Supplemental Material
MetaISP – Exploiting Global Scene Structure for Accurate Multi-Device Color Rendition

1 Training Details

The images are divided into 1463, 186, and 180 respectively for training, validation, and testing for each device during the pre-training. During the effective training the division is 137 for training and 26 for testing.

The same pictures are available for all devices and were pre-aligned as described below. To avoid memory issues, during training, the data is divided into patches. First, the images are cropped to 2688×2688 resolution. Each image is divided into 36 patches with a resolution of 448×448. Resulting in a final dataset size of 65844 for pre-training and 5868 for training.

MetaISP considered as input the iPhone RAW image and, as possible, output the Google Pixel 6 Pro, Samsung S22 Ultra, and the iPhone XR sRGB images. Hence, the RAW from Pixel and S22 is not used, only their metadata. Therefore, when learning the white balance, we can match the specific values of each device. When the white balance is not learned, the metadata is taken from the source device (iPhone), since the meta data of the target model is not available during inference. This was empirically shown to produce better results than not using metadata.

Our model is training for 100 epochs, with a learning rate (LR) of 10^{-4}, ADAM optimizer, and a scheduler decreasing the LR linearly to 0 after half the epochs. During training, we randomly select the device id and its ground truth. Hence, different mobiles are considered in one single minibatch. Horizontal and vertical flips are used as augmentations.

2 Alignment and Pre-processing

After making efforts to capture the aligned data, misalignments are still present due to the grid shape in which the phones were arranged during the capture process. To address this issue, we initially perform rough alignment using the RANSAC [1] algorithm, aligning the RGB images from the Pixel and S22 phones with the RGB image from the iPhone. Consequentially, we align them with the corresponding raw images, as the iPhone RGB and its raw images are already perfectly aligned. However, this may not be sufficient, as it can result in blurry outputs and color inconsistencies during training. To overcome this challenge, we employ optical flow during training to align all the ground truth images with the iPhone RGB image. This technique was initially introduced by Zhang et al. [2], who proposed a module for matching raw image with rgb images color, then warp the original ground truth images to the raw-colored. In our case, we simplify this process by utilizing the iPhone RGB image as the reference for aligning the Pixel and S22 images, eliminating the need for an additional module (see Equation 1. Furthermore, we implement a pixel occlusion technique, as suggested by [3], where pixels exhibiting a mismatch between forward and backward optical flows are occluded.

$$y_d^w = \mathcal{W}(y_2, y_d), \tag{1}$$

Where $d \in \{0, 1\}$ and $\mathcal{W}$ is the warping operation, using the flow computed by the pre-trained PWC-Net [4].

On the raw side, we need to pre-process to better position our input and have a "neutral" raw space. Hence, we normalize the image using the white and black level information in the metadata, which also provides white balance, ISO and exposure time.

3 Loss Function

As the ground-truth images are aligned using the optical flow from a pre-trained PWC-Net [4], we need to mask out pixels that were removed from the scene to avoid color inconsistencies. Therefore our losses follow Equation 2.

$$\begin{array}{rcl} \mathcal{L}_{L1}(y_d^*, y_d^w) & = & \|m \cdot (y_d^* - y_d^w)\|_1 \\ \mathcal{L}_{VGG}(y_d^*, y_d^w) & = & \|m \cdot (y_d^* - y_d^w)\|_1 \\ \mathcal{L}_{SSIM}(y_d^*, y_d^w) & = & \|m \cdot (y_d^* - y_d^w)\|_1 \end{array} \tag{2}$$

where m is the mask calculated based on the correct optical positions. We occluded pixels with flow < 0.999. We also considered occluding pixels with a mismatch between forward and backward flows as proposed by [3]. In Equation 2, L1 is the mean absolute error, VGG is the perceptual loss calculated with features extracted from a VGG-19 [5] pre-trained model, and SSIM [6] stands for the structural similarity index. Perceptual and structural loss is even more critical in reconstruction tasks when any source of misalignment can be found. The total loss is described by Equation 3.

$$\begin{array}{rc} \mathcal{L}_{Total}(y_d^*, y_d^w) = & \lambda_{L1}\mathcal{L}_{L1}(y_d^*, y_d^w) \\ & +\lambda_{VGG}\mathcal{L}_{VGG}(y_d^*, y_d^w) \\ & +\lambda_{SSIM}\mathcal{L}_{SSIM}(y_d^*, y_d^w), \end{array} \tag{3}$$

where λ_{L1},λ_{VGG} and λ_{SSIM} are respectively 1,1, and 0.1.

Our proposed strategy also learns the white balance with the image translation task. Therefore, another loss element should be added to account for this. We observed that training the illumination branch sharing the same optimizer as the reconstruction network produces better results. Equation 4 shows the final loss structure.

$$\begin{array}{c} \mathcal{L}_{TotalWB}(y_d^*, y_d^w, WB_{dg}) = \mathcal{L}_{Total}(y_d^*, y_d^w) \\ +\lambda_{illu} \left\|(WB_d - WB_{dg})\right\|_1, \end{array} \tag{4}$$

where λ_{illu} is equals to 0.1.

4 Additional Ablations

Table 1: Evaluation regarding the attention strategy. The proposed one outperform the channel attention idea. The metrics correspond to the average of all devices.

	PSNR ↑	ΔE_{ab}^*76 ↓	SSIM ↑
Proposed attention (D)	**26.16**	**6.06**	**0.807**
Channel attention	25.44	6.35	0.791

Table 2: MetaISP evaluate if the pre-training with monitor data and the frozen the statistics strategy were removed. The metrics correspond to the average of all devices.

	PSNR ↑	$\Delta E^*_{ab}76$ ↓	SSIM ↑
- Pre-training Monitor	24.96	7.07	0.782
- Frozen Normalization	26.22	6.26	0.811
All Contributions (E)	**26.63**	**5.91**	**0.817**

Table 1 and 2 demonstrate that the proposed attention outperforms channel attention idea and the normalization technique impact together with pre-training with monitor data. Table 3 demonstrate that our method performs well even when trained individually for each device. We observe that iPhone XR accuracy had a slight improvement. Meanwhile, the Pixel 6 Pro and Samsung S22 Ultra decreased their metrics. This shows that an individual translation between the raw image and its RGB version is more accurate when tackled alone. However, when translating and processing images from different devices, the jointly learned version has the advantage of having access to various color possibilities, serving as a kind of data augmentation that enriches the training. Finally, Table 4 shows the parameter counting for all proposed models.

Table 3: MetaISP also achieves state-of-the-art results when trained to independently translation from iPhone raw to Pixel, Samsung, and iPhone color appearances.

	PSNR ↑	$\Delta E^*_{ab}76$ ↓	SSIM ↑
Iphone XR	31.64	3.13	0.931
Pixel 6 Pro	24.80	7.19	0.780
Samsung S22 Ultra	23.56	7.82	0.742

Table 4: MetaISP compared against SOTA in terms of number of parameters.

	Parameters (M) ↓
MW-ISPnet [7]	29.2
LiteISP [2]	11.9
SwinIR [8]	11.8
MetaISP	**6.4**

5 Additional Results

In this section, we present additional visualizations that include the results of illuminant estimation as well as failure cases, patch visualizations of the translation process from raw images captured with an iPhone XR to Pixel 6 Pro, Samsung S22 Ultra, and iPhone XR devices. Lastly, we showcase zero-shot examples of image translation from raw images taken with a Xiaomi C40.

5.1 Illuminants Estimation

In Figure 4, the left side illustrates cases where the illuminant estimation outperforms the use of iPhone white balance information. This is because the white balance training is based on the original scalar values of the target phone, which can result in improved color accuracy for Pixel and Samsung devices. Without this strategy, the network relies solely on learning the illuminant transfer without any hint of the original values, which may lead to suboptimal results. This approach may fails in challenging lighting conditions that are not well represented in the training dataset, resulting in inaccuracies during inference.

5.2 Additional Visualizations

Figures 4,5, and6 present detailed qualitative comparisons, showcasing the color accuracy of our model across various lighting conditions, such as night scenes, indoor settings, and outdoor environments, for all target devices. In Figure 3 and 2, we provide an additional full-resolution image visualization, comparing our results against the ground truth. These visualizations offer comprehensive insights into the performance of our model in different scenarios, highlighting its effectiveness in achieving accurate color reproduction.

5.3 Zero-shot

Figures 7,8,9,10,11, and 12 showcase additional examples of our method's generalization capabilities using raw images from Xiaomi C40 without further training.

References

[1] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 1469–1472. [Online]. Available: https://doi.org/10.1145/1873951.1874249

[2] Z. Zhang, H. Wang, M. Liu, R. Wang, W. Zuo, and J. Zhang, "Learning raw-to-srgb mappings with inaccurately aligned supervision," in *Proc. ICCV*, 2021.

[3] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[4] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. CVPR*, 2018.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[6] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

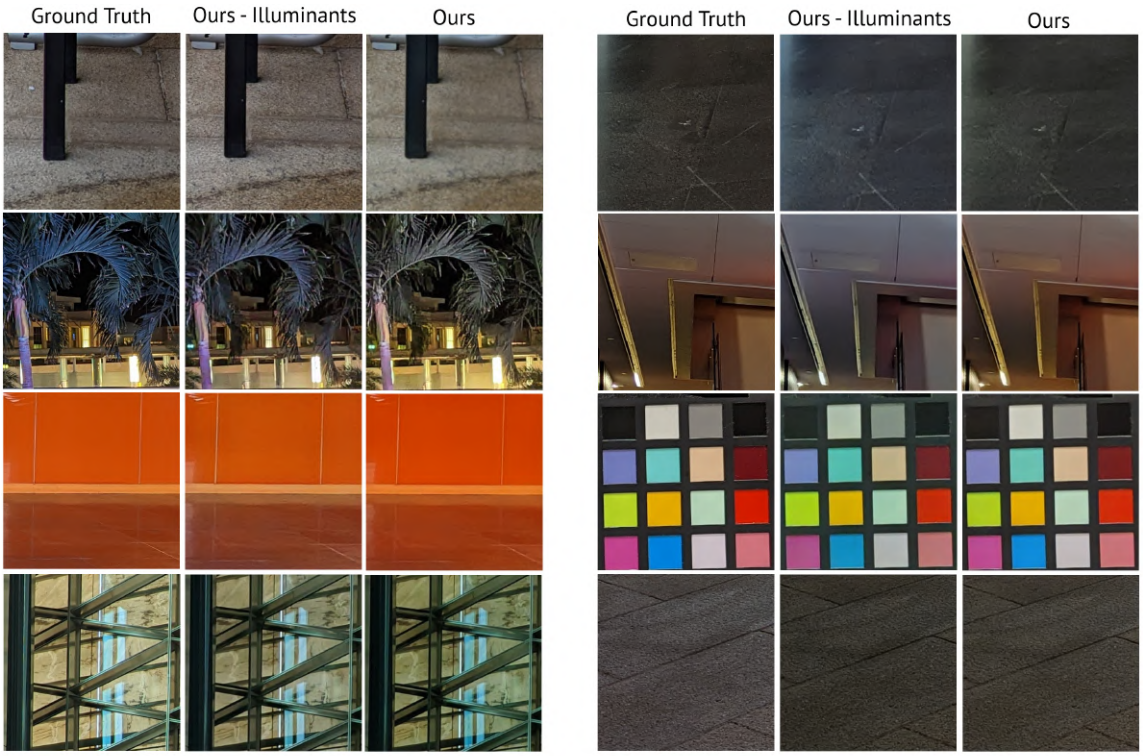

Figure 1: The illuminants analysis is depicted in the images on the left side, which illustrate examples where the estimation of illuminants aids in achieving more accurate colors. On the right side, we present some failure cases where the estimation layer is unable to provide accurate white balance information, resulting in color deviations.

[7] A. Ignatov, R. Timofte, Z. Zhang, M. Liu, H. Wang, W. Zuo, J. Zhang, R. Zhang, Z. Peng, S. Ren, L. Dai, X. Liu, C. Li, J. Chen, Y. Ito, B. Vasudeva, P. Deora, U. Pal, G. Zhenyu, and B. Hou, *AIM 2020 Challenge on Learned Image Signal Processing Pipeline*, 01 2020, pp. 152–170.

[8] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *Proc. CVPR*, 2021, pp. 1833–1844.

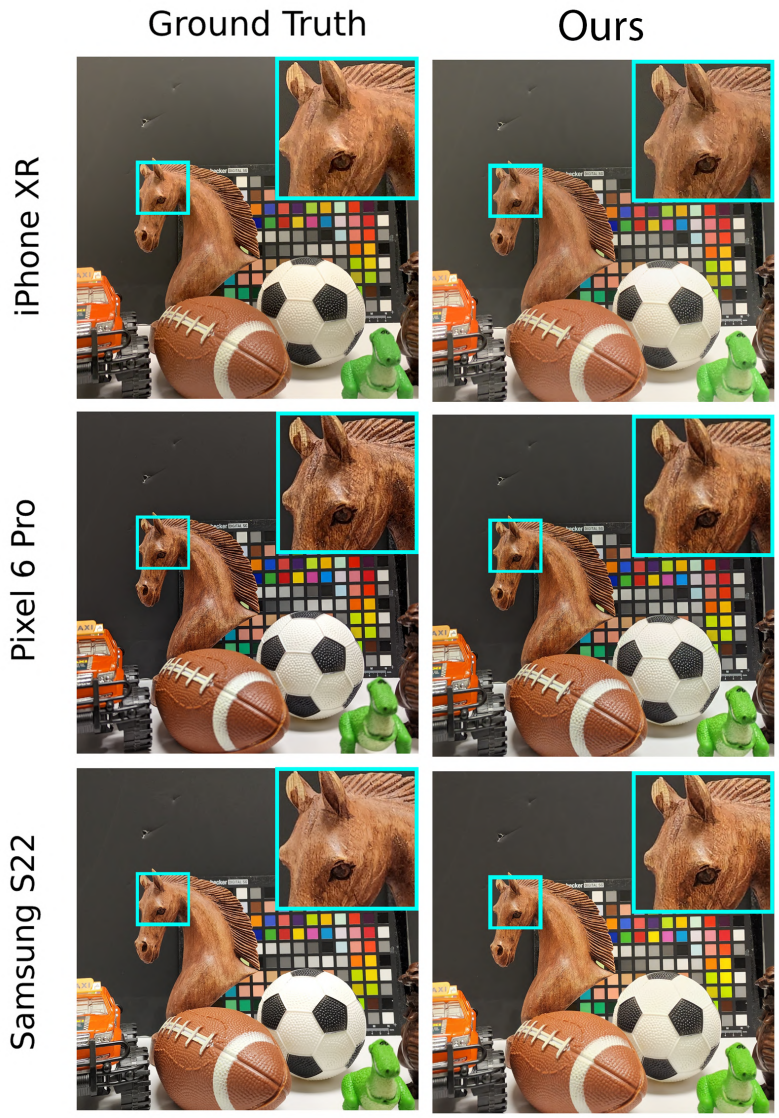

Figure 2: Full-resolution inference visualization. Here highlight the details of the horse that are more apparent in the Pixel images. MetaISP is capable of accurately reproducing the color perception using the raw input from iPhone, showcasing its ability to generate visually pleasing and accurate results, even in cases where the color rendition of different devices vary. This demonstrates the effectiveness of our approach in preserving and reproducing device-specific color characteristics, as evident in the high-quality output visualizations.

Figure 3: Full resolution with comparison - Night Scene.

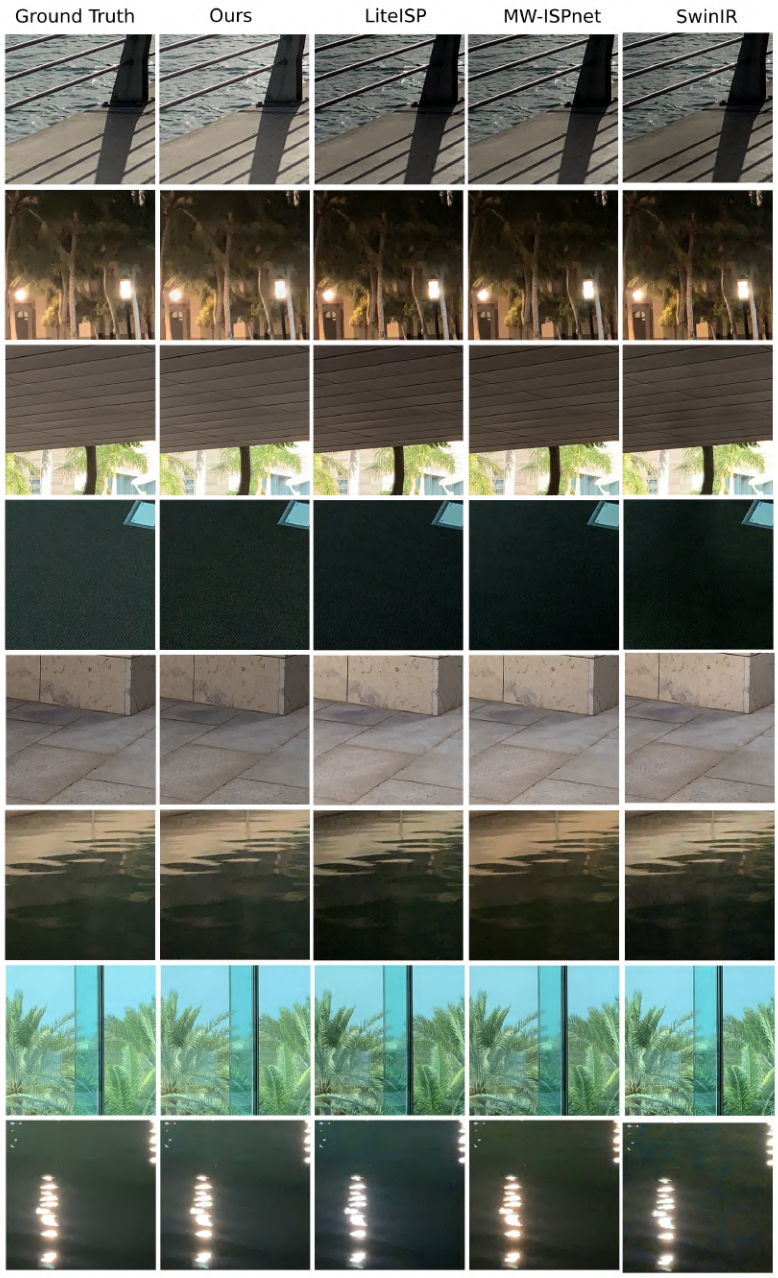

Figure 4: Patch comparison - iPhone XR.

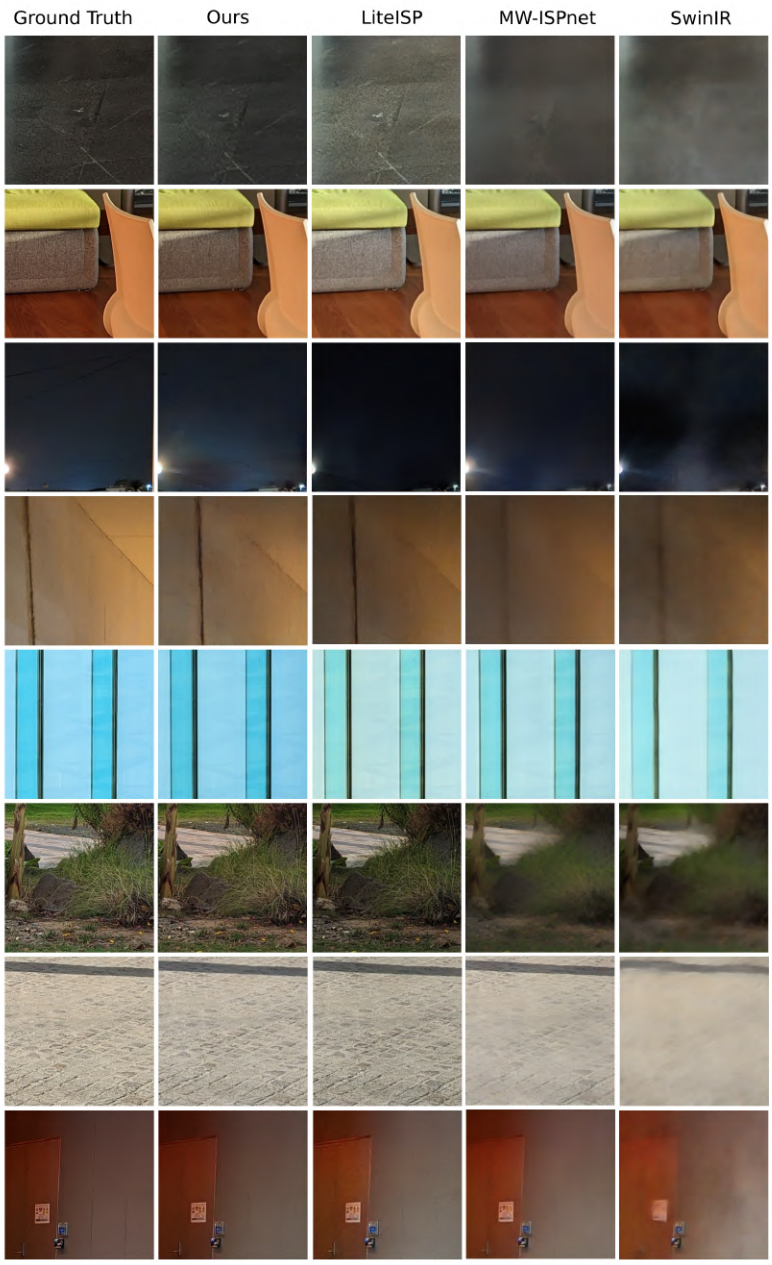

Figure 5: Patch comparison - Pixel 6 Pro.

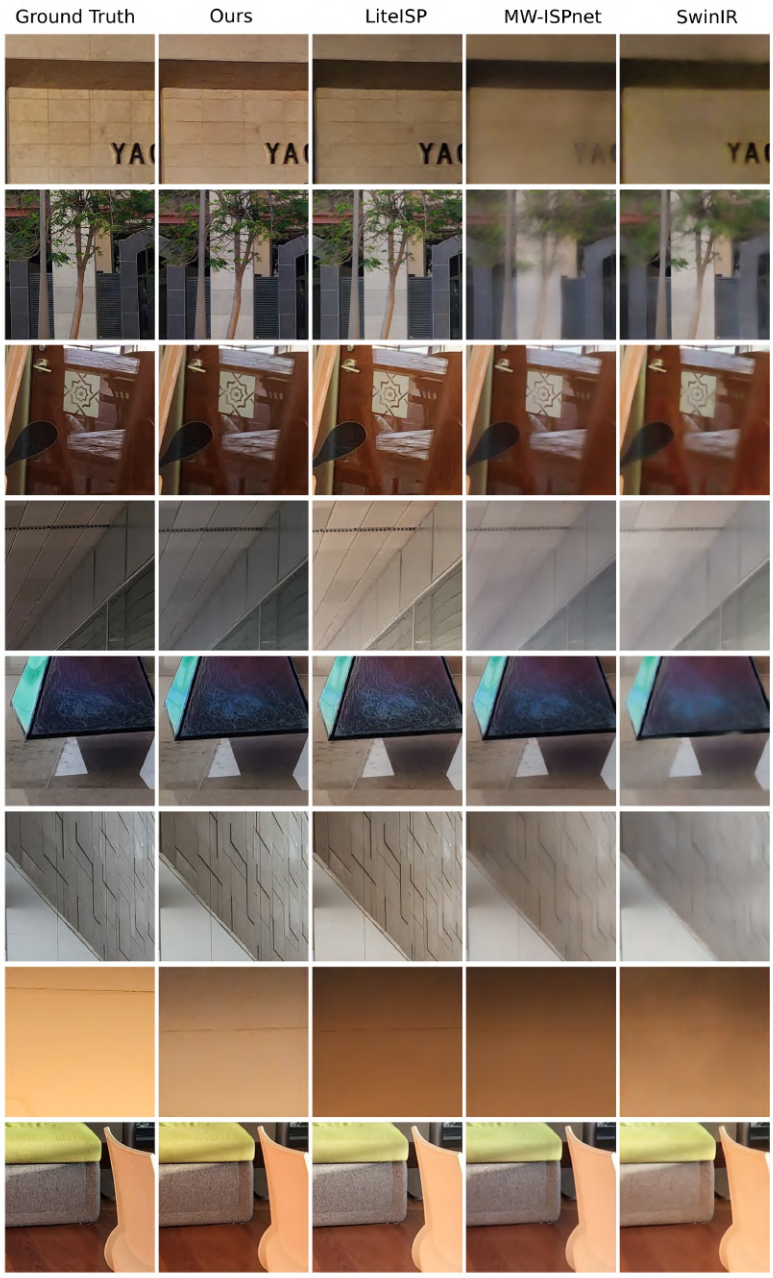

Figure 6: Patch comparison - Samsung S22 Ultra.

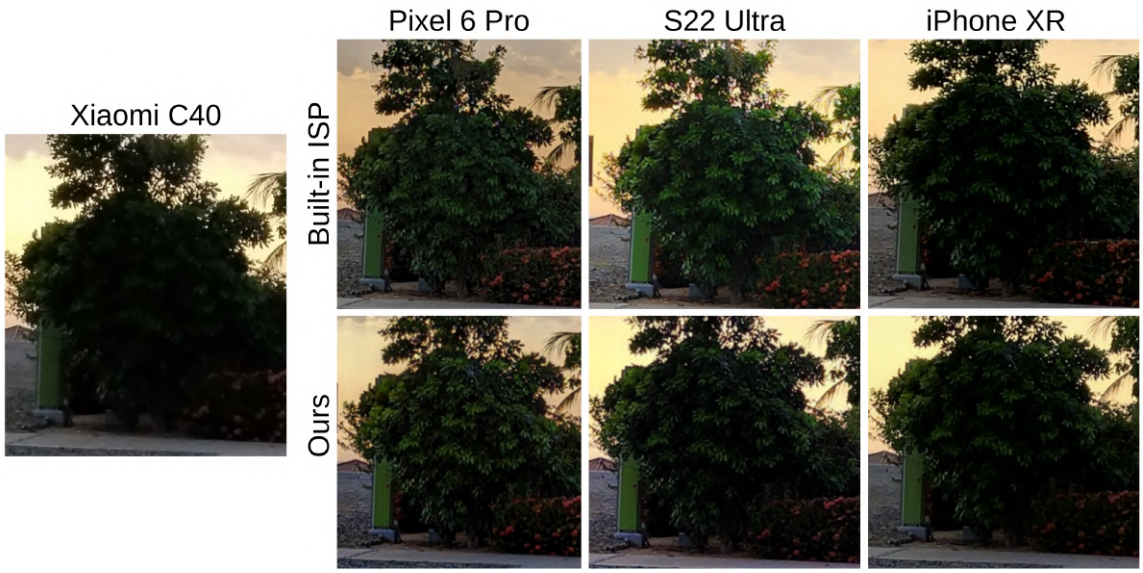

Figure 7: Xiaomi C40 - 1

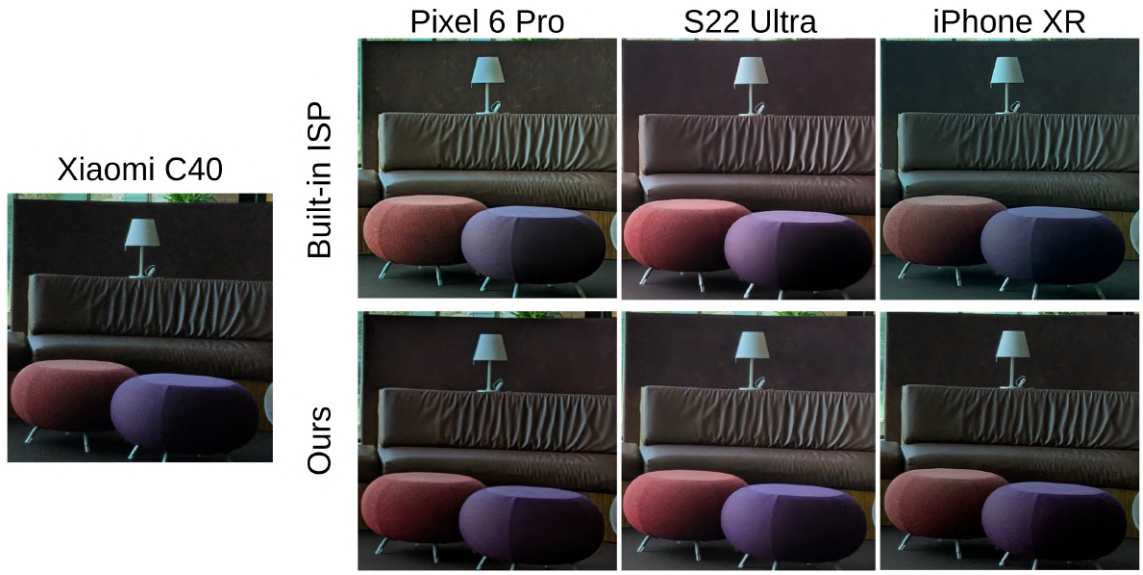

Figure 8: Xiaomi C40 - 2

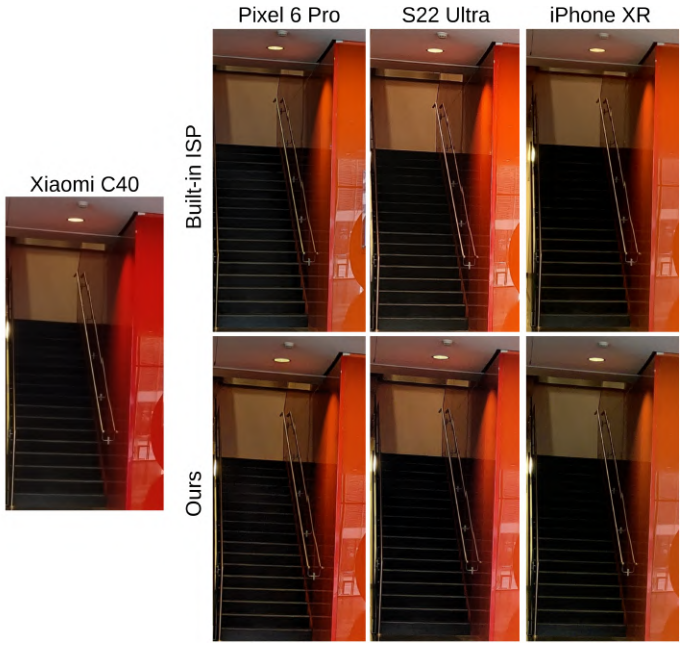

Figure 9: Xiaomi C40 - 3

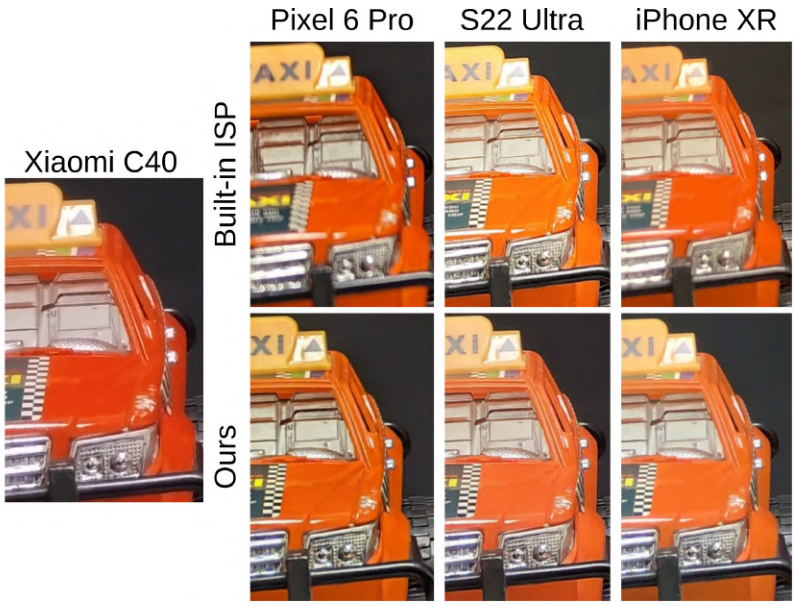

Figure 10: Xiaomi C40 - 4

Figure 11: Xiaomi C40 - 5

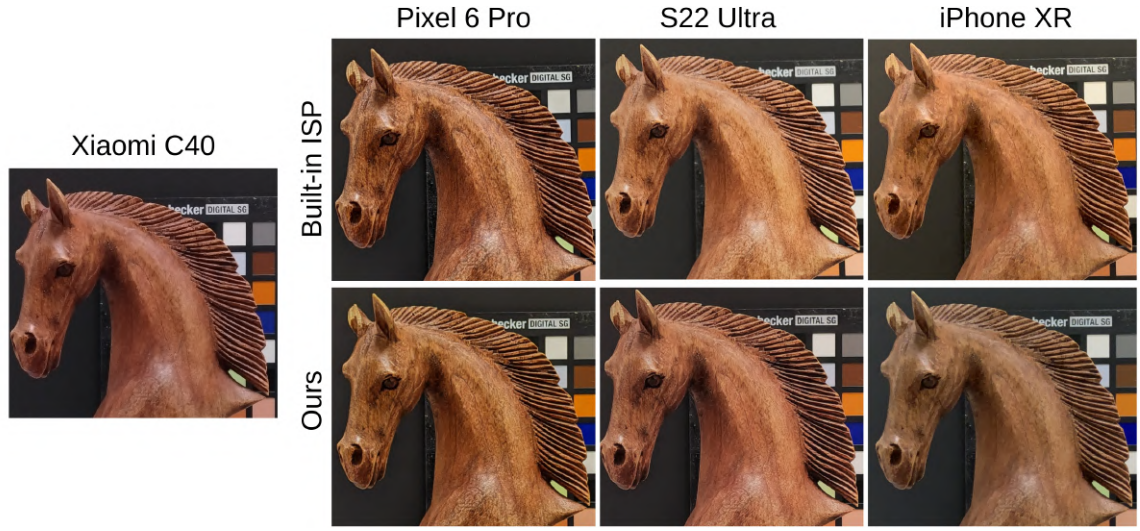

Figure 12: Xiaomi C40 - 6