LAN: Lightweight Attention-based Network for RAW-to-RGB Smartphone Image Processing

Daniel Wirzberger Raimundo
ETH Zürich, Switzerland
wirdanie@ethz.ch

Andrey Ignatov
ETH Zürich, Switzerland
andrey@vision.ee.ethz.ch

Radu Timofte
University of Würzburg, Germany and ETH Zürich, Switzerland
radu.timofte@uni-wuerzburg.de

Abstract

The number of pictures taken by smartphones is growing exponentially. However, the smartphones' limitations both in size and cost negatively impact on the quality of the implemented sensors. At the same time, their computing power has also been steadily improving, allowing the usage of more complex processing methods to enhance images. In prior works, deep neural networks trained with matched sensor outputs and DSLR images have shown to bring substantial improvements to the images, compared to classical and handcrafted methods. We propose a lightweight attention-based network (LAN) that employs a convolutional layer to learn the input mosaic and an unsupervised pre-training strategy. Our method is validated on standard benchmarks and shown to improve over the state-of-the-art in both perceptual and fidelity terms without hindering GPU inference time on smartphone devices. Our code is available at: github.com/draimundo/LAN

1. Introduction

Smartphone photography has been constantly evolving in the past decades. Initially, the images taken by smartphones were of low quality, and compact cameras dominated the consumer market for digital photography. By 2013, smartphones were outselling digital cameras by a factor of ten-to-one. Nowadays, smartphones are used for most of the stills taken worldwide. While often advertised for similar output resolutions, the quality of shots taken by smartphones still is inferior to the one attained by digital single-lens reflex cameras (DSLR), which generally offer better dynamic range, color accuracy and less digital noise, to name a few.

One disadvantage of smartphones, compared to DSLR cameras, is that due to their compactness, the integrated lens and sensor systems are of smaller size, leading to a poorer signal-to noise ratio (SNR) and other undesirable physical effects on the (unprocessed) image. On the other hand, the computational power of smartphones has been on a steady rise, allowing the use of more powerful methods to compensate for hardware limitations.

Most of the digital sensors embedded in cameras are based on the Bayer filter mosaic, which is a color filter array (CFA) superimposed on the digital image sensor, making specific pixels more sensitive to certain wavelengths (mainly green, red, blue). This allows imagers to capture the color information of a scene, and output a RAW image. However, this spatial separation of colors requires a reconstruction step, called demosaicing, to obtain an image containing complete color information at each pixel position and a final RGB image.

In classical methods, this demosaicing step is a part of the image signal processing (ISP) system [28], that additionally alleviates the effects of sensor noise, adjusts the color balance, and improves the overall image quality. Even so, the latter is limited by the sensor characteristics, and the added processing often trades noise for a lack of details in the result given to the end-user.

In learned ISP, a deep learning model is trained to reproduce high-quality images taken with a DSLR camera from the lower-quality RAW output of a smartphone sensor, showing substantial improvement over the default smartphone output. We build upon previous results and show that using a strided convolutional layer to learn the input mosaic greatly improves sharpness without increasing the mobile inference time by a significant amount. Furthermore, we propose an unsupervised method to pre-train the network on classically demosaiced images.

(a) Smartphone output (b) DSLR reference

(c) CSANet [10] (d) LAN (ours)

Figure 1. Quality and alignment of the different outputs. Best viewed zoomed-in on an electronic version.

As an introductory example, Fig. 1 shows a comparison between the different outputs encountered in this work. In Fig. 1a, the output produced by the smartphone ISP displays good contrast, but the details are exaggerated by a watercolor effect, which also washes out the tones. Fig. 1b depicts the same scene taken by the DSLR camera, used as a ground truth during training. Finally, Figs. 1c and 1d show the result obtained with CSANet [10], the highest-quality solution at the Mobile AI 2021 learned smartphone ISP challenge (MAI21) [13], and the proposed solution. Notice the differences in color balance and detail (especially visible on the blinds in the background).

2. Related Work

The usage of machine learning for RAW-to-RGB image mapping has been increasingly popular in the last years, which is also related to the recent boost in the computational power of smartphones. So far, research has been focusing on two main objectives: finding a good network design and training process to increase image quality, and adapting the network to the computational constraints of smartphones.

Smartphone Image Enhancement was introduced with convolutional neural networks (CNNs) mapping the RGB output of the smartphone to an enhanced RGB version, by training on smartphone-DSLR image pairs, as in DPED [14]. A first RAW-to-RGB dataset, Zürich RAW-to-RGB (ZRR) was introduced for the AIM2019 challenge [15], which PyNET [17] built upon to obtain state-of-the-art results by using a powerful multi-scale network. At AIM2020 [16], the focus was still on image quality. Recently, attention was also given to lightweight models, notably at the MAI21 smartphone ISP challenge [13], where scoring also included the mobile runtimes.

Joint Demosaicing and Denoising started by using classical methods, showing that combining both processes brought superior performance [9], compared to solutions treating the problem sequentially (first demosaicing the RAW input into an RGB image, which then gets denoised). The creation of large-scale (partly synthetic) datasets then enabled the usage of CNNs, to improve results further [7]. Because noise is especially visible in low-light conditions, Chen *et al*. [3] introduced a paired dataset, consisting of noisy underexposed images still in the RAW format, and reference RGB images taken with longer exposure, as well as a CNN for recovering high-quality RGB images from corrupted RAW inputs.

Single Image Super-Resolution (SISR) aims to recover details in a downsampled RGB image, using a model trained on low- and high-resolution RGB image pairs. Initially, CNNs were shown to outperform classical methods [4]. Afterwards, adaptations to the architecture and increasing the model capacity [20, 25, 29] lead to progressively better results. Furthermore, the usage of more complex loss functions instead of the mean-square-error (MSE) [40], as well as training a discriminator in parallel, as done for generative adversarial networks (GANs) [22] have shown to improve the output quality even further.

Multi-Frame Super-Resolution (MFSR) combines multiple input frames to increase the output resolution. This input, combined with classical processing methods, is already in used on flagship phones to reduce the noise level in zoomed-in and low-luminosity contexts [35]. For this task, CNNs have also shown promising results [1]. The drawback to these methods is that they rely on a high number of input frames (10-20), and therefore long acquisition times to get the best results.

High Dynamic Range imaging (HDR) aims to produce outputs with an improved dynamic range compared to the input, which can consist of a single image [5, 6, 27] or multiple shots [19, 27, 36, 37]. The models are trained with one synthetically generated side (low- or HDR), and in the multi-exposure case have to deal with movements between the input frames.

The *Perception-Distortion Tradeoff* is an observation that often, models obtaining good numerical results, seem to have a perceptually (for the human viewer) lower quality [2]. Multiple metrics have been introduced to somewhat compensate for this effect, the most used ones being the VGG loss [18], derived from the VGG image classification network [30] and LPIPS [39].

(a) RAW input (equalized for readability) (b) Warped RGB DSLR target (c) AMAZE demosaiced RAW target

Figure 2. Example of input patch, warped training ground truth, and ground truth obtained by classical demosaicing

2.1. Paired RAW-RGB Dataset

The *MAI2021 dataset* [13] was generated using a Sony IMX586 quad Bayer mobile sensor, and a Fujifilm GFX100 DSLR. The images were adjusted by first converting the RAW images to RGB using a classical algorithm, and then using PDC-Net [32] to warp the RGB high-quality objective to the demosaiced input. This not only tries to compensate for the possible physical misalignment of the cameras during image collection, but also for the different sensor characteristics (e.g sensor size, focal length) which change structural characteristics of the image. Differences can be seen between Fig. 1b (unprocessed DSLR output) and Fig. 1a where the branches on the left side do not intersect the window borders at the same height. While this method generally brings good results, Fig. 2a shows a selected input from the MAI2021 dataset, with the processed RGB high-quality target displayed in 2b showing severe warping. Fig. 2c displays the output obtained by using the classical AMAZE demosaicing method [26] on the input RAW image, and shows a higher level of noise, but no warping or alignment issues. The warping seems to occur mainly on images containing repeating patterns or large flat surfaces, likely by wrong keypoint matching.

2.2. Baseline

CSANet [10] was introduced during the MAI2021 challenge [13], and was the solution with the highest PSNR ($0.43dB$ above all other solutions), best SSIM, and a moderate runtime allowing it to place second overall. The original architecture starts by separating the color channels with a space-to-depth transformation, downsamples the input using a strided convolution and then uses two double attention modules (DAMs) to learn spatial and channel dependencies, finally upscaling the image to the desired resolution by first using a transposed convolution and then a depth-to-space transformation. At the lowest scale, skip-connections are used to improve trainability.

The DAMs, shown in Fig. 4a, are inspired by the convolutional block attention module (CBAM) [34], which combines both spatial and channel attention, and was shown to significantly improve the results on classification tasks.

Channel Attention first uses global average pooling on each feature map, and then uses a bottleneck convolutional layer with decreased channel number and a final dimensionality increasing layer to restore the channel dimension, as displayed in Fig. 4b. The result of this operation weighs the input tensor channel-wise, as introduced in SENet [11].

Spatial Attention, illustrated in Fig. 4c, bases on a depth-wise $2-$dilated 5×5-convolution to increase the receptive field, without growing the complexity as much as a standard convolution (as used in CBAM). The result of this operation is multiplied with the input tensor, which highlights specific regions.

3. Proposed Method

We build upon the baseline and add the modifications proposed in the upcoming Sections. The network architecture is visually detailed in Figs. 3 and 4.

3.1. Learned Demosaicing

In most RAW-to-RGB mapping CNNs, the Bayer mosaic encountered in the RAW input image is removed by stacking each 2x2 block in the original image in 4 input channels. This ensures translational color invariance in each channel, and biases the input interpretation towards a color separation. On the other hand, the resulting 4-channel input has a spatial misalignement between its channels, as in the original RAW input, every pixel carries information about a separate spatial location. One advantage of using the stacked method is that for a fixed kernel size, the receptive field of the first layer is doubled compared to the flat input, as encountered in dilated convolutions [38].

Classical demosaicing methods often explicitly use the correlation of the luminance information between differently colored yet neighboring pixels, by interpolation over larger regions [8]. Basing on this concept, an alternative is to use no stacking at the input, but a stride of 2 for the first filter which gets convolved with the RAW image, to ensure the colors can be learned by the network. As the stride divides the number of operations by 4, the computational cost is not increased drastically by processing at full-scale. Furthermore, removing the slow space-to-depth operation [31] further reduces the latency drawback when using mobile GPU inference.

3.2. High-Level Skip Connection

By using learned demosaicing, the span of the feature space after the first dimensionality reduction is increased. This allows using the resulting feature representation in a skip connection, to circumvent downsampling high-frequency detail. However, to avoid forcing input noise into the output, it is added by concatenation instead

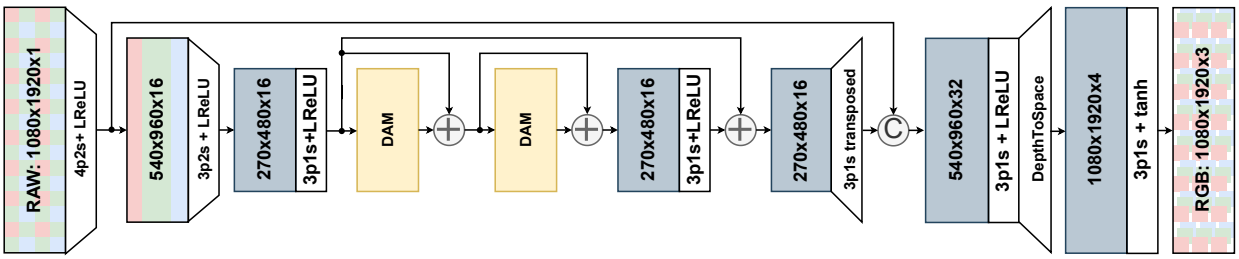
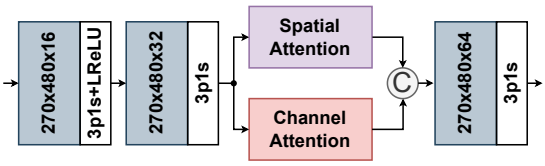
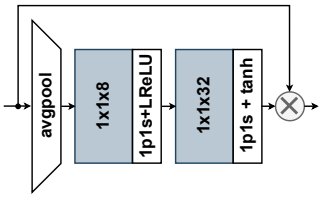


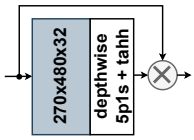
Figure 3. LAN model. $XpYs$ indicates a convolutional layer with kernel size X and stride Y. Skip connections are done either by addition or concatenation, illustrated by a circled + or C symbol, respectively. Details to the DAM blocks are in Figure 4.



(a) Double Attention Module (DAM)



(b) Channel Attention Block



(c) Spatial Attention Block

Figure 4. Submodule details. $XpYs$ indicates a convolutional layer with kernel size X and stride Y. C indicates a channel concatenation, and $\times$ a layer-wise multiplication.

of addition. Unlike CSANet, the final skip connection at the lowest resolution is done by addition, to compensate part of the computational complexity added by the high-level modification.

3.3. Classical Pre-Training

As mentioned in Sec. 2.1, the alignment of training RAW inputs with DSLR images is not trivial, and often not perfect. Even though advanced processing methods can somewhat improve the results, the intrinsic physical sensor parameters limit perfect matches, and processing can introduce further artifacts, shown in Fig. 2.

Although classical joint demosaicing and denoising methods do not reach state-of-the-art anymore, they only rely on input images (and generally some user fine-tuning). This can be used to pre-train the network: instead of starting with a random mapping, the network is first trained using input RAW images and their classically demosaiced RGB equivalent for a certain number of epochs. In the main training step, the network is fed low-quality RAW images and DSLR RGB images, and ideally only needs to learn the new color space and denoising step.

Additionally, this pre-training can be done in an unsupervised way, allowing to easily expand the amount of training examples fed to the network overall, which usually leads to better performance.

3.4. Loss Function

We use a combination of the different loss categories detailed in the upcoming paragraphs to build the loss function in Eq. (1).

Pixel losses: the Mean-Squared-Error (MSE or ℓ_2) loss is the most popular loss in super-resolution tasks. A commonly used substitute is the Mean-Absolute-Error (MAE or ℓ_1) [40]. Alternatively, the Huber loss behaves as ℓ_2 for small errors and ℓ_1 for larger errors, making it theoretically less sensitive to outliers.

Structural losses: the structural-similarity index measure (SSIM) works on grayscale images, and bases on the hypothesis that the human visual system (HVS) is more sensitive to texture, which consists the pixel and its surroundings, than absolute values. An extension of SSIM, called multi-scale SSIM (MS-SSIM), also computes SSIM on downsampled versions of the input and was shown to provide better results than the original implementation [33].

Perceptual losses: perceptual losses, which are based on perceptual similarity instead of similarity in pixel space, were introduced with SRGAN [23], and showed improved opinion scores over other loss functions. SRGAN based on using the feature maps produced by the 5th convolution, before the 4th maxpooling layer in the pretrained VGG19 [30] network as transformations into the perceptual space. The Euclidean distance between the feature representation of the reconstructed image and the reference image was then used as a loss function. More recently, the learned perceptual image patch similarity (LPIPS) metric [39] was introduced, which bases on the VGG [30], AlexNet [21] and SqueezeNet [12] classification networks to produce a score. Using smaller networks such as AlexNet or SqueezeNet makes the loss backpropagation computationally easier, and allows for increased batch sizes.

Color losses: the previously mentioned loss functions generally push the network to high PSNR and SSIM scores,

the most popular full-reference metrics to evaluate image reconstruction. However, some networks attaining high scores can show some color deviation, as even small changes in the RGB space can lead to big perceptual color differences. To compensate for this, DPED [14] proposed to compute the MSE between blurred output and target images, to minimize the effect of textures, and make the loss more global. Alternatively, we propose first blurring the image using the same Gaussian blur operator with variances $\sigma_{x,y} = 3$, then using linear transformations to map from the RGB to the Y'UV space, before computing the MSE between the UV (chrominance) channels of the enhanced and reference image.

Noting the reconstructed image as $\tilde{x}$ and the ground truth as y, we use the following loss function (with $\overline{z}_{[U,V]}$ corresponding to the U and V channels of the Gaussian blurred version of z):

$$\begin{aligned} l(\tilde{x}, y) = & 300 \cdot l_{Huber}(\tilde{x}, y) \\ & +100 \cdot (\overline{\tilde{x}}_{[U,V]} - \overline{y}_{[U,V]})^2 \\ & +30 \cdot \big(1 - \text{MS-SSIM}(\tilde{x}, y)\big) \\ & +10 \cdot \text{LPIPS}_{Alex}(\tilde{x}, y) \end{aligned} \tag{1}$$

which is a weighed sum of the Huber, Color, MS-SSIM and LPIPS losses. The coefficients approximately set the magnitude of each term to 1 for a fully trained model, on the validation set.

3.5. Activation Functions

For LAN, we propose changing the $ReLU$ activations used in the baseline to $LeakyReLU$ (with 0.2 leakage coefficient), and use a scaled $tanh$ ($f(x) = 0.58 \cdot tanh(x) + 0.5$) instead of the $sigmoid$ activation, as in DPED [14]. The latter allows the network to reach the extremes of the $[0, 1]$ pixel value range without needing to use large coefficients.

4. Experiments

Training was performed on an Nvidia Titan Xp GPU, with a batch size of 50. The network parameters were optimized using the Rectified Adam (RAdam) [24] algorithm. When pre-training is used, it is done for $200k$ iterations, with a constant learning rate of 10^{-4}. Afterwards, the network is trained on RAW-RGB pairs for another $200k$ iterations, with the same learning rate, and during a fine tuning step of $50k$ iterations with a learning rate of 10^{-5}.

Inference was done using a 16-bit floating point format, to mimic to the 10 to 14-bit depth of the most common Bayer RAW sensors. Inference times were evaluated using the TFLite models of the networks, on a MediaTek Dimensity 1000+ APU, specifically the ARM Mali-G77 MC9 GPU, as the runtimes were faster by a factor of 4 to 10 compared to CPU or NNAPI inference. Inference output resolution is kept to Full HD (1920×1088) as in the MAI21 challenge.

4.1. Dataset

The MAI21 dataset is used, with the original splits consisting of $93k$ training image pairs, $2.2k$ validation image pairs and $3.1k$ testing image pairs.

Additionally, the RAW images of the validation pairs were also demosaiced as described in Sec. 2.1, to be used as aligned pretraining images, proposed in Sec. 3.3. The advantages of this method could be more significant if the RAW patches were part of another (possibly without DSLR ground truth) training split.

4.2. Ablation Study

We perform an ablation study by analysing the effect of removing the different modifications on pixel, structural and perceptual metrics, and compare them to the original CSANet model. The results are reported in Tab. 1. The results for all variants except the original CSANet were obtained by using the loss function introduced in Eq. (1), and the training procedure described in Sec. 4. The CSANet models were trained using the original loss function and training method. We add a classically pre-trained version to analyze performance improvements.

All LAN variants show a significantly higher SSIM score, indicating a better reproduction of textures, which could be attributed to the spatial alignment between layers of the filters at the beginning of the network. This is particularly visible for high-frequency patterns, as for example the garage door in Fig. 5.

Adding a pre-training step increases the PSNR noticeably, both for the CSANet and LAN architectures. This could be related to the ideal alignment between input and ground truth, and that in the second step, the network only needs to adjust to the color space of the DSLR images, learn removing sensor noise and interpolate for high-frequency data, instead of learning the complete pipeline immediately.

Finally, the LAN model, which uses all the proposed modifications, shows the best numerical and also visual results, discussed in the upcoming Section.

Architecture	PSNR$[dB]\uparrow$	SSIM$[1]\uparrow$	$LPIPS_{VGG}[1]\downarrow$
LAN (*ours*)	**24.29**	**0.882**	**0.2263**
LAN-HS	24.15	0.875	0.2434
LAN-HS-PT	23.87	0.871	0.2627
CSANet+PT	23.86	0.859	0.2618
CSANet [10]	23.73	0.849	0.2552

Table 1. Results of the ablation study. Modifications: PT: *Pre-Training*; HS: *High-level Skip Connection.*

Model	PSNR$[dB]\uparrow$	SSIM $[1]\uparrow$	Size $[KB]\downarrow$	Latency$[ms]\downarrow$	Final Score$[1]\uparrow$
LAN (*ours*)	**24.29**	**0.882**	197	80.6	**26.68**
dh_isp	23.20	0.847	21	**61**	25.98
CSANet [10]	23.73	0.849	123	90.8	25.91
ENERZAi Research	22.97	0.839	**9**	65	25.67
isp_forever	22.78	0.847	175	77	25.24
NOAHCTV	23.09	0.824	244	94.5	25.19

Table 2. Comparison with the best results of the MAI21 smartphone ISP challenge [13]. The latency values are computed for a Full HD (1920×1088) image, on the ARM Mali-G77 MC9 GPU.

(a) CSANet [10] (b) LAN (*ours*)

Figure 5. Effects of Learned Demosaicing.

4.3. Performance Evaluation

We compare the proposed solution to the 5 best competitors at the MAI21 smarthphone ISP challenge [13], where the final score (FS) was computed using the original score function, in Eq. (2).

$$\begin{aligned} \text{FS} &= \text{PSNR} + \alpha\Big(0.2 - clip\big(\text{runtime}[s]\big)\Big); \\ clip &= min\Big(max\big(\text{runtime}[s], 0.03\big), 5\Big); \\ \alpha &= \begin{cases} 20, \text{if runtime}[s] \leq 0.2, \\ 0.5, \text{otherwise.} \end{cases} \end{aligned} \tag{2}$$

Tab. 2 shows the quality and performance metrics, as well as the obtained final score. The model size does not correlate well with the latency, due to the variations in architecture and fixed costs. For example, doing convolutions at higher resolutions requires more computations, even if the kernel size (and therefore the memory size of the operation) remains the same. Furthermore, the delays introduced by memory allocation is the same for all the networks, so a latency floor disadvantages very small architectures. Even tough it has an increased number of layers, LAN does not use a GPU-inefficient space-to-depth operation [31], explaining the lower latency of LAN compared to the CSANet baseline.

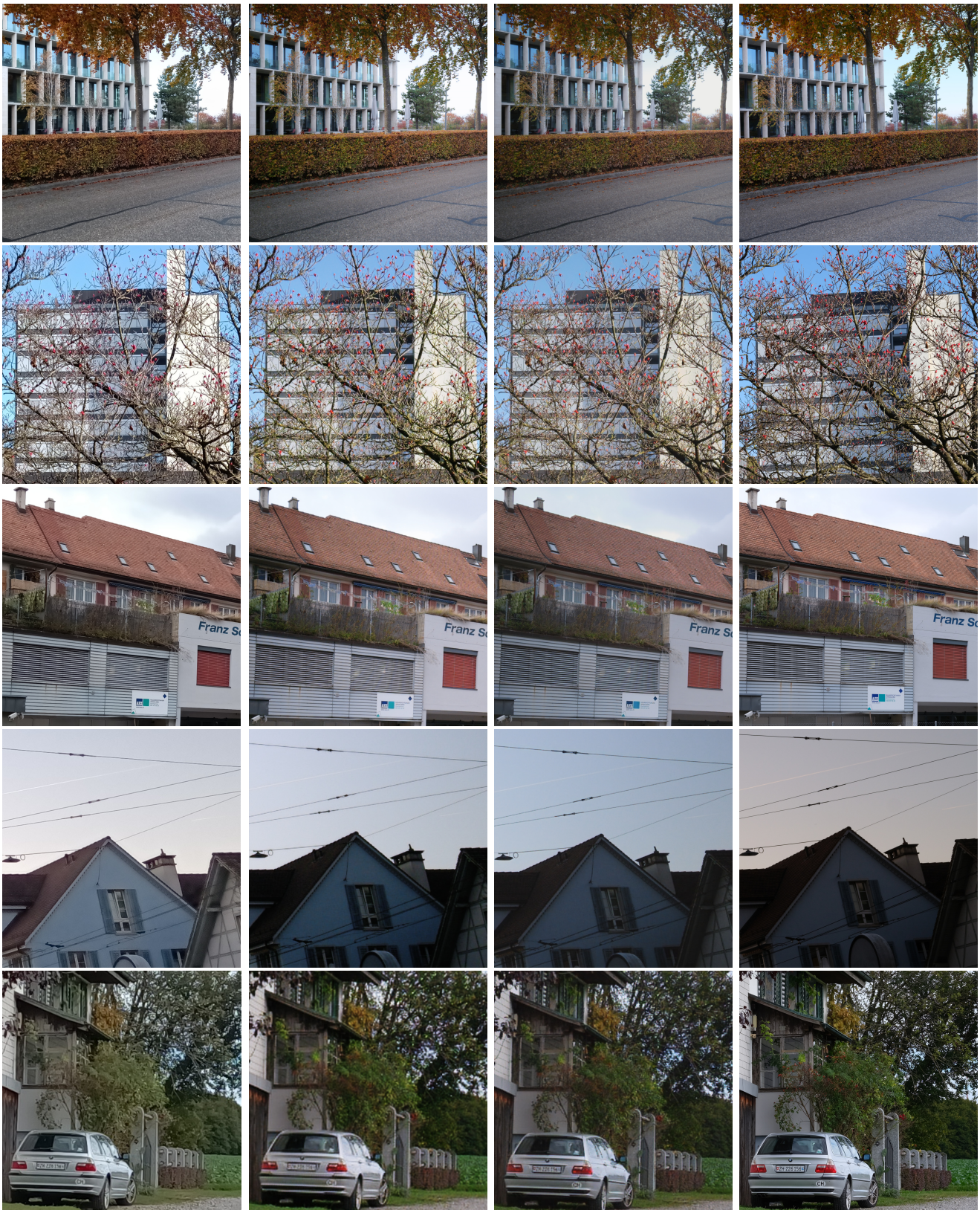
The numerical scores obtained by the LAN model are significantly higher than for all other models. While their PSNR scores vary, the SSIM metrics of the best runner ups are all in the same range, indicating a plateau in structural reproduction accuracy. The increase for LAN can be attributed to the learned demosaicing part, as discussed in the last Section. We now compare some output patches obtained with the Mediatek ISP, CSANet and LAN, displayed in Fig. 6. DSLR images are provided as a comparison basis.

Detail Level: the small texts contained in the 3rd and last rows (company info and number plate) give a good idea of the detail level attained by the different methods. The Mediatek ISP provides the best perceived sharpness of the 3 solutions, but on closer inspection on the company info, the processing alters the text, which makes it harder to interpret than the noisier output produced by LAN. Furthermore, the details on more complex textures such as the tree in the 5th row, are heavily flattened by the embed ISP and almost not recognizable, unlike the methods based on CNNs.

Color Reproduction: as the ISP was not tuned with (the same) DSLR references, absolute color comparison is not sensible. However, contrast can be used (as it is relative), and is clearly lower than in the CNN solutions. This is especially visible on the balcony in the 3rd row, or the plants in the 1st and last rows. The postprocessing of the exposure and detail flattening gives also very unnatural looks to the image, as exemplified by the 4th row. Comparing the two CNN solutions, the differences are not as noticeable, and can also be related to the training methods.

Artifacts: between the CNN solutions, CSANet seems to be more sensitive to sensor noise, which is especially visible on the walls of the house in the 4th row. Furthermore, in the same image, the aliasing of the cables against the sky is more pronounced for this solution. The 5th image also shows more unnatural colors for CSANet. At the same time, both solutions show some residual reconstruction artifacts in the 5th image, most noticeably on car.

Overall, the CNN solutions deliver visually more pleasant images compared the ISP, and LAN provides a significant sharpness boost over CSANet.



(a) Mediatek ISP (b) CSANet [10] (c) LAN (*ours*) (d) DSLR

Figure 6. Visual comparison of the different outputs and ground truths. Best viewed zoomed-in on an electronic version.

5. Conclusion

In this paper, we showed that adding a strided convolutional layer to demosaic the input RAW image, instead of doing the usual space-to-depth conversion allows the network to produce RGB outputs with significantly higher sharpness, which shows visually and in pixel, structural and perceptual metrics. Furthermore, it can be processed faster on mobile GPUs, leading to lower inference times.

Additionally, classically demosaicing the images allows to pre-train the network in an unsupervised way on perfectly aligned images, and can lead to better performance, which we showed empirically on both the baseline and the proposed model.

Finally, we compared LAN to the best solutions at the Mobile AI 2021 Smartphone ISP challenge. While outperforming the PSNR runner-up by $0.56dB$, it is only $20ms$ slower than the fastest solution, leading to the best Final Score among all competitors, by a large margin.

Acknowledgments

This work was supported by ETH Zürich General Fund and by Humboldt Foundation.

References

[1] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deep burst super-resolution. *CVPR*, 2021. 2

[2] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. *CVPR*, 2018. 2

[3] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. *CVPR*, 2018. 2

[4] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE TPAMI*, 38(2):295–307, 2016. 2

[5] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, et al. Hdr image reconstruction from a single exposure using deep cnns. *ACM TOG*, 36(6):1–15, 2017. 2

[6] Yuki Endo, Yoshihiro Kanamori, and Jun Mitani. Deep reverse tone mapping. *ACM TOG*, 36(6):1–10, 2017. 2

[7] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM TOG*, 35(6):1–12, 2016. 2

[8] B.K. Gunturk, J. Glotzbach, Y. Altunbasak, et al. Demosaicking: color filter array interpolation. *IEEE Signal Process. Mag.*, 22(1):44–54, 2005. 3

[9] K. Hirakawa and T.W. Parks. Joint demosaicing and denoising. *ICIP*, 2005. 2

[10] Ming-Chun Hsyu, Chih-Wei Liu, Chao-Hung Chen, et al. Csanet: High speed channel spatial attention network for mobile isp. In *CVPRW*, pages 2486–2493, 2021. 2, 3, 5, 6, 7

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *CVPR*, 2018. 3

[12] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, et al. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016. 4

[13] Andrey Ignatov, Cheng-Ming Chiang, Hsien-Kai Kuo, et al. Learned smartphone isp on mobile npus with deep learning, mobile ai 2021 challenge: Report. *CVPRW*, pages 2503–2514, 2021. 2, 3, 6

[14] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, and Kenneth Vanhoey. Dslr-quality photos on mobile devices with deep convolutional networks. *ICCV*, 2017. 2, 5

[15] Andrey Ignatov, Radu Timofte, Sung-Jea Ko, et al. Aim 2019 challenge on raw to rgb mapping: Methods and results. *ICCVW*, 2019. 2

[16] Andrey Ignatov, Radu Timofte, Zhilu Zhang, et al. Aim 2020 challenge on learned image signal processing pipeline. *ECCVW*, pages 152–170, 01 2020. 2

[17] Andrey Ignatov, Luc Van Gool, and Radu Timofte. Replacing mobile camera isp with a single deep learning model. In *CVPRW*, pages 2275–2285, 2020. 2

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *ECCV*, page 694–711, 2016. 2

[19] Nima Khademi Kalantari and Ravi Ramamoorthi. Deep high dynamic range imaging of dynamic scenes. *ACM TOG*, 36(4):1–12, 2017. 2

[20] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. *CVPR*, 2016. 2

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, volume 25, 2012. 4

[22] Christian Ledig, Lucas Theis, Ferenc Huszar, et al. Photo-realistic single image super-resolution using a generative adversarial network. *CVPR*, 2017. 2

[23] Christian Ledig, Lucas Theis, Ferenc Huszár, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 105–114, 2017. 4

[24] Liyuan Liu, Haoming Jiang, Pengcheng He, et al. On the variance of the adaptive learning rate and beyond. In *ICLR*, 2020. 5

[25] Xiao-Jiao Mao, Chunhua Shen, and Yubin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NeurIPS*, 2016. 2

[26] Emil Martinec and Paul Lee. Amaze demosaic algorithm. *rawtherapee.com*, 2010 [Online]. 3

[27] Eduardo Pérez-Pellitero, Sibi Catley-Chandar, Aleš Leonardis, et al. Ntire 2021 challenge on high dynamic range imaging: Dataset, methods and results. In *CVPRW*, pages 691–700, 2021. 2

[28] R. Ramanath, W.E. Snyder, Y. Yoo, and M.S. Drew. Color image processing pipeline. *IEEE Signal Process. Mag.*, 22(1):34–43, 2005. 1

[29] Wenzhe Shi, Jose Caballero, Ferenc Huszar, et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *CVPR*, 2016. 2

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2, 4

[31] TensorFlow. Tensorflow lite on gpu. *tensorflow.org/lite/performance/gpu_advanced#tips_and_tricks*, 2022 [Online]. 3, 6

[32] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *CVPR*, 2021. 3

[33] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. *ACSSV*, 2003. 4

[34] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. *ECCV*, page 3–19, 2018. 3

[35] Bartlomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, et al. Handheld multi-frame super-resolution. *ACM TOG*, 38(4):1–18, 2019. 2

[36] Shangzhe Wu, Jiarui Xu, Yu-Wing Tai, and Chi-Keung Tang. Deep high dynamic range imaging with large foreground motions. *ECCV*, page 120–135, 2018. 2

[37] Qingsen Yan, Dong Gong, Qinfeng Shi, et al. Attention-guided network for ghost-free high dynamic range imaging. *CVPR*, 2019. 2

[38] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 3

[39] Richard Zhang, Phillip Isola, Alexei A. Efros, et al. The unreasonable effectiveness of deep features as a perceptual metric. *CVPR*, 2018. 2, 4

[40] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Trans. Comput. Imaging*, 3(1):47–57, 2017. 2, 4