RL-SeqISP: Reinforcement Learning-Based Sequential Optimization for Image Signal Processing

Xinyu Sun[1,2*], **Zhikun Zhao**[1,2*], **Lili Wei**[1*], **Congyan Long**[1†],
Mingxuan Cai[3], **Longfei Han**[4], **Juan Wang**[2], **Bing Li**[2,5], **Yuxuan Guo**[6]

[1]Key Laboratory of Big Data & Artificial Intelligence in Transportation (Ministry of Education), School of Computer and Information Technology, Beijing Jiaotong University
[2]Institute of Automation, Chinese Academy of Sciences
[3]Shanghai JiaoTong University
[4]Beijing Technology and Business University
[5]PeopleAI Inc. Beijing, China
[6] Shenzhen Heytap Technology Corp., Ltd
{xinyusun, zhikunzhao, cylang}@bjtu.edu.cn, caimingxuan@sjtu.edu.cn, bli@nlpr.ia.ac.cn, longfeihan@btbu.edu.cn

Abstract

Hardware image signal processing (ISP), aiming at converting RAW inputs to RGB images, consists of a series of processing blocks, each with multiple parameters. Traditionally, ISP parameters are manually tuned in isolation by imaging experts according to application-specific quality and performance metrics, which is time-consuming and biased towards human perception due to complex interaction with the output image. Since the relationship between any single parameter's variation and the output performance metric is a complex, non-linear function, optimizing such a large number of ISP parameters is challenging. To address this challenge, we propose a novel Sequential ISP parameter optimization model, called the RL-SeqISP model, which utilizes deep reinforcement learning to optimize all ISP parameters for different imaging applications. Concretely, inspired by the sequential tuning process of human experts, the proposed model can progressively enhance image quality by seamlessly integrating information from both the image feature space and the parameter space. Furthermore, a dynamic parameter optimization module is introduced to avoid ISP parameters getting stuck into local optima, which is able to more effectively guarantee the optimal parameters resulting from the sequential learning strategy. These merits of the RL-SeqISP model as well as its high efficiency are substantiated by comprehensive experiments on a wide range of downstream tasks, including two visual analysis tasks (instance segmentation and object detection), and image quality assessment (IQA), as compared with representative methods both quantitatively and qualitatively. In particular, even using only 10% of the training data, our model outperforms other SOTA methods by an average of 7% mAP on two visual analysis tasks.

Introduction

Hardware Image Signal Processors (ISPs) are low-level image processing hardwares that convert RAW data to RGB images. Due to their high reliability and high efficiency, they are widely used in many fields, *e.g.*, camera phones (Ignatov et al. 2023; Ratnasingam 2019) and video surveillance (Lee et al. 2015; Xu et al. 2018; Baina and Dublet 1995). Generally, a typical hardware ISP pipeline consists of a set of serialized processing blocks (*e.g.*, denoising block and sharpening block), each of which contains multiple parameters, resulting in a large number of parameters, which affects the quality of the generated RGB image (Bardenet et al. 2013; Yahiaoui et al. 2019; Yogatama and Mann 2014). Traditionally, these parameters are tuned by imaging experts based on human visual perception, which is time-consuming and labor-intensive (Board 2017). Moreover, due to biases in human perception, it is challenging for experts to tune the parameters that favor actual applications. To solve this

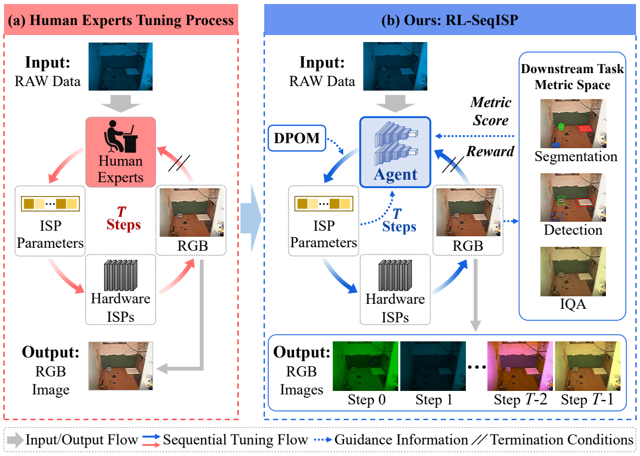


Figure 1: The motivation of our method. (a) Human experts tuning process. The imaging experts iteratively tune the ISP parameters through empirical perception. (b) The proposed RL-SeqISP utilizes deep reinforcement learning to progressively optimize ISP parameters under the guidance of feedback from various specific downstream applications.

[*]These authors contributed equally.
[†]Corresponding author.

dilemma, automatic ISP tuning becomes a new trend.

To develop automatic ISPs, some methods (Chen et al. 2018; Ignatov, Van Gool, and Timofte 2020; Yu et al. 2021a) design software ISPs based on Convolutional Neural Network (CNN) (Chen et al. 2018; Yu et al. 2021a) to replace the hardware ISPs to directly predict RGB images. In these methods, the trainable CNN weights can be regarded as implicit ISP parameters. Though effective in some ways, software ISPs are computationally intensive and time-consuming (Tseng et al. 2019). Besides, they keep fixed implicit ISP parameters for all RAW inputs during the evaluation stage, even though each RAW input should have different ISP parameters to adapt to various image contents. In contrast, other CNN-based methods (Qin et al. 2022; Tseng et al. 2019) explicitly predict the specific optimal ISP parameters for each input in a high-dimensional space. However, the performance of these methods is not satisfactory, since CNNs are data-hungry for training effectively and the available RAW-RGB datasets are limited.

In fact, human experts tune hardware ISP parameters in a step-by-step manner, *i.e.*, they first roughly tune the parameters, and then gradually fine-tune the parameters according to visual observation, to obtain visually pleasing or downstream task-friendly RGB images, as shown in Fig. 1 (a). Inspired by this process, we formulate the hardware ISP parameter tuning problem as a sequential progressive optimization problem, since the sequential optimization enables better handling of non-convex problems through fine-tuning, and may be able to train the network and update ISP parameters from smaller datasets in a target-driven manner.

Motivated by the remarkable ability of reinforcement learning (RL) to solve sequential decision-making problems, in this paper, we propose an **RL**-based **Seq**uential **ISP** parameter tuning model (*i.e.*, RL-SeqISP model), as illustrated in Fig. 1 (b). To our knowledge, we are the first to put forward an RL-based method for hardware ISP parameter tuning. Concretely, we formulate a comprehensive state representation including current parameters and the corresponding RGB image. Then, RL-SeqISP trains an agent, which consists of an actor and a critic for estimating the variation of ISP parameters and measuring the value of the input state respectively, in an end-to-end manner to optimize hardware ISP parameters for multiple steps. Furthermore, to prevent parameters from converging to local optimum during the ISP parameter tuning process (Shin, Lee, and Kweon 2022), we further propose a **d**ynamic **p**arameter **o**ptimization **m**odule (abbreviated as DPOM), which dynamically updates ISP parameters by weighing and aggregating current parameters, agent decisions, and downstream task feedbacks. To evaluate the proposed RL-SeqISP, we conduct extensive experiments on two visual analysis tasks (*i.e.*, image instance segmentation, object detection), and a human visual perception task (*i.e.*, image quality evaluation, IQA). The experimental results demonstrate that our method achieves promising performance using very limited training images.

The main contributions are summarized as follows:

- We formulate the hardware ISP parameter tuning problem as a sequential and progressive optimization problem and propose an RL-based Sequential ISP parameter tuning model (*i.e.*, RL-SeqISP model). To our knowledge, we pioneer RL for hardware ISP parameter tuning.
- To prevent the ISP parameters from converging to the local optimum, we propose a **d**ynamic **p**arameter **o**ptimization **m**odule (*i.e.*, DPOM), which guides parameter optimization through current parameters and the feedback performance from downstream tasks.
- The experimental results on three downstream tasks demonstrate the superiority of the proposed RL-SeqISP model, even with a small number of training images.

Related Works

ISP Tuning

Implicit ISP Tuning. With the development of deep learning, some studies (Schwartz, Giryes, and Bronstein 2018; Chen et al. 2018; Gharbi et al. 2016; Ignatov, Van Gool, and Timofte 2020; Dong et al. 2022; Yu et al. 2021b) have introduced end-to-end deep neural networks (DNNs) to replace the traditional hardware ISPs to directly generate RGB images. Specifically, *Chen et al.* (Chen et al. 2018) used DNN to execute the denoising process of low-light images, *Ignatov et al.* (Ignatov, Van Gool, and Timofte 2020) designed a PyNET to replace ISPs of mobile cameras. However, the above methods keep fixed implicit parameters (*i.e.*, parameters of the DNNs) for all RAW inputs during the validation phase, without considering that each image should have specific parameters based on its feature.

Explicit ISP Tuning. The purpose of explicit ISP tuning is to directly learn the optimal parameters of ISPs. Some methods (Tseng et al. 2019; Nishimura et al. 2018; Qin et al. 2022) have formulated the parameter prediction for ISP proxies as a black-box optimization problem driven by end-to-end loss. In particular, *Qin et al.* (Qin et al. 2022), which performs best among the above methods, directly infered all ISP parameters via an attention-based CNN. However, their generality and effectiveness will be greatly reduced due to the gap between CNN proxies and hardware ISPs. In contrast, some Covariance Matrix Adaptation Evolutionary Strategies-based (CMA-ES-based) methods (Mosleh et al. 2020; Robidoux et al. 2021) aim to optimize ISP parameters for actual hardware ISPs. However, the ES-based methods inevitably introduce errors into the parameter optimization process. Our approach aims to sequentially optimize the parameters of the actual hardware ISP based on the evaluation metrics of several applications.

Reinforcement Learning

Reinforcement learning (RL) is a classic machine learning paradigm (Watkins and Dayan 1992; Lillicrap et al. 2015) known for its ability to solve sequential problems. Recently, many researchers have investigated RL in low-level image quality enhancement (Bajaj et al. 2021; Wang et al. 2020; Furuta, Inoue, and Yamasaki 2019; Zhang et al. 2021b,a). More specifically, in the field of image signal processing, *Shin et al.* (Shin, Lee, and Kweon 2022) first replaced the actual ISP with an ISP toolbox, and trained an RL agent to determine the processing order of each tool. However, the

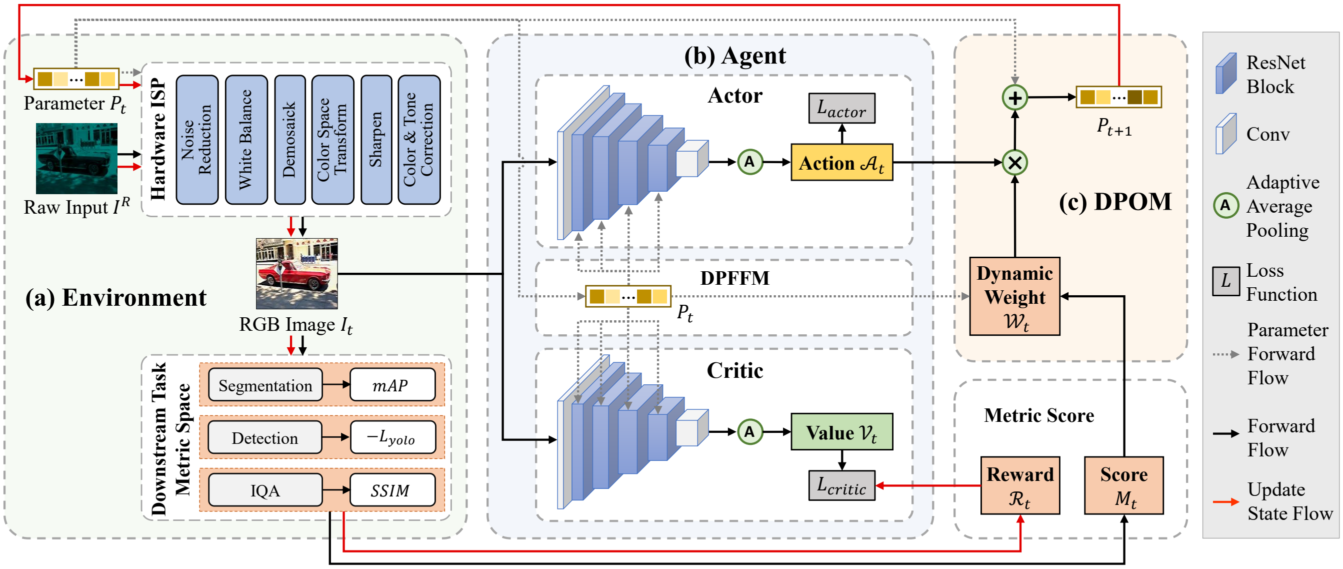


Figure 2: The overall pipeline of the proposed RL-SeqISP model, which tackles the hardware ISP parameter tuning problem as a sequential and progressive parameter optimization problem. (a) At each time step t, taking I^R and P_t as input, the environment generates an RGB image I_t and a metric score M_t. (b) The Agent observes the state $\mathcal{S}_t$ (including P_t and I_t) to decide an action $\mathcal{A}_t$ for parameter optimization. (c) The dynamic parameter optimization module (DPOM) updates P_t to P_{t+1} according to the trade-off among $\mathcal{A}_t$, P_t and M_t. The above processes iterate T steps to continuously optimize hardware ISP parameters.

parameters of each tool are still fixed for all RAW inputs. In contrast, we further propose a new paradigm that utilizes an agent to sequentially predict specific parameters for each ISP module rather than select ISP modules.

METHODS

Overview

Problem Formulation. Given a RAW image I^R from an image sensor, the goal of ISP parameter prediction is to predict a set of image-specific ISP parameters $P \in R^k$, where k denotes the number of parameters. Then, an output RGB image I can be generated by the hardware ISP $F_{ISP}(;)$ with the parameters P, formulated as $I = F_{ISP}(I^R; P)$. In this paper, we formulate the ISP parameter prediction problem as a sequential parameter optimization problem. Specifically, given an environment $\mathcal{E}$, including a hardware ISP and a performance metric function $M(\cdot)$ for a pre-trained downstream task, the RL agent interacts with $\mathcal{E}$ and executes an action to optimize P in multiple steps, expressed as $\{\mathcal{S}_t, \mathcal{A}_t, \mathcal{R}_t, \mathcal{S}_{t+1}\}_{t=0}^{T-1}$, where T, t, $\mathcal{S}_t$, $\mathcal{A}_t$, $\mathcal{R}_t$ and $\mathcal{S}_{t+1}$ denote the max optimization steps T, current timestamp t, current state $\mathcal{S}_t = \{P_t, I_t = F_{ISP}(I^R; P_t)\}$, an action $\mathcal{A}_t$, a reward value $\mathcal{R}_t = M(I_{t+1}) - M(I_t)$, and the next state $\mathcal{S}_{t+1}$ after executing the action $\mathcal{A}_t$. Note that P_0 is randomly initialized. After the T steps of parameter optimization, the final optimal ISP parameters P^* can be obtained.

Overview Pipeline. Fig. 2 depicts the overall framework of the proposed **RL**-based **Seq**uential **ISP** parameter tuning model (*i.e.*, RL-SeqISP model). When optimizing the ISP parameters, a RAW image I^R and the ISP parameters P_t are fed to the environment $\mathcal{E}$ to obtain a state $\mathcal{S}_t$ and a metric score $M(I_t)$, as shown in Fig. 2 (a). Then the agent receives $\mathcal{S}_t$ to decide an action $\mathcal{A}_t$ for optimizing the ISP parameters P_t, as shown in Fig. 2 (b). Specifically, the agent is actually an Actor-Critic Network, which consists of two core networks, *i.e.*, an actor network for predicting an action $\mathcal{A}_t$ to optimize P_t, and a critic network for predicting the value V_t to measure the value of the input state $\mathcal{S}_t$, respectively. For better prediction, both the actor network and the critic network are guided by P_t at multiple scales with a dual-branch parameter-guided feature fusion module (DPFFM). Furthermore, to prevent ISP parameters from converging to the local optimum, we further propose a dynamic parameter optimization module (DPOM), as shown in Fig. 2 (c). After the above forward process, once the agent executes $\mathcal{A}_t$, P_t and $\mathcal{S}_t$ are updated to P_{t+1} and $\mathcal{S}_{t+1}$, thus can obtain a reward value $\mathcal{R}_t$ by measuring the quality of $\mathcal{A}_t$. Following this process, P_0 are iteratively optimized to become the optimal parameter P^*. During the training phase, the agent is optimized by two different loss functions, respectively.

Agent for ISP Parameter Optimization

The agent, aiming to make decisions based on the current state space $\mathcal{S}_t$ to optimize ISP parameters, is essentially an Actor-Critic network. To guide the agent to make better decisions, we further propose a dual-branch parameter-guided feature fusion module (DPFFM).

Actor-Critic Network. The Actor-Critic network contains two subnetworks: *i.e.*, an actor network and a critic network. Both networks adopt the same structure, containing several modules in series: *i.e.*, a convolutional (conv) layer, several Residual blocks (He et al. 2016), a conv layer, and an adaptive average pooling. Specifically, taking the current state $\mathcal{S}_t$ as input, the actor network aims to predict the variation of ISP parameters as action $\mathcal{A}_t \in R^k$, formulated as:

$$\mathcal{A}_t = \pi(\mathcal{S}_t; \theta_\pi), \tag{1}$$

where $\pi(; \theta_\pi)$ denotes the actor network with network parameters θ_π. Based on $\mathcal{A}_t$, the agent can optimize P_t via:

$$P_{t+1} = clip(Norm(P_t) + Norm(\mathcal{A}_t), -1, 1), \tag{2}$$

where $clip(\cdot)$ is a clipping function (Schulman et al. 2017) that limits each value of ISP parameters to $[-1, 1]$, $Norm(\cdot)$ denotes a normalization operation. The critic network aims to predict a value $\mathcal{V}_t$ to measure the value of the input state $\mathcal{S}_t$, formulated as:

$$\mathcal{V}_t = V(\mathcal{S}_t; \theta_V), \tag{3}$$

where $V(; \theta_V)$ denotes a critic network with network parameters θ_V.

Dual-branch Parameter-guided Feature Fusion Module. To make better decisions, the agent should make full use of the information in $\mathcal{S}_t$, *i.e.*, visual information of the RGB image I_t, and the parameter preference implied by P_t. For each subnetwork, DPFFM fuses the image feature F_{I_t} and P_t to generate a better feature $\hat{F}_{I_t}$, formulated as:

$$\hat{F}_{I_t} = F_{I_t} \copyright \xi(P_t), \tag{4}$$

where © and $\xi(\cdot)$ denote the channel-wise concatenation and the reshaping operation (Tseng et al. 2019) which replicates P_t to the same spatial dimension as F_{I_t}. Furthermore, considering the different representation capabilities of low-level and high-level features, we apply multi-layer feature fusion before each residual block, formulated as:

$$\hat{F}^l_{I_t} = F^l_{I_t} \copyright \xi^l(P_t), l \in \{1, ..., L\}, \tag{5}$$

where l denotes the layer of Residual blocks.

Dynamic Parameter Optimization Module

Equ. 2 introduces a naive way to update the ISP parameters. However, during the ISP parameter optimization process, some ISP parameters tend to approach the parameter boundary (*i.e.*, -1 or 1), due to insufficient exploration of state space by the agent. Similar effects also exist in RL domains (Silver et al. 2016; Schaul et al. 2015; Osband et al. 2019; Sutton and Barto 2018). More seriously, this effect will accumulate during the sequential optimization, causing the optimized ISP parameters to stick to the local optimum. To alleviate this problem, we propose a dynamic parameter optimization module (DPOM) that guides parameter optimization by dynamically weighing the current parameters and the feedback performance from a downstream task.

Feedback performance metric of downstream tasks. To measure the quality of the RGB images I_t, a metric function $M(\cdot)$ needs to be defined. However, since the aesthetic judgments of an image are influenced by many factors and vary from person to person, it is inconvenient to directly formulate quality assessment metrics for RGB images. Instead, we evaluate the quality of I_t based on its performance in a downstream task, expressed as:

$$M(I_t) = M_{dt}\big(F_{dt}(I_t; \theta)\big), \tag{6}$$

where $F_{dt}(I_t; \theta)$ denotes a pre-trained network with frozen network parameters θ, $M_{dt}(\cdot)$ denotes an evaluation metric of F_{dt}. Note that a higher feedback metric score implies a better-quality RGB image. In this paper, we adopt the following metrics on two visual analysis tasks, *i.e.*, negative YOLO loss ($-L_{yolo}$) for object detection (Redmon and Farhadi 2018) and mAP for instance segmentation (Qin et al. 2022). In addition, intuitively from the perspective of human visual perception, we adopt $SSIM$ for the image quality assessment task (IQA) (Qin et al. 2022).

Dynamic Parameter Optimization. For dynamic parameter optimization, we modify Equ. 2 to:

$$P_{t+1} = clip(Norm(P_t) + \mathcal{W}_t \times Norm(\mathcal{A}_t), -1, 1), \tag{7}$$

where $\mathcal{W}_t \in R^k$ denotes a set of adaptive weights, determined by the trade-off between the state of P_t and the feedback performance metric score $M(I_t)$, formulated as:

$$\mathcal{W}_t = \alpha(e^{1-|P_t|} - 1) \cdot log\big[\beta\big(M_{tgt}(I_t) - M(I_t)\big) + 1\big], \tag{8}$$

where α and β denote two hyperparameters to balance the weight, $1 - |P_t|$ represents the distance to the parameter boundary (*i.e.*, -1 or 1), $M_{tgt}(I_t)$ denotes the desired target score (*i.e.*, 0.99). According to Equ. 8, when a parameter in P_t approaches the boundary or the feedback metric score approaches our desired target score, the agent tends to suppress update amplitude and keep the original parameter P_t.

Training Actor-Critic Network of Agent

Reward. To train the agent, a reward function needs to be defined. After the agent executes an action $\mathcal{A}_t$ at time step t, our goal is to generate better ISP parameters P_{t+1} and RGB image I_{t+1} for higher downstream task performance $M(I_{t+1})$. Here we define the reward function $\mathcal{R}(\cdot)$ as the gain of $M(\cdot)$ before and after executing $\mathcal{A}_t$, formulated as:

$$\mathcal{R}_t = M(I_{t+1}) - M(I_t). \tag{9}$$

After the agent iteratively optimizes the ISP parameters T steps, a set of rewards can be generated, *i.e.*, $\{\mathcal{R}_0, ..., \mathcal{R}_{T-1}\}$. Then the accumulated reward score R_t for each step can be defined as:

$$R_t = \mathcal{R}_t + \gamma\mathcal{R}_{t+1} + \cdots + \gamma^{T-t+1}\mathcal{R}_{T-1} + \gamma^{T-t}V(\mathcal{S}_T; \theta_V), \tag{10}$$

where γ denotes the discount factor.

Optimization of Actor-Critic Network. We follow the PPO algorithm (Schulman et al. 2017) to train the RL-SeqISP model, as described in Algorithm 1.

Specifically, to optimize the critic network, we aim to minimize the expectation of the difference between its predicted value $\mathcal{V}_t$ and the accumulated reward score R_t. We adopt a loss L_{critic} for critic network, formulated as:

$$\mathrm{L}_{critic} = \hat{E}_t[(R_t - V(\mathcal{S}_t; \theta_V))^2]. \tag{11}$$

To optimize the actor network, we aim to maximize cumulative return expectations. To this end, we adopt a loss function L_{actor} for actor network, formulated as:

$$\mathrm{L}_{actor} = \hat{E}_t\left[\min\left(r_t(\theta_\pi)\hat{A}_t, clip\left(r_t(\theta_\pi), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right], \tag{12}$$

Algorithm 1: Training procedure of the RL-SeqISP

```
Input: Training set
Output: The agent including an actor network
        $\pi(;\theta_\pi)$ and a critic network $V(;\theta_V)$.
Given: hardware ISP $F_{ISP}(;)$, metric function
  $M(\cdot)$, an agent including an actor network $\pi(;\theta_\pi)$
  and a critic network $V(;\theta_V)$
repeat
    Randomly input a RAW image $I^R$;
    Randomly initialize ISP parameters $P_0$;
    $d\theta_\pi \leftarrow 0, d\theta_V \leftarrow 0$;
    for $t \in \{0, 1, ..., T-1\}$ do
        Update state: $\mathcal{S}_t = \{P_t, I_t = F_{ISP}(I^R; P_t)\}$;
        Calculate feedback score:
        $M(I_t) = M_{dt}\big(F_{dt}(I_t;\theta)\big)$;
        Forward action:
        $\mathcal{A}_t = \pi(\mathcal{S}_t;\theta_\pi), \mathcal{V}_t = V(\mathcal{S}_t;\theta_V)$;
        Optimize ISP parameters:
        $P_{t+1} = P_t + \mathcal{W}_t \times \mathcal{A}_t$;
        Re-update state:
        $\mathcal{S}_{t+1} = \{P_{t+1}, I_{t+1} = F_{ISP}(I^R; P_{t+1})\}$;
        Calculate Reward score:
        $\mathcal{R}_t = M(I_{t+1}) - M(I_t)$;
    end
    $R_T = 0$;
    for $i \in \{T-1, T-2, ..., 0\}$ do
        $R_i \leftarrow \mathcal{R}_i + \gamma R_{i+1}$;
        Gradient Backpropagation:
        $\Delta\theta_\pi \leftarrow \nabla_{\theta_\pi} L_{actor}(\theta_\pi)$;
        Gradient Backpropagation:
        $\Delta\theta_V \leftarrow \nabla_{\theta_V} L_{critic}(\theta_V)$;
    end
    Update $\theta_\pi$ with $\Delta\theta_\pi$ and $\theta_V$ with $\Delta\theta_V$;
until Terminal condition is satisfied;
```

where ϵ denote a hyperparameter (set to 0.2 following (Schulman et al. 2017)), $\hat{A}_t$ indicate an advantage function (*i.e.*, $\hat{A}_t = (R_t - \mathcal{V}_t)$), and $r_t(\theta_\pi)$ is an objective function to constrain on the size of the actor update, respectively. For detailed descriptions, please refer to (Schulman et al. 2017).

Experiments

Image Signal Processing Pipelines

Taking a Bayer-format RAW image I^R as input, after the RL-SeqISP model outputs the optimal ISP parameters P^* for I^R, the hardware ISP processes I^R into the RGB image I^* through six stages under the ISP parameter setting of P^*, formulated as $I^* = F_{ISP}(I^R, P^*)$.

Experiment Settings

Downstream tasks. We conduct experiments on a wide range of downstream tasks. Our main purpose is to verify on two visual analysis tasks, *i.e.*, instance segmentation and object detection. In addition, to better simulate the aesthetic of RGB images from the human perspective, we use the image quality assessment task (IQA).

Dataset settings. For instance segmentation and object detection, due to the lack of a large-scale RAW-RGB image dataset, we employ a simulation method (Kim et al. 2012) to modify MSCOCO dataset (Lin et al. 2014), which is also been adopted by other comparable sota methods.

For more implementation details (including network settings, training settings and evaluation details on each task), please refer to the *Supplementary Material*.

Comparison with State-of-the-arts

Evaluation on Instance Segmentation Task. For instance segmentation, we use the modified MSCOCO dataset as stated above. We randomly select a small subset from the MSCOCO training set (including 1,000 images) to train our model. We adopt the Mask-RCNN (He et al. 2017) pre-trained on the MSCOCO set, and freeze its parameters in the process of training and evaluation. During training, we adopt Mean Average Precision (mAP) as the feedback metric. Finally, we evaluate the model on the MSCOCO validation set and report the mAP score. Note that for this task, we adopt a hardware ISP with 20 ISP parameters.

We compare the RL-SeqISP model with many explicit ISP tuning methods, *i.e.*, hardware ISP with fixed empirical parameters provided by experts (*i.e.*, "Default parameters"), "Expert-tuned", "Blockwise-tuned" (Nishimura et al. 2018), "Hardware-tuned" (Mosleh et al. 2020) and "Attention-aware" (Qin et al. 2022). Note that "Expert-tuned" refers to the process in which six image processing experts from mobile phone manufacturers tune ISP parameters for each RAW image and obtain RGB images using Qualcomm Spectra 580 ISP sensor. Then, they vote to determine the optimal RGB image and its corresponding ISP parameters. Experimental results are illustrated in column 3 of Tab. 1. Specifically, our "RL-SegISP" exceeds other methods by a large margin, exceeding 0.11∼0.41 on mAP. It is worth pointing out that, compared with the latest method "Attention-aware" (Qin et al. 2022) which requires 10,000 images for training, our method uses only 1/10 images of (Qin et al. 2022) to outperform (Qin et al. 2022) by 0.11 with mAP. That effectively confirms the advantages of sequential parameter optimization even under fewer training samples. In addition, compared with "Expert-tuned", we exceed 0.17 on mAP, indicating that sequential parameter optimization guided by downstream task feedback is more suitable for instance segmentation than from a human subjective perspective.

Evaluation on Object Detection Task. To verify the generalization of the proposed RL-SeqISP model, we evaluate our method on the object detection task. For this task, we use the same modified MSCOCO dataset as the instance segmentation task and a YOLOv3 model (Redmon and Farhadi 2018) pre-trained on the MSCOCO dataset. For training, we aim to minimize the YOLO loss L_{yolo} (*i.e.*, maximize negative YOLO loss "$-L_{yolo}$"). For evaluation, we report the mAP score. Note that for this task, our hardware ISP also includes 20 ISP parameters.

The experimental results in column 4 of Tab. 1 show a similar phenomenon to the instance segmentation, *i.e.*, we

Methods	Training images	Instance Segmentation	Object Detection
		mAP	mAP
Default parameters	-	0.22	0.34
Expert-tuned	-	0.46	0.56
Blockwise-tuned (Nishimura et al. 2018)	-	-	0.20
Hardware-tuned (Mosleh et al. 2020)	-	0.32	0.39
Attention-aware (Qin et al. 2022)	10,000	0.52	0.61
RL-SepISP (Ours)	1,000	**0.63**	**0.64**

Table 1: Quantitative comparison with state-of-the-art methods on MSCOCO benchmark on two visual analysis downstream tasks, *i.e.*, instance segmentation and object detection, respectively.

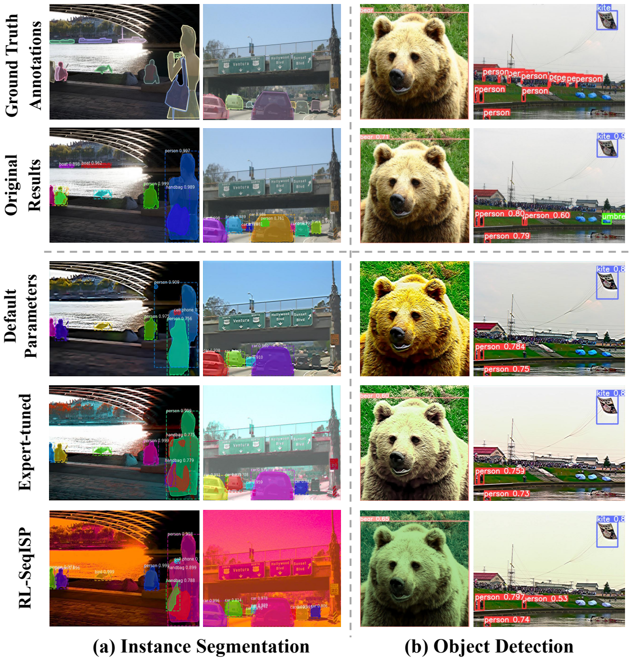
consistently outperform other methods by 0.03$\sim$0.44 on mAP, proving the generalization, robustness, and superiority of the proposed RL-SeqISP in the object detection task.

Evaluation on Image Quality Assessment Task. In order to assess image quality directly from a human perspective, we further conduct experiments on the IQA task. For the IQA task, since there are no open-source IQA datasets, we use a SONY IMX766 CMOS sensor to collect 252 RAW images, including 61 instances shot in the laboratory and 191 instances shot outdoors. Then, several imaging experts tune the ISP parameters for each RAW image and use the Qualcomm Spectra 580 ISP to generate the reference RGB image, *i.e.*, ground truth (GT), named "Expert-tuned (GT)". Note that it takes about 12 hours for an expert to tune the parameters for each image, which is time-consuming. In addition, human experts also provide a fixed set of uniform default ISP parameters for all images based on their experience, named "Default parameters". For training and evaluation, we divide the dataset into 177 and 75 images (about 7:3). Before training, we adopt data argumentation to the training set, such as adding noise, reducing brightness, *etc*. In this task, we can directly compute the evaluation metrics without relying on any network, including Structure Similarity Index Measure (SSIM) and Peak Signal-to-Noise Ratio (PSNR) (Hore and Ziou 2010). Specifically, we adopt SSIM as a feedback metric during training, and we report both metrics during evaluation. Note that for IQA, there exist 48 ISP parameters for the hardware ISP.

The results in Tab. 2 show that our RL-SeqISP model exceeds the "Default parameters" on $SSIM$ and $PSNR$ by 0.051 and 3.324, respectively, proving that our model can better simulate the sequential tuning process of human experts to obtain better ISP parameters.

Visualization Analyse

In Fig. 3 (a), we visualize the results of the proposed RL-SeqISP model and other methods in the instance segmentation. We can find that the RGB images generated by the RL-SeqISP model can produce segmentation results closer to "Ground Truth Annotations" than other methods. Specifically, the results generated by "Default parameters" and "Human-tuned" have either misclassification of occluded instances (*e.g.*, "handbag" in the first column) or omission segmentation of small instances (*e.g.*, smaller "car" instances in the second column), while these instances can be segmented with high confidence scores in RGB images generated by the RL-SeqISP model. The main reason is that human experts tend to subjectively perceive the entire image and are not sensitive to changes in the detail regions when tuning ISP parameters, while our RL-SeqISP model can customize the ISP parameters through the feedback of the downstream task and the sequential parameter optimization, so as to pay attention to more detail regions. In addition, it is worth mentioning that although there are differences between RGB images generated by RL-SeqISP and human visual perception, RL-SeqISP performs better in instance segmentation, which proves that RL-SeqISP is more suitable for instance segmentation than from a human subjective perspective.



(a) Instance Segmentation

(b) Object Detection

Figure 3: Visualization examples of the results on two visual analysis downstream tasks, *i.e.*, instance segmentation (a) and object detection (b). The first two rows provide the ground truth annotations and the evaluation results of the original RGB images for the MSCOCO validation set. The last three lines show the results of RGB images generated by different explicit ISP tuning methods, *i.e.*, fixed default parameters, expert-tuned parameters, and ours, respectively.

In Fig. 3 (b), we visualize the results of the object detection. The RGB images generated by "Default parameters" and "Human-tuned" also have situations of missed or false detection, with low confidence to correctly detected objects. Our RL-SeqISP can detect objects well, which proves the generalization of RL-SeqISP in the object detection.

In Fig. 4, we visualize some RGB images of different methods on the IQA task. To compare the details more intu-

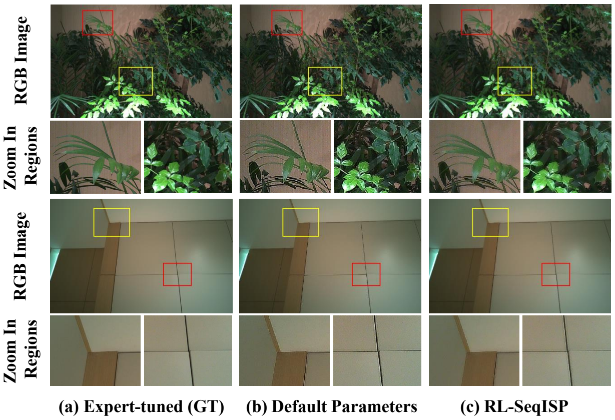


Figure 4: Visualization examples on Image Quality Assessment Task. (a), (b) and (c) indicate the expert-tuned images, images generated by fixed default parameters, and images generated by our model, respectively.

Methods	$SSIM$	$PSNR$
Expert-tuned (GT)	1.000	-
Default parameters	0.901	31.904
RL-SepISP (Ours)	**0.952**	**35.228**

Table 2: Experimental results on IMX766 CMOS sensor dataset for image quality assessment task.

itively, we zoom in on two small regions for each image. We observe that the RGB images generated by our RL-SeqISP model are closer to "Expert-tuned (GT)" images compared to using fixed default parameters. In particular, the images under default parameters face distortion issues in edges and detail textures, which we can avoid well.

Ablation Study

We conduct rich ablation studies on three tasks. Specifically, we first design a set of ablation experiments to validate two key components proposed for RL-SeqISP, *i.e.*, Dual-branch Parameter-guided Feature Fusion Module (DPFFM) and Dynamic Parameter Optimization Module (DPOM). Then we analyze the effects of the max optimization step. In addition, we also visualized some sequential optimization results during evaluation in the *Supplementary Material.*

Effects of Different Components. As shown in Tab. 3, we remove DPFFM and DPOM to construct our baseline, named "Baseline". Then, we add DPFFM and DPOM modules to verify their effectiveness. In these experiments, T is set to 4. Experimental results show that after adding DPFFM to "Baseline", our model has achieved significant improvements in the instance segmentation task and object detection task, and has made slight progress in the IQA task, which proves that DPFFM can improve feature quality through parameter-guided feature fusion and thus improve model performance. Further, when adding DPOM to the "Baseline", we are pleasantly surprised to see that the experimental results are somewhat improved compared to the "Baseline" on each task, demonstrating the effectiveness of DPOM in preventing ISP parameters from convergence to local optimum. Finally, we add both DPPFM and DPOM to build a complete RL-SeqISP model, and the results show that the RL-SeqISP model can achieve optimal results.

Methods	Instance Segmentation	Object Detection	IQA	
	mAP	mAP	$SSIM$	$PSNR$
Baseline	0.49	0.48	0.91	32.82
w/o DPOM	0.52	0.55	0.93	33.70
w/o DPFFM	0.54	0.53	0.93	33.50
RL-SeqISP	**0.58**	**0.64**	**0.95**	**35.23**

Table 3: Ablation studies for different components (*i.e.*, DPFFM and DPOM) on three downstream tasks.

Max Step	Instance Segmentation	Object Detection	IQA	
	mAP	mAP	$SSIM$	$PSNR$
T=1	0.53	0.54	0.92	32.82
T=2	0.57	0.52	0.91	32.65
T=4	0.58	**0.64**	**0.95**	**35.23**
T=8	**0.63**	0.53	0.94	34.05
T=16	0.60	0.53	0.94	34.61

Table 4: The impact of Max Optimization Step T on three downstream tasks.

Effect of the Max Optimization Step T. As illustrated in Tab. 4, we explore the effect of the maximum optimization step T during the training and validation phases. As T gradually increases, the performance of the model gradually improves, proving that sequential parameter optimization does help to learn better ISP parameters and produce more downstream task-friendly RGB images. For object detection, our model achieves the best results when T is set to 4. For the instance segmentation task, since the dense prediction task requires higher image quality in the detail regions, more steps of optimization are required to reach the optimal parameters, such as 8 steps. Besides, for the IQA task, the trend of experimental results is similar to that of the other two downstream tasks, with the best experimental results obtained when the maximum step is set to 4. Note that the model performance degrades when T is set too large for the three downstream tasks. Therefore, in this paper, T is set to 8, 4 and 4 for instance segmentation, object detection and IQA, respectively.

Conclusions

In this paper, we propose a novel RL-based Sequential ISP parameter tuning model (*i.e.*, RL-SeqISP) that mimics the stepwise tuning procedure of human experts. To prevent the ISP parameters from converging to the local optimum, we also propose a dynamic parameter optimization module. Extensive experiments on three downstream tasks show that our RL-SeqISP model outperforms other explicit ISP tuning methods, demonstrating the effectiveness and generalization of the RL-SeqISP model.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2022ZD0118502), the National Natural Science Foundation of China (No. 62072027, No. 62202470, No. 62202015) and the Beijing Natural Science Foundation (Grant No. 4224093).

References

Baina, J.; and Dublet, J. 1995. Automatic focus and iris control for video cameras.

Bajaj, C.; Wang, Y.; Yang, Y.; and Zheng, Y. 2021. Recursive Self-Improvement for Camera Image and Signal Processing Pipeline. *arXiv preprint arXiv:2111.07499*.

Bardenet, R.; Brendel, M.; Kégl, B.; and Sebag, M. 2013. Collaborative hyperparameter tuning. In *International conference on machine learning*, 199–207.

Board, C. A. 2017. IEEE Standard for Camera Phone Image Quality. *IEEE Std 1858-2016*, 1–146.

Chen, C.; Chen, Q.; Xu, J.; and Koltun, V. 2018. Learning to see in the dark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3291–3300.

Dong, X.; Xu, W.; Miao, Z.; Ma, L.; Zhang, C.; Yang, J.; Jin, Z.; Teoh, A. B. J.; and Shen, J. 2022. Abandoning the Bayer-filter to see in the dark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17431–17440.

Furuta, R.; Inoue, N.; and Yamasaki, T. 2019. Pixelrl: Fully convolutional network with reinforcement learning for image processing. *IEEE Transactions on Multimedia*, 22(7): 1704–1719.

Gharbi, M.; Chaurasia, G.; Paris, S.; and Durand, F. 2016. Deep joint demosaicking and denoising. *ACM Transactions on Graphics*, 35(6): 1–12.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–778.

Hore, A.; and Ziou, D. 2010. Image quality metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*, 2366–2369.

Ignatov, A.; Timofte, R.; Liu, S.; Feng, C.; Bai, F.; Wang, X.; Lei, L.; Yi, Z.; Xiang, Y.; Liu, Z.; et al. 2023. Learned smartphone ISP on mobile GPUs with deep learning, mobile AI & AIM 2022 challenge: report. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, 44–70.

Ignatov, A.; Van Gool, L.; and Timofte, R. 2020. Replacing mobile camera isp with a single deep learning model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 536–537.

Kim, S. J.; Lin, H. T.; Lu, Z.; Süsstrunk, S.; Lin, S.; and Brown, M. S. 2012. A new in-camera imaging model for color computer vision and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12): 2289–2302.

Lee, S.; Kim, N.; Jeong, K.; Paek, I.; Hong, H.; and Paik, J. 2015. Multiple moving object segmentation using motion orientation histogram in adaptively partitioned blocks for high-resolution video surveillance systems. *Optik-International Journal for Light and Electron Optics*, 126(19): 2063–2069.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755.

Mosleh, A.; Sharma, A.; Onzon, E.; Mannan, F.; Robidoux, N.; and Heide, F. 2020. Hardware-in-the-loop end-to-end optimization of camera image processing pipelines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7529–7538.

Nishimura, J.; Gerasimow, T.; Sushma, R.; Sutic, A.; Wu, C.-T.; and Michael, G. 2018. Automatic ISP image quality tuning using nonlinear optimization. In *2018 25th IEEE International Conference on Image Processing*, 2471–2475.

Osband, I.; Van Roy, B.; Russo, D. J.; Wen, Z.; et al. 2019. Deep Exploration via Randomized Value Functions. *J. Mach. Learn. Res.*, 20(124): 1–62.

Qin, H.; Han, L.; Wang, J.; Zhang, C.; Li, Y.; Li, B.; and Hu, W. 2022. Attention-Aware Learning for Hyperparameter Prediction in Image Processing Pipelines. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIX*, 271–287.

Ratnasingam, S. 2019. Deep camera: A fully convolutional neural network for image signal processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 0–0.

Redmon, J.; and Farhadi, A. 2018. Yolov3: an incremental improvement. arXiv e-prints. *arXiv preprint arXiv:1804.02767*.

Robidoux, N.; Capel, L. E. G.; Seo, D.-e.; Sharma, A.; Ariza, F.; and Heide, F. 2021. End-to-end high dynamic range camera pipeline optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6297–6307.

Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schwartz, E.; Giryes, R.; and Bronstein, A. M. 2018. Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing*, 28(2): 912–923.

Shin, U.; Lee, K.; and Kweon, I. S. 2022. DRL-ISP: Multi-Objective Camera ISP with Deep Reinforcement Learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 7044–7051.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*.

Tseng, E.; Yu, F.; Yang, Y.; Mannan, F.; Arnaud, K. S.; Nowrouzezahrai, D.; Lalonde, J.-F.; and Heide, F. 2019. Hyperparameter optimization in black-box image processing using differentiable proxies. *ACM Trans. Graph.*, 38(4): 27–1.

Wang, Z.; Zhang, J.; Lin, M.; Wang, J.; Luo, P.; and Ren, J. 2020. Learning a reinforced agent for flexible exposure bracketing selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1820–1828.

Watkins, C. J.; and Dayan, P. 1992. Q-learning. *Machine learning*, 8: 279–292.

Xu, K.; Li, Y.; Han, B.; Zhang, X.; Liu, X.; and Ai, J. 2018. A Low-power Computer Vision Engine for Video Surveillance. In *2018 IEEE International Conference on Integrated Circuits, Technologies and Applications*, 92–93.

Yahiaoui, L.; Hughes, C.; Horgan, J.; Deegan, B.; Denny, P.; and Yogamani, S. 2019. Optimization of ISP parameters for object detection algorithms. *Electronic Imaging*, 2019(15): 44–1.

Yogatama, D.; and Mann, G. 2014. Efficient transfer learning method for automatic hyperparameter tuning. In *Artificial intelligence and statistics*, 1077–1085.

Yu, K.; Li, Z.; Peng, Y.; Loy, C. C.; and Gu, J. 2021a. Reconfigisp: Reconfigurable camera image processing pipeline. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4248–4257.

Yu, K.; Wang, X.; Dong, C.; Tang, X.; and Loy, C. C. 2021b. Path-restore: Learning network path selection for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10): 7078–7092.

Zhang, R.; Guo, L.; Huang, S.; and Wen, B. 2021a. Rellie: Deep reinforcement learning for customized low-light image enhancement. In *Proceedings of the 29th ACM international conference on multimedia*, 2429–2437.

Zhang, R.; Zhu, J.; Zha, Z.; Dauwels, J.; and Wen, B. 2021b. R3l: Connecting deep reinforcement learning to recurrent neural networks for image denoising via residual recovery. In *2021 IEEE International Conference on Image Processing (ICIP)*, 1624–1628. IEEE.