thus, when designing a network running on a mobile device for an image challenge, there is always a trade-off consideration between the model accuracy and running speed. One practical strategy is to take the advantage of using a floating-point model for inference rather than a quantized one. The reason is that no additional conversion or retraining of the model is needed [9, 10]. Moreover, the accuracy we get in the server environment will be the same as in the mobile device. However, for the consideration of a mobile device's running speed, we take measures to carefully control the network architecture and computing operators so that the runtime of the model will not exceed our predefined limit.

3. Network Architecture

To restore RGB images from camera sensor outputs, a novel network architecture with emphases on inference speed and high PSNR, which we call CSANet, is illustrated in Fig. 2.

3.1. Channel Spatial Attention Network (CSANet)

In order to reduce the computation time and the total training parameters, gradual down-sampling of the input is first under consideration in the design strategy. A simplified but still well-performed attention module should be applied to boost up the reconstructed image quality. Thus, our design follows the aforementioned three-part architecture design in the previous section. In the beginning part, a strided convolution block and a conventional convolution one each with the activation function relu are used to perform feature extraction and downsize the input RAW data I_{RAW}. After that, a series of processing blocks are cascaded. The middle double attention modules (DAM) with skip connections are mainly designed to enhance the spatial dependencies and to highlight the prominent objects in the feature maps [27, 25]. These skip connections [18, 5] are used not only to avoid the vanishing gradient problem but also to keep the similarities between the learned feature maps from different blocks. Next, the last part of the network uses "convolution transpose" and "depth to space" to upscale the size of the feature maps. Finally, a conventional convolution and a following sigmoid function restore the output RGB image $\hat{I}_{RGB}$.

3.2. Double Attention Module (DAM)

The sub-network structure of DAM is shown in Fig. 3. The structure is inspired by the works of Woo et al. [25]. Given input feature maps that are obtained by applying two convolutions, DAM performs this feature recalibration by using two attention mechanisms: (1) spatial attention (SA), and (2) channel attention (CA). The result of these concatenated attentions is then followed by convolutional layer with filter size 1×1 to yield adaptive feature refinement.

Spatial Attention. This module is designed to learn the spatial dependencies in the feature maps. Specifically, in order to have distant vision over the feature maps, a depth-wise dilated convolution [6, 26] is used to extract information. The kernel size is set to 5×5 and the dilated rate is set to 2. After this layer follows a sigmoid activation function to produce pixel-wise attention $z' \in R^{\frac{H}{4} \times \frac{W}{4} \times C}$. Finally, the output $F_{sa} \in R^{\frac{H}{4} \times \frac{W}{4} \times C}$ of the spatial attention module will be the elemental-wise multiplication of the input feature maps F_{in} and the pixel-wise attention z'.

Channel Attention. This module originated from the SENet [27, 25, 7]. It utilizes squeeze and excite operations to learn the inter-channel relationship of feature maps given an input image. The squeeze operation is realized by computing the mean values over the individual feature maps, thus yielding a descriptor in $z \in R^{1 \times 1 \times C}$. The excite operation is composed of two 1×1 convolution layers but each with different channel sizes and activation functions, relu and sigmoid, respectively. This excite operation re-calibrates the squeeze output and produces a

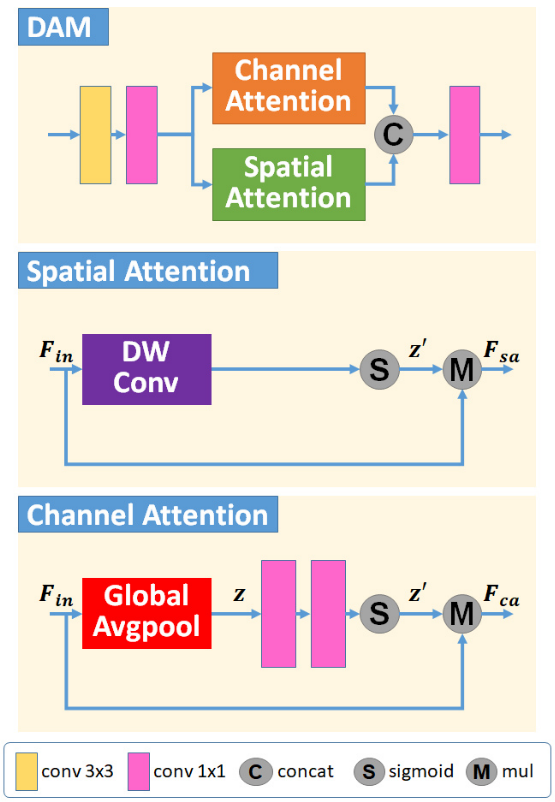

Figure 3. The structure of double attention module (DAM), spatial attention and channel attention.

calibrated descriptor $z' \in R^{1\times1\times C}$. Finally, the output $F_{ca} \in R^{\frac{H}{4}\times\frac{W}{4}\times C}$ of the channel attention module will be the elemental-wise multiplication of the input F_{in} of the squeeze operation and the calibrated descriptor z'.

3.3. Loss Function

In this section, we introduce our loss function that sums up pixel loss, perception loss and, structure similarity loss. We denote $\hat{I}$ as the predicted image and I as the ground truth RGB image.

Pixel Loss. The Charbonnier [29, 1] loss is adopted as an approximate loss function. This loss has been believed to outperform the traditional penalty [29] in image reconstruction tasks. The Charbonnier loss function is defined as:

$$L_{char} = \sqrt{(I - \hat{I})^2 + \varepsilon} \tag{1}$$

where ε is set to 10^{-6}.

Perceptual Loss. To deal with the pixel misalignment problem, the perceptual loss from the output of the pre-trained VGG-19 network [19] is employed. The loss function is defined as:

$$L_p = L_{MSE}(F_{VGG}(I) - F_{VGG}(\hat{I})) \tag{2}$$

where F_{VGG} denotes the output of the last convolution in the pretrained VGG-19 network. This L_{MSE} loss on such feature maps is used to minimize the perceptual difference between the reconstructed image and the ground truth.

SSIM Loss. The structural similarity loss L_{SSIM} [24] is used to enhance the reconstructed RGB images by the structural similarity index. The loss function can be defined as:

$$L_{SSIM} = 1 - F_{SSIM}(I, \hat{I}) \tag{3}$$

where F_{SSIM} calculates the structural similarity index.

Finally, the total loss is expressed as:

$$L_{total} = L_{char} + \alpha L_p + \beta L_{SSIM} \tag{4}$$

where α and β are set to 0.001 and 0.1, respectively.

4. Experiment

4.1. Experimental environment

We used Tensorflow 1.15.0 and python 3.6 to implement the proposed neural network and then trained the model with the server environment (Ubuntu 16.4, Intel Xeon CPU E5-2650 v4, 512G Ram, and Tesla P100 16G GPU x1).

4.2. Datasets

The data set we used was provided by Mobile AI 2021 workshop for the online contest. According to the organization, to get real data for the RAW-to-RGB mapping problem, a large-scale dataset consisting of photos collected using the Sony IMX586 Quad Bayer RGB mobile sensor for capturing RAW photos and a professional high-end Fujifilm GFX100 camera for RGB ground truths was obtained. Since the captured RAW-RGB image pairs are not perfectly aligned, they were matched using an advanced deep learning-based algorithm, and then smaller patches of size 256×256 pixels were extracted. We were provided with 24K training RAW-RGB image pairs (of size $256 \times 256 \times 1$ and $256 \times 256 \times 3$, respectively). It should be mentioned that all alignment operations were performed only on RGB DSLR images, therefore RAW photos from the Sony sensor remained unmodified. We divided the dataset into:

- **Train data**: A random selection 90% of the 24K aligned RAW-RGB image pairs.
- **Self-validation data**: The other 10% of the 24K aligned RAW-RGB image pairs.
- **Validation data**: The participants received the RAW images when the validation phase started; the corresponding ground truth RGB images were released when the final phase of the challenge started.
- **Test data**: The participants could not receive the RAW testing images.

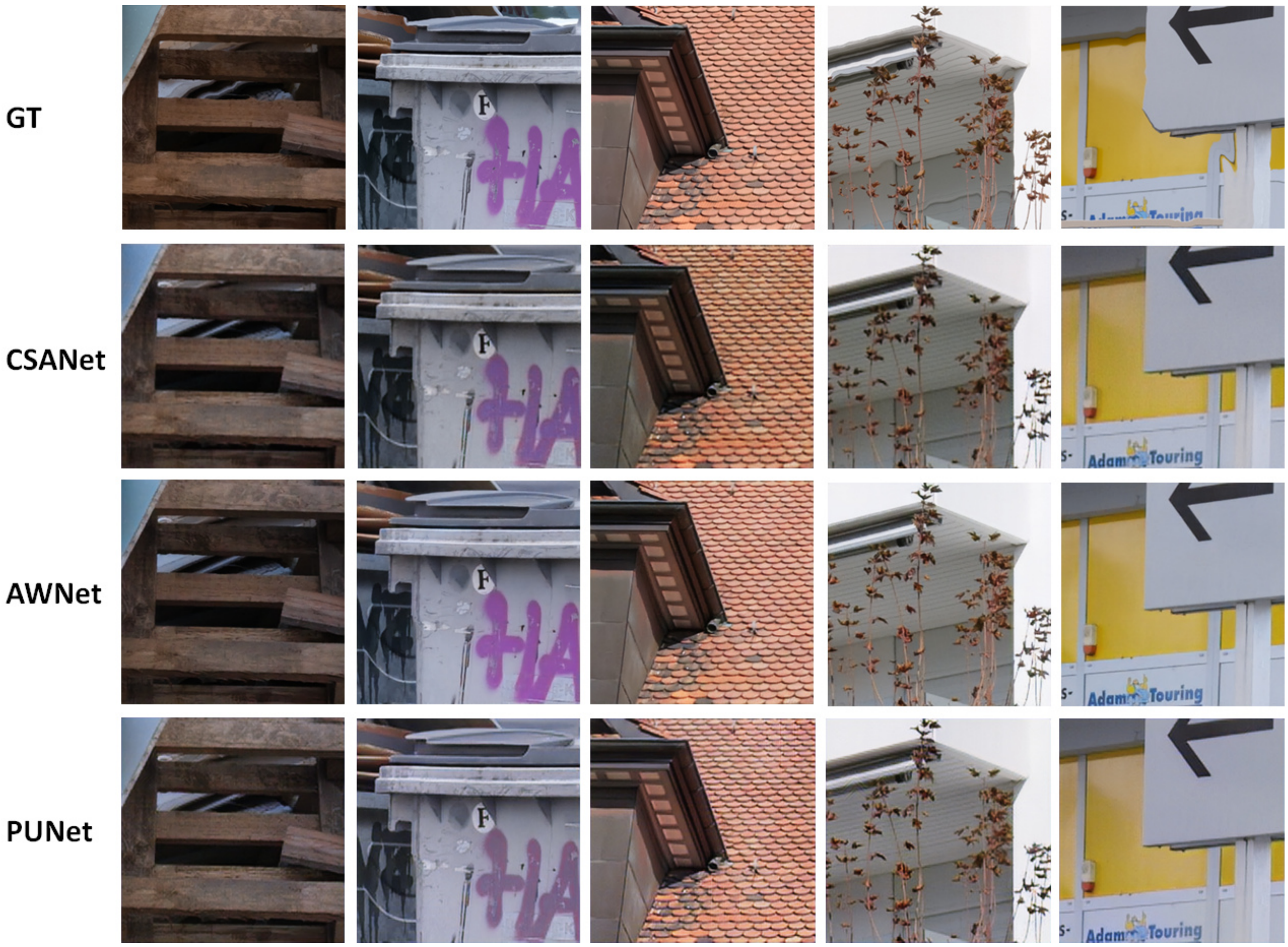

Figure 4. Qualitative comparisons of different networks. From top to down, the first row is the ground truth images captured by Fujifilm GFX100 camera; and the following rows are the reconstructed RGB images of our CSANet, AWNet, and PUNet.

4.3. Training Details

Our model was trained from scratch with $1 \times$ 16G Tesla P100 GPU, taking about 3 days. During the training, all the training images were augmented by random horizontal flipping, and the batch size was set to 100. The weights of the model were trained for 100K iterations using Adam optimizer with an initial learning rate of $5 \times e^{-4}$ which would later be set to 1×10^{-4}, 5×10^{-5}, and 1×10^{-5} at the 20kth, 50kth, and 80kth iteration, respectively. In this work, we only use floating point computation to generate the RGB images. In the final result, our model inferred at 82.8 ms per image and achieved PSNR 24.31 dB on Codalab during the development phase(using the validation set).

4.4. Performance Comparison

To evaluate the performance of our model, we conducted an experiment and compared results with other popular models' (AWNet and PUNet). PUNet was the baseline model provided by Mobile AI 2021 workshop, which is mainly based on PyNet. Our proposed method was tested on the online validation data that was provided during the development stage. The quantitative comparison was shown in Table 1. As can be seen from it, our model not only is capable of generating images with quality as good as others but also infers with a significantly shorter runtime. Fig. 4

Network	PSNR	SSIM	Runtime (ms)
CSANet	24.31	0.84	82.8
AWNet	24.78	0.87	N/A
PUNet	22.74	0.82	200.0

Table 1. Validation scores by different models (using the validation set). All models were trained with the same dataset ad run on MediaTek Dimensity 1000+ (APU). The runtime of AWNet was not available, but run approximately 2 seconds on GPU (Tesla P100).

NN Architecture	PSNR/SSIM	Runtime (ms)
DAM *2 (this work)	24.31 / 0.843	82.8
DAM *1	24.13 / 0.835	74.5
DAM *1 (Only CA)	23.70 / 0.818	71.8
ResBlocks * 4	23.80 / 0.834	73.5

Table 2. The result of the ablation test. These variants are trained under the same condition. We can see that one channel attention module performs as well as four residual blocks. Furthermore, using both channel and spatial attention modules gives an even better PSNR score at a reasonable cost of runtime.

Model	SoC	CPU (ms)	GPU (ms)	NNAPI (ms)
Realme x7 pro	Dim. 1000+	138	228	150
HTC U12+	Snap. 845	280	513	1624
Nokia 9	Snap. 845	238	439	820
Google Pixel5	Snap. 765G	282	827	328
Samsung S10	Exynos 9820	244	299	933

Table 3. The runtime of our proposed model measured by AI benchmark 4.0. The abbreviation Dim. stands for the Dimensity series, and the Snap. stands for Snapdragon series. We can see that CSANet runs mush faster on newer generation SoCs. However, for NNAPI and GPU parts, it didn't perform as well as we expected.

ID	PSNR/SSIM	Runtime (s)	Score
838363	23.20 / 0.8467	0.0610	25.98
838650	**23.73 / 0.8487**	**0.0908**	**25.91**
838466	23.30 / 0.8395	0.0780	25.74
838312	22.97 / 0.8392	0.0650	25.67
838424	22.78 / 0.8472	0.0770	25.24
837988	23.08 / 0.8237	0.0945	25.19
838514	22.03 / 0.8217	0.0763	24.50
838604	22.84 / 0.8379	0.1672	23.50
838328	23.41 / 0.8534	0.2310	23.39
838698	23.23 / 0.8481	1.8610	22.40
836753	19.11 / 0.7987	ERROR	ERROR
836795	8.45 / 0.2274	ERROR	ERROR

Table 4. The results of Mobile AI 2021 Learned Smartphone ISP Challenge. Our result is shown in Boldface (All teams used the same test data from Mobile AI 2021 workshop). Our method achieved the best image quality while remaining competitive on runtime.

shows the reconstructed images of each model. For a more detailed comparison, our method has a better capability of recovering color into RGB space in a pixel-to-pixel matter, as the expected functionality of the double attention modules. However, our proposed method tends to obscure image details a little. Although lacking direct experimental evidence, we think this might result from the steep shrinkage in the size of feature maps in the first extracting part of the network. It is also interesting to point out that, on some occasions, all ISP models tend to "fix" the input RAW image. For example, this phenomenon happened in the images of column 4 and column 5 (from left to right direction). With a close look, we can see that, in the 4th ground truth image, there is a curvy wire stick on the wall. However, all models "fixed" this curve to a straight one. For another example, all models made more changes to the 5th input image. We can see that, the "Adam Touring" sign in the original image is partially blocked by the armrest, and the "arrow" sign has a rounded corner. However, all models "sharpened" the corner, "deleted" the armrest from the picture, and "fixed" the missing part of the alphabets. This behavior is likely caused by the fact that the models learned these similar patterns from the training dataset and considered the original patterns polluted by noise. Therefore, they tend to modify image contents to lower their loss functions when encountering such rare image patterns. For the purpose of developing ISP substitution, this unwanted outcome might be a downside that needs further improvement. However, this also shines a new light on other possible applications (etc. image fidelity) on mobile devices.

4.5. Ablation and AI Benchmark

This section reports the ablation study of the proposed model and the AI Benchmarks for our model in several mobile devices. The results of the ablation study are presented in Table 2. In this study, we compared CSANet with its 4 variants which were trained in the same way as before and were tested on the validation dataset from AIM 2021 Learned Smartphone ISP Challenge. As our baseline, the variant ResBlock * 4 used four 3×3 residual blocks instead of two DAMs. As we can see, one channel attention module has the equivalent performance of four residual block. Moreover, adding an extra spatial attention module boosted performance further around 0.43 db in the PSNR metric comparing to Only CA, while the runtime increases around 1 ms comcaring to ResBlock * 4. Our proposed model thus came from the final decision of balancing the performance and runtime. Additionally, the final part of the proposed model that upscales image sizes is believed to be the bottleneck of the model speed, since changes in our experiment didn't increase the model runtime greatly.

AI Benchmark 4.0 [10] is a mobile software package to measure the neural network performance of a smartphone such as accuracy, speed, initialization time, and so on. Our proposed model was offered to this software package to measure the AI performances on several mobile devices. After providing the path of our tflite model, tests would be conducted to measure the runtimes using CPU, GPU, and

NNAPI separately. The CPU test was set to FP16 and 4 CPU threads. The GPU test was set to FP32. The NNAPI test was set to FP16. Table 3 summarizes the detailed results. We can see that the smart phone with Soc Dimensty 1000+ overwhelmingly beat the rest in all the tests due to its enhancement in AI aspect (with a Device AI-Score of 130.9). However, it seems that for running CSANet, NNAPI and GPU have no lesser runtime than CPU across all mobile devices.

4.6. Contest Performance

Table 4 shows the result of the Mobile AI 2021 Learned Smartphone ISP Challenge. We were ranked 2^{nd} place with the highest image quality (PSNR/SSIM) and a formidable runtime.

5. Conclusion

In this paper, we proposed CSANet, a DNN architecture that utilizes spatial and channel attention modules to model a mobile device's ISP pipeline. Our proposed method generates images with quality as good as AWNet does but with a significantly lower runtime. Moreover, our proposed method won 2^{nd} place in the Mobile AI 2021 Learned Smartphone ISP Challenge.

References

[1] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International journal of computer vision*, 61(3):211–231, 2005.

[2] Linhui Dai, Xiaohong Liu, Chengqi Li, and Jun Chen. Awnet: Attentive wavelet network for image isp. *arXiv preprint arXiv:2008.09228*, 2020.

[3] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[4] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[7] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[8] Andrey Ignatov, Jimmy Chiang, Hsien-Kai Kuo, Anastasia Sycheva, and Radu Timofte. Learned smartphone isp on mobile npus with deep learning, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2021.

[9] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. Ai benchmark: Running deep neural networks on android smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[10] Andrey Ignatov, Radu Timofte, Andrei Kulik, Seungsoo Yang, Ke Wang, Felix Baum, Max Wu, Lirong Xu, and Luc Van Gool. Ai benchmark: All about deep learning on smartphones in 2019. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3617–3635. IEEE, 2019.

[11] Andrey Ignatov, Radu Timofte, Zhilu Zhang, Ming Liu, Haolin Wang, Wangmeng Zuo, Jiawei Zhang, Ruimao Zhang, Zhanglin Peng, Sijie Ren, et al. Aim 2020 challenge on learned image signal processing pipeline. *arXiv preprint arXiv:2011.04994*, 2020.

[12] Andrey Ignatov, Luc Van Gool, and Radu Timofte. Replacing mobile camera isp with a single deep learning model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 536–537, 2020.

[13] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.

[14] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. *arXiv preprint arXiv:2009.11551*, 2020.

[15] Jie Liu, Wenjie Zhang, Yuting Tang, Jie Tang, and Gangshan Wu. Residual feature aggregation network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2359–2368, 2020.

[16] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 773–782, 2018.

[17] Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe. Binary neural networks: A survey. *Pattern Recognition*, 105:107281, 2020.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[21] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017.

[22] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[23] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[24] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[25] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[26] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[27] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2696–2705, 2020.

[28] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.

[29] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018.

[30] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.

[31] Yu Zhu, Zhenyu Guo, Tian Liang, Xiangyu He, Chenghua Li, Cong Leng, Bo Jiang, Yifan Zhang, and Jian Cheng. Eednet: enhanced encoder-decoder network for autoisp. In *European Conference on Computer Vision*, pages 171–184. Springer, 2020.