Attention-aware Learning for Hyperparameters Prediction in Image Processing Pipelines (Supplemental Material)

Haina Qin[1,2], Longfei Han[3], Juan Wang[1], Congxuan Zhang[4], Yanwei Li[5], Bing Li[1,2(✉)], and Weiming Hu[1,2]

[1] NLPR, Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
{qinhaina2020@,jun_wang@,bli@nlpr.,wmhu@nlpr.}ia.ac.cn
[3] Beijing Technology and Business University longfeihan@btbu.edu.cn
[4] Nanchang Hangkong University [5] Zeku Technology, Shanghai

1 Additional Details on the ISP used for Experiments

We developed a synthetic ISP in order to validate our attention-aware parameter prediction method on several computer vision tasks. A series of processing blocks in the synthetic ISP convert the input RAW sensor data into the output RGB image, Here, the synthetic ISP consists of *Noise Reduction*, *White Balance*, *DigitalGain*, *Demosaicking*, *Color Space Transform*, *Sharpening*, *Color and Tone Correction* and *Compression* block. It replaces the hardware ISP for object detection, image segmentation, and human vision experiments in the main document. Our parameter prediction network performs these parameters in the blocks mentioned above. The details of each processing blocks in the synthetic ISP are described as follows.

- Noise Reduction: For simple overall BM3D filtering strength adjustment, use the 'filter-strength' parameter. The type of noise is determined by 'noise-type', which has a list of accepted types. Noise variance of the resulting noise is controlled by 'noise-var'. 'lambda-thr3d' is threshold parameter for the hard-thresholding in 3D transform domain. PSDs multiplied with the value of 'mu2'.
- DigitalGain and White Balance: In this block, 'gain-r', 'gain-g' and 'gain-b' are multiplied by the r, g and b pixels on the bayer filter respectively. To adjust the gain of the imaging sensor signal, the image signal is multiplied by 'digital-gain'.
- Color Space Transform: To transform native RGB values of sensors, the original R, G, B is multiplied with Color Correction Matrix (CCM) to obtain R', G', B':

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} c00\ c01\ c02 \\ c10\ c11\ c12 \\ c20\ c21\ c22 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

Table 1: Hyperparameters of the synthetic ISP and their operational ranges.

(a) Noise Reduction

Parameter	Operational Range
filter-strength	$\{0, 0.1, \ldots, 2.0\}$
noise-type	$\{\text{gw}, \text{g0}, \ldots, \text{g1w}, \ldots\}$
noise-var	$\{0.01, 0.02, \ldots, 1.0\}$
lambda-thr3d	$\{1.0, 2.0, \ldots, 100.0\}$
mu2	$\{0.1, 0.2, \ldots, 100.0\}$

(b) DigitalGain and White Balance

Parameter	Operational Range
gain-r	$\{0.01, 0.02, \ldots, 5.0\}$
gain-g	$\{0.01, 0.02, \ldots, 5.0\}$
gain-b	$\{0.01, 0.02, \ldots, 5.0\}$
digital-gain	$\{0.01, 0.02, \ldots, 2.0\}$

(c) Color Space Transform

Parameter	Operational Range
ccm 00	$\{0.01, 0.02, \ldots, 4.0\}$
ccm 01	$\{-4.0, -3.99, \ldots, 4.0\}$
ccm 02	$\{-4.0, -3.99, \ldots, 4.0\}$
ccm 10	$\{-4.0, -3.99, \ldots, 4.0\}$
ccm 11	$\{0.01, 0.02, \ldots, 4.0\}$
ccm 12	$\{-4.0, -3.99, \ldots, 4.0\}$
ccm 20	$\{-4.0, -3.99, \ldots, 4.0\}$
ccm 21	$\{-4.0, -3.99, \ldots, 4.0\}$
ccm 22	$\{0.01, 0.02, \ldots, 4.0\}$

(d) Sharpening

Parameter	Operational Range
usm-amount	$\{0.1, 0.2, \ldots, 3.0\}$
usm-sigma	$\{0.1, 0.2, \ldots, 10.0\}$

– Sharpening: Amount of sharpening can be controlled via 'usm-amount', a multiplication factor for the sharpened signal. 'use-sigma' is a scalar or sequence of scalars.

The table 1 shows that the operational range of the hyperparameters in synthetic ISP that we predicted in our experiments. For the discrete parameters, we encode them continuously. The prediction result which have minimal distance from the predicted value is chosen from the operational ranges.

2 Additional Details on Training and Inference Stages

The training process of the ISP hyperparameter prediction framework has two main phases. The first stage is to train a differentiable ISP proxy network. The goal is to make the output images ($\mathbf{O}_{\text{ISP}}$) as the same as $\mathbf{O}_{\text{proxy}}$ possible when the ISP proxy has the same input (input image $\mathbf{I}$, parameters $\mathcal{P}$) as the target ISP. The goal of the second stage is to train the parameter prediction network. It predicts all the parameters for the input image $\mathbf{I}$ to reduce the Loss of the downstream task $\hat{\mathcal{P}}$. In this stage, the weights of the ISP proxy and the downstream task network are fixed, and only the weights in parameter prediction network are fine-tuned.

Fig. 1: Samples from the ISP proxy dataset.

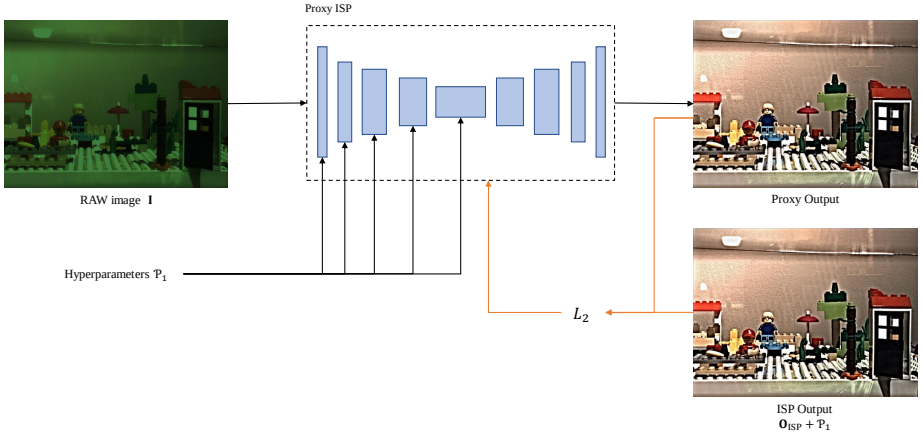

Fig. 2: ISP proxy training process. The proxy network is trained by the $\mathcal{L}_2$ of the proxy output with the ISP output

2.1 ISP Proxy Tranining Stage

An additional dataset is prepared for the training of the ISP proxy network. We collect the dataset as follows.

- RAW image $\mathbf{I}$ and parameters $\mathcal{P}$ as the input of ISP.
- Output image $\mathbf{O}_{\text{ISP}}$. $\mathbf{O}_{\text{ISP}}$ is generated by ISP with input $\mathbf{I}$ and $\mathcal{P}$.

In the dataset, for a RAW image $\mathbf{I}$, we generate a set of output images $\{\mathbf{O}_{\text{ISP}}^{(0)}, \mathbf{O}_{\text{ISP}}^{(1)}, \ldots \mathbf{O}_{\text{ISP}}^{(m)}\}$ with the corresponding parameter sets $\{\mathcal{P}_0, \mathcal{P}_1, \ldots \mathcal{P}_m\}$, the various parameter sets are sampled from the parameter space by latin-hypercube (LHC) random sampling method. For every combination $\{\mathbf{I}, \mathcal{P}_i\}$, the reference image is $\mathbf{O}_{\text{ISP}}^{(i)}$, and we use each pair as one training sample. The dataset is shown in Figure 1. The images in the dataset are sliced into 512x512 image patches, where the input RAW image is split into R, G, and B channels according to bayer array as the input to the ISP proxy.

As described in the paper, the ISP proxy used is a variant of U-net architecture [3], which aims to construct a mapping function from $\{\mathbf{I}, \mathcal{P}_i\}$ to $\mathbf{O}_{\text{ISP}}^{(i)}$. In our approach, the input RAW image $\mathbf{I}$ is ingested into the U-net to extract feature maps. Meanwhile, we expand the parameters $\mathcal{P}$ as a new channel, and concatenate it with the channels of feature maps to produce an output image $\mathbf{O}_{\text{PROXY}}$. We use $\mathbf{O}_{\text{PROXY}}$ with the corresponding $\mathbf{O}_{\text{ISP}}$ to train the network by minimizing the $\mathcal{L}_2$ loss between them. This process is shown in Figure 2. The network is optimized via RMSprop optimizer, with the learning rate set to 0.0001 and the learning rate decaying by 10% after every 10 epochs. We train the network on the COCO dataset and human viewing dataset respectively, and it takes about 200 epochs to converge. The output of ISP Proxy is presented in Figure 5.

2.2 Parameter Prediction Network Tranining Stage

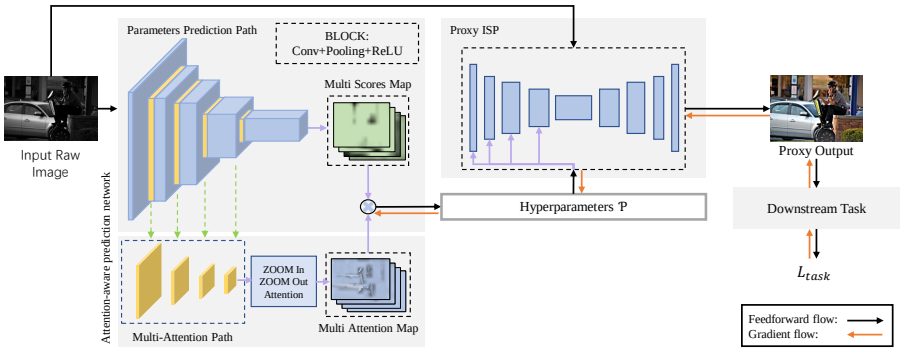

Fig. 3: The training process of the parameter prediction network.

The training dataset used for the parameter prediction network contains two parts.

- RAW image **I** is also the input of ISP.
- Ground truth information corresponding to **I**. For the object detection and image segmentation tasks, ground truth is the object localization and object categories on the image. For the human viewing task, ground truth is the reference image corresponding to the RAW image.

The training process of the parameter prediction network is described in the main manuscript, and its flow is shown in Figure 4. During the training stage, the weights of the ISP proxy and the downstream task network are fixed, and the weights of the parameter prediction network are trained by minimizing the loss of the downstream task.

2.3 Inference Stage

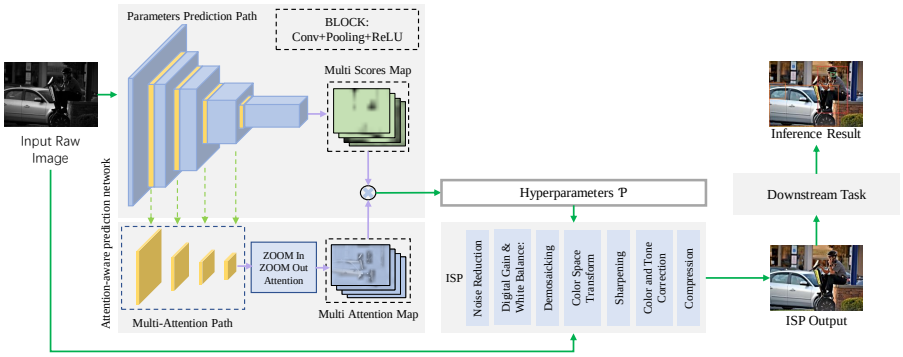

Fig. 4: Prediction of ISP parameters using the parameter prediction network. The output of the ISP is used as input for downstream tasks.

In the inference stage, the RAW image is used as the input of the parameter prediction network and the predicted parameters are obtained. Then, the predicted parameters and the RAW image are used as the input of the ISP to get the RGB image. The flow is shown in Fig. In the evaluation process of downstream tasks, we transform the RAW test set into RGB test set by the above process, and the RGB test set is used as the input of downstream tasks to get the evaluation results.

Table 2: Object Detection Task using [2] and COCO [1]

	EfficientDet Model	$mAP_{0.5}$	$mAP_{0.75}$	$mAP_{0.5:0.95}$
Default $\mathcal{P}$	D0	0.309	0.200	0.192
Predicted $\hat{\mathcal{P}}$ (Our)		0.490	0.334	0.317
Default $\mathcal{P}$	D1	0.327	0.216	0.207
Predicted $\hat{\mathcal{P}}$ (Our)		0.552	0.392	0.369
Default $\mathcal{P}$	D2	0.353	0.238	0.229
Predicted $\hat{\mathcal{P}}$ (Our)		0.588	0.424	0.401
Default $\mathcal{P}$	D3	0.362	0.251	0.239
Predicted $\hat{\mathcal{P}}$ (Our)		0.616	0.456	0.430
Default $\mathcal{P}$	D4	0.451	0.314	0.299
Predicted $\hat{\mathcal{P}}$ (Our)		0.656	0.495	0.463
Default $\mathcal{P}$	D5	0.412	0.289	0.275
Predicted $\hat{\mathcal{P}}$ (Our)		0.666	0.508	0.473
Default $\mathcal{P}$	D6	0.427	0.302	0.287
Predicted $\hat{\mathcal{P}}$ (Our)		0.673	0.516	0.481
Default $\mathcal{P}$	D7	0.456	0.324	0.308
Predicted $\hat{\mathcal{P}}$ (Our)		0.693	0.538	0.501
Default $\mathcal{P}$	D7X	0.494	0.353	0.335
Expert-tuned		0.689	0.534	0.498
Predicted $\hat{\mathcal{P}}$ (Our)		0.707	0.556	0.515

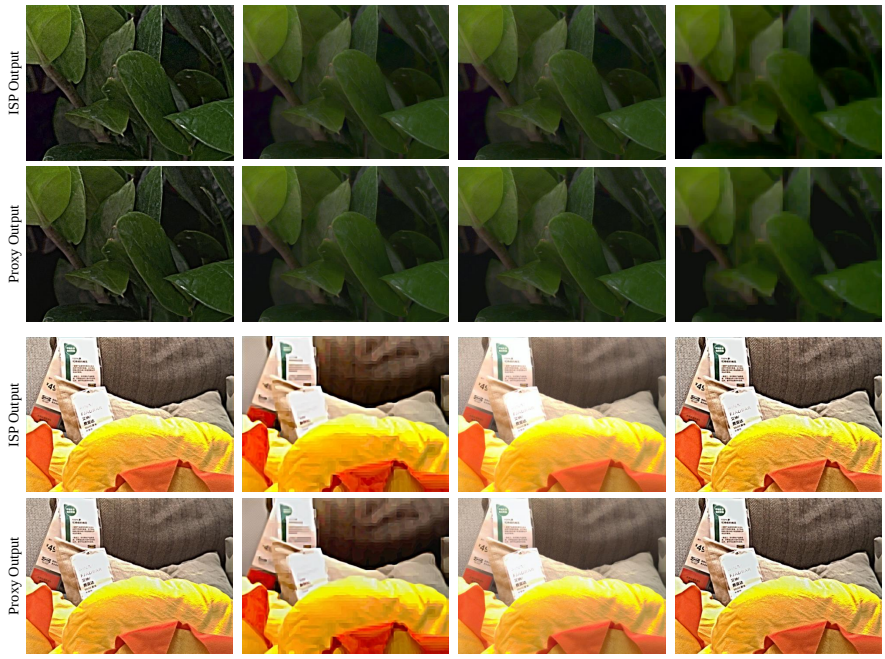

Fig. 5: Proxy outputs with the corresponding ISP outputs on the human viewing dataset.

2.4 Additional objective experiments

In this section, we demonstrate additional experiments for the object detection task. To validate the effectiveness of the proposed method towards different models, we use EfficientDet [2] on the COCO dataset [1] as the downstream task of our method to predict hyperparameters on the synthetic ISP. We trained and tested our hyperparameter prediction results. The results in Table 2 show that our method has a great improvement over the default parameters and the expert-tuned parameters. We also conducted experiments on different backbones of the downstream model, and the experimental results show that our method is effective for different models.

References

1. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755 (2014)
2. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781–10790 (2020)
3. Tseng, E., Yu, F., Yang, Y., Mannan, F., Arnaud, K.S., Nowrouzezahrai, D., Lalonde, J.F., Heide, F.: Hyperparameter optimization in black-box image processing using differentiable proxies. ACM Transactions on Graphics **38**(4), 27–1 (2019)