Hierarchical Bilevel Learning with Architecture and Loss Search for Hadamard-based Image Restoration

Guijing Zhu[1] , **Long Ma**[1,3] , **Xin Fan**[2] and **Risheng Liu**[2*]
[1]School of Software Technology, Dalian University of Technology
[2]International School of Information Science & Engineering, Dalian University of Technology
[3]Peng Cheng Laboratory
{zgjing, longma}@mail.dlut.edu.cn, {rsliu, xin.fan}@dlut.edu.cn,

Abstract

In the past few decades, Hadamard-based image restoration problems (e.g., low-light image enhancement) attract wide concerns in multiple areas related to artificial intelligence. However, existing works mostly focus on heuristically defining architecture and loss by the engineering experiences that came from extensive practices. This way brings about expensive verification costs for seeking out the optimal solution. To this end, we develop a novel hierarchical bilevel learning scheme to discover the architecture and loss simultaneously for different Hadamard-based image restoration tasks. More concretely, we first establish a new Hadamard-inspired neural unit to aggregate domain knowledge into the network design. Then we model a triple-level optimization that consists of the architecture, loss and parameters optimizations to deliver a macro perspective for network learning. Then we introduce a new hierarchical bilevel learning scheme for solving the built triple-level model to progressively generate the desired architecture and loss. We also define an architecture search space consisting of a series of simple operations and an image quality-oriented loss search space. Extensive experiments on three Hadamard-based image restoration tasks (including low-light image enhancement, single image haze removal and underwater image enhancement) fully verify our superiority against state-of-the-art methods.

1 Introduction

Image restoration with deep learning methods has developed rapidly in recent years, and Convolutional Neural Networks (CNNs) have been used widely in many image restoration tasks such as low-light image enhancement (LLIE), single image haze removal (SIHR) and underwater image enhancement (UIE) [Ma *et al.*, 2022; Ma *et al.*, 2021; Qin *et al.*, 2020; Fu and Cao, 2020]. The performance of CNNs largely depends on the network architecture and training loss. In order to obtain networks with high performance, the automatic search of these two components has been explored recently.

1.1 Related Work

Network Architecture Search. Great progress has been made in designing Neural Architecture Search (NAS) [Liu *et al.*, 2021a; Liu *et al.*, 2021e]. Early attempts [Zoph and Le, 2016] used evolutionary algorithms to optimize the architectures and parameters of the networks. Another approach [Tan *et al.*, 2020] used reinforcement learning to train a meta-controller to find architectures. However, using evolutionary algorithms and reinforcement learning required a large amount of computations and made the search process more inefficient. Recently, differentiable search methods [Liu *et al.*, 2018] were proposed. They converted the super-network into a differentiable form by introducing architectural parameters which could be optimized by gradient descent and the search cost could be reduced largely. The above algorithms have achieved competitive performance in various tasks. However, these existing methods tend to obtain unsatisfactory visual results in complicated scenarios. The main reason is that although these methods make use of NAS technique, they lacked physical constraints in principle. As a result, the architectures searched by the aforementioned methods often failed when meeting with complex degraded images because they couldn't capture the inherent image structure without the help of principle prior knowledge.

Training Loss Search. Training loss is a critical component affecting performance. And most existing methods employed hand-craft losses with human experience. ℓ_2 loss is a popular training loss in image restoration problem. However, it often brings about blurry in the images. Perceptual loss [Johnson *et al.*, 2016] calculates ℓ_2 distance between the activations of the hidden layers of a pre-trained network. However, using this loss could result in a large misalignment between training loss and evaluation metrics. So, many works combine multiple losses with carefully designed trade-off weights for complementation [Lv *et al.*, 2018]. To find a better loss combination or form, training loss search has also aroused great interest very recently [Liu *et al.*, 2021b; Li *et al.*, 2019b; Li *et al.*, 2022]. A cross-entropy based loss space was proposed in face recognition [Li *et al.*, 2019b]. It conducted the search process including a set of cross-entropy losses for face recognition. But the types of candidate losses were too

*Corresponding author.

(a) LLIE (b) SIHR (c) UIE

Figure 1: **Hadamard-based image restoration**. We present a Hadamard-based model which searches network architecture and training loss for restoring a given image captured on different Hadamard-based tasks such as (a) Low-light Image Enhancement (LLIE), (b) Single Image Haze Removal (SIHR), and (c) Underwater Image Enhancement (UIE).

simple and of the same type. And it brought about a large misalignment between training loss and target metrics. The work in [Li *et al.*, 2022] designed an elementary search space which was composed of primitive mathematical operators to accommodate the heterogeneous tasks and evaluation metrics. However, it abandoned the current existing loss evaluation system completely. More importantly, the searched combination result of these operators changed with randomness and the search space was too sparse, which may lead to unpleasing results. By contrast, our loss space is image quality-oriented and more relevant to the evaluation metrics. As a result, it is suitable for a lot of image restoration tasks.

1.2 Our Contributions

To address the above issues, we propose a novel hierarchical bilevel learning scheme with architecture and loss search for Hadamard-based image restoration. A Hadamard-inspired neural unit is firstly built to convert the domain knowledge into the network layer. Then we construct a novel triple-level modeling for architecture and loss search by characterizing the latent connection among loss, architecture and parameters. Considering the difficulty of solving the triple-level model, we define a new hierarchical bilevel learning scheme with an approximate solving process. A series of experiments verify our effectiveness on different tasks and the result is shown in Figure 1. In summary, our contributions can be concluded as

- We develop a new Hadamard-inspired neural unit to integrate task-related domain knowledge into the network design to provide performance support for different Hadamard-based image restoration tasks.
- We supply a general triple-level modeling perspective for joint architecture and loss search. It can accurately depict the latent correspondence between loss, architecture, and parameters.
- To solve the triple-level model, we propose a hierarchical bilevel learning scheme that can alternatively estimate the architecture and loss to progressively generate the desired solutions.

2 The Proposed Method

In this section, we first construct our Hadamard-inspired neural unit. And then, we establish a new triple-level modeling for learning architecture and training loss. After that, we propose a stage-wise learning strategy for solving the built triple-level model. Finally, we introduce the architecture search space and image quality-oriented loss search space.

2.1 Hadamard-inspired Neural Unit

A set of image degradations can be represented as clear images with a degradation factor in Hadamard product form. And it can be formulated as $\mathbf{y} = \mathbf{u} \otimes \mathbf{v}$, where $\mathbf{y}$ denotes the degraded image, $\mathbf{u}$ denotes the original image and $\mathbf{v}$ denotes the degradation factor. $\otimes$ represents the element-wise multiplication. And $\mathbf{u} = \mathbf{y} \oslash \mathbf{v}$ can be derived by performing inverse process, where $\oslash$ represents the element-wise division. By applying this rule to feature layer, we establish a Hadamard-inspired Neural Unit (HNU) to perform task-related domain knowledge. The designed HNU is consists of two feature enhancement blocks and is used for estimating $\mathbf{u}$ and $\mathbf{v}$ to realize degradation removal progressively. We cascade multiple HNUs and embed them into feature convert modules to form the optimization process of the model. The l-th intermediate HNU can be formulated as

$$\texttt{HNU}_l : \begin{cases} \mathcal{F}_{\hat{\mathbf{u}}}^{l+1} = \mathcal{R}_{\mathbf{u}}(\mathcal{F}_{\mathbf{u}}^{l}), \mathcal{F}_{\hat{\mathbf{v}}}^{l+1} = \mathcal{R}_{\mathbf{v}}(\mathcal{F}_{\mathbf{u}}^{l}), \\ \mathcal{F}_{\mathbf{u}}^{l+1} = \mathcal{F}_{\hat{\mathbf{u}}}^{l+1} \oslash \mathcal{F}_{\hat{\mathbf{v}}}^{l+1}, l \geq 1, \end{cases} \tag{1}$$

for the l-th block $\texttt{HNU}_l$, the feature $\mathcal{F}_{\mathbf{u}}^{l}$ obtained from the previous block (the $\mathcal{F}_{\mathbf{u}}^{1}(l=1)$ is the feature generated by FCM) is inputted into two feature enhancement blocks $\mathcal{R}_{\mathbf{u}}$ and $\mathcal{R}_{\mathbf{v}}$ which will be searched to get $\mathcal{F}_{\hat{\mathbf{u}}}^{l+1}$ and $\mathcal{F}_{\hat{\mathbf{v}}}^{l+1}$. Then the intermediate enhanced feature $\mathcal{F}_{\mathbf{u}}^{l+1}$ is generated by executing the element-wise division $\oslash$ on $\mathcal{F}_{\hat{\mathbf{u}}}^{l+1}$ and $\mathcal{F}_{\hat{\mathbf{v}}}^{l+1}$. The multiple cascaded HNUs framework is built by applying this equation. We design a Feature Convert Module (FCM) for conversion between image and feature domain. The whole framework is generated by embedding them into FCMs. The component of feature convert module also needs to be searched. The search space of the above modules can be found in Section 2.4. The overall framework is shown in Figure 2.

2.2 Triple-Level Modeling

The crucial factors of performance for learning-based methods lie in how to define the architecture and training loss. As described above, we have clearly introduced a new block for Hadamard-based image restoration. Here we provide a

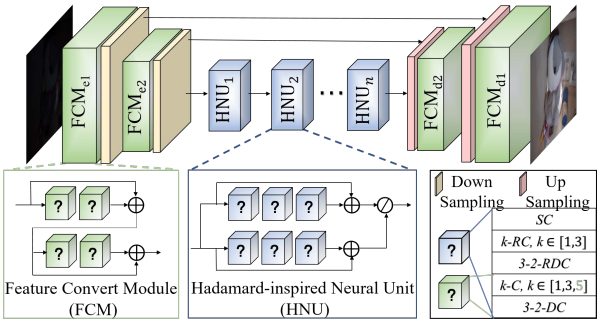


Figure 2: The overall network architecture.

general perspective for depicting the overall learning process. The formulation can be written as

$$\min_{\boldsymbol{\beta}\in\mathcal{B}} \mathcal{L}^{\mathtt{C}}_{\mathtt{val}}(\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\omega}),\ s.t.\ (\boldsymbol{\alpha},\boldsymbol{\omega})\in\mathcal{S}_{\boldsymbol{\alpha}}(\boldsymbol{\beta})\times\mathcal{S}_{\boldsymbol{\omega}}(\boldsymbol{\beta},\boldsymbol{\alpha}),\quad (2)$$

where

$$\begin{aligned}&\mathcal{S}_{\boldsymbol{\alpha}}(\boldsymbol{\beta}) := \arg\min_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{L}^{\mathtt{Q}}_{\mathtt{val}}(\boldsymbol{\alpha},\boldsymbol{\omega};\boldsymbol{\beta}),\\ &s.t.\ \boldsymbol{\omega}\in\mathcal{S}_{\boldsymbol{\omega}}(\boldsymbol{\alpha},\boldsymbol{\beta}),\ \mathcal{S}_{\boldsymbol{\omega}}(\boldsymbol{\alpha},\boldsymbol{\beta}) := \arg\min_{\boldsymbol{\omega}} \mathcal{L}^{\mathtt{Q}}_{\mathtt{tr}}(\boldsymbol{\omega};\boldsymbol{\beta},\boldsymbol{\alpha}),\end{aligned}\quad (3)$$

where $\boldsymbol{\beta}$ represents the hyper-parameters (including trade-off weights of the losses and inner parameters when calculating a specific loss) that needs to be defined in the loss space $\mathcal{B}$. The variable $\boldsymbol{\alpha}$ represents the desired architecture determined in the defined search space $\mathcal{A}$. The variable $\boldsymbol{\omega}$ denotes the network parameters related to $\boldsymbol{\alpha}$. The loss function $\mathcal{L}^{\mathtt{C}}_{\mathtt{val}}$ represents the classification-driven loss used on validation dataset. The loss $\mathcal{L}^{\mathtt{Q}}_{\mathtt{val}}$ and $\mathcal{L}^{\mathtt{Q}}_{\mathtt{tr}}$ denote the image quality-oriented loss[1] used on validation and training dataset, respectively.

The above triple-level modeling actually builds an explicit relationship among training loss, network architecture, and network parameters. To be concrete, both of the loss and the architecture play a decisive role in the network parameters, i.e., the network parameters $\boldsymbol{\omega}$ is the variable for the loss and architecture search. When searching the architecture, the adopted loss function directly dictates the form of the searched architecture, that is, the architecture $\boldsymbol{\alpha}$ becomes the variable for the optimization of training loss.

From above, the selection of the validation loss is the key to the success of searching training loss and should be designed carefully. However, the image quality-oriented losses (such as VGG and LPIPS loss) are employed in our loss space and can't be used as validation loss for search. So which loss to use as the validation loss is the key point. Considering how to evaluate whether an image is clear or not, we find that classification network is able to distinguish the image from its degraded form, which is related to the image restoration tasks. As a result, we employ a classification loss as our validation loss to estimate the performance of the searched training loss more accurately. What's more, by setting the degraded image as the negative sample and setting the clear image as the positive sample, the network can extract abstract representation and high level information of natural images. This information is helpful for generating visually pleasing results. More specifically, for each Hadamard-based image restoration task, we train an image classifier to distinguish whether the image is the clear one or not. And then we use the classification loss as the validation loss to search training loss. The specially trained classifier can capture more task-related information, which could evaluate the searched losses more accurately.

[1]The specific forms about it can be found in Section 2.4.

Name	Expression	Variable
$\mathcal{L}_a$: ℓ_1	$\lambda_a * \|\mathbf{u}^\star - \mathbf{u}\|_1$	λ_a
$\mathcal{L}_b$: ℓ_2	$\lambda_b * \|\mathbf{u}^\star - \mathbf{u}\|_2$	λ_b
$\mathcal{L}_c$: Color	$\lambda_c * \sum_i \angle((\mathbf{u}^\star)_i, (\mathbf{u})_i)$	λ_c
$\mathcal{L}_d$: SSIM	$\lambda_d * l(\mathbf{u}^\star,\mathbf{u})^\alpha \cdot c(\mathbf{u}^\star,\mathbf{u})^\beta \cdot s(\mathbf{u}^\star,\mathbf{u})^\gamma$	λ_d
$\mathcal{L}_e$: LPIPS	$\lambda_e * \sum_l \frac{\sum_{h,w} \left\| w_l \odot \left(\mathcal{F}^l(\mathbf{u}^\star) - \mathcal{F}^l(\mathbf{u})\right)\right\|_2^2}{H_l W_l}$	λ_e
$\mathcal{L}_f$: VGG	$\lambda_f * \frac{\sum_{n=1} \left\| \mathcal{V}^l(\mathbf{u}^\star) - \mathcal{V}^l(\mathbf{u})\right\|_1}{N_l}$	λ_f, l
$\mathcal{L}_g$: TV	$\lambda_g * \frac{((\bigtriangledown_x \mathbf{u})^2 + (\bigtriangledown_y \mathbf{u})^2)}{H*W}$	λ_g
$\mathcal{L}_h$: Laplacian	$\lambda_h * \frac{(\bigtriangledown_x^2 \mathbf{u} + \bigtriangledown_y^2 \mathbf{u})}{H*W}$	λ_h

Table 1: The overall search space in the search phase for training loss. $\mathbf{u}$ is the restored image and $\mathbf{u}^\star$ is the ground truth.

2.3 Hierarchical Bilevel Learning Scheme

Actually, solving the above built triple-level model is a challenging issue because of the complex functional relationship. Here we propose a hierarchical bilevel solution scheme to convert the triple-level model into two bilevel models. Because bilevel optimization is more simple and has been researched many years compared to the triple-level problem [Liu *et al.*, 2022b; Liu *et al.*, 2022a; Liu *et al.*, 2021d; Liu *et al.*, 2021c]. By fixing the variable $\boldsymbol{\alpha}$ when optimizing the $\boldsymbol{\beta}$, we can obtain the following solution scheme

$$\begin{cases}\boldsymbol{\beta}^{t+1} = \min_{\boldsymbol{\beta}\in\mathcal{B}} \mathcal{L}^{\mathtt{C}}_{\mathtt{val}}(\boldsymbol{\beta},\boldsymbol{\omega};\boldsymbol{\alpha}^t),\ s.t.\ \boldsymbol{\omega}\in\mathcal{S}_{\boldsymbol{\omega}}(\boldsymbol{\alpha}^t,\boldsymbol{\beta}),\\ \boldsymbol{\alpha}^{t+1} = \min_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{L}^{\mathtt{Q}}_{\mathtt{val}}(\boldsymbol{\alpha},\boldsymbol{\omega};\boldsymbol{\beta}^{t+1}),\ s.t.\ \boldsymbol{\omega}\in\mathcal{S}_{\boldsymbol{\omega}}(\boldsymbol{\alpha},\boldsymbol{\beta}^{t+1}),\end{cases}\quad (4)$$

where t represents the alternative number. The first equation indicates the optimization process of $\boldsymbol{\beta}$. The classification loss $\mathcal{L}^{\mathtt{C}}_{\mathtt{val}}$ is used for validation. $\boldsymbol{\alpha}$ is fixed as its initial value, and the solving process of $\boldsymbol{\beta}$ is converted into a bilevel problem involving the trainable variable $\boldsymbol{\beta}$, $\boldsymbol{\omega}$ and the fixed variable $\boldsymbol{\alpha}$. The second equation indicates the optimization process of $\boldsymbol{\alpha}$. The $\mathcal{L}^{\mathtt{Q}}_{\mathtt{val}}$ is obtained via the solution procedure of $\boldsymbol{\beta}$. So the solving process of $\boldsymbol{\alpha}$ is converted into another bilevel problem involving the trainable variable $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$. What's more, in the solving process of the two bilevel problems, the gradients of weight parameters and architecture are computed using first-order approximation following standard differential NAS techniques [Liu *et al.*, 2018].

2.4 Search Space of Architecture and Loss

Training Loss Search Space. We design an image quality-oriented losses set as the training loss search space. The candidate losses include ℓ_1, ℓ_2, color [Wang *et al.*, 2019], SSIM, VGG [Johnson *et al.*, 2016], LPIPS [Zhang *et al.*, 2018], TV [Osher *et al.*, 2005] and Laplacian [Rosenfeld, 1976] loss. For each loss, we define the weighting coefficient as the parameter to be optimized. What's more, the VGG loss has

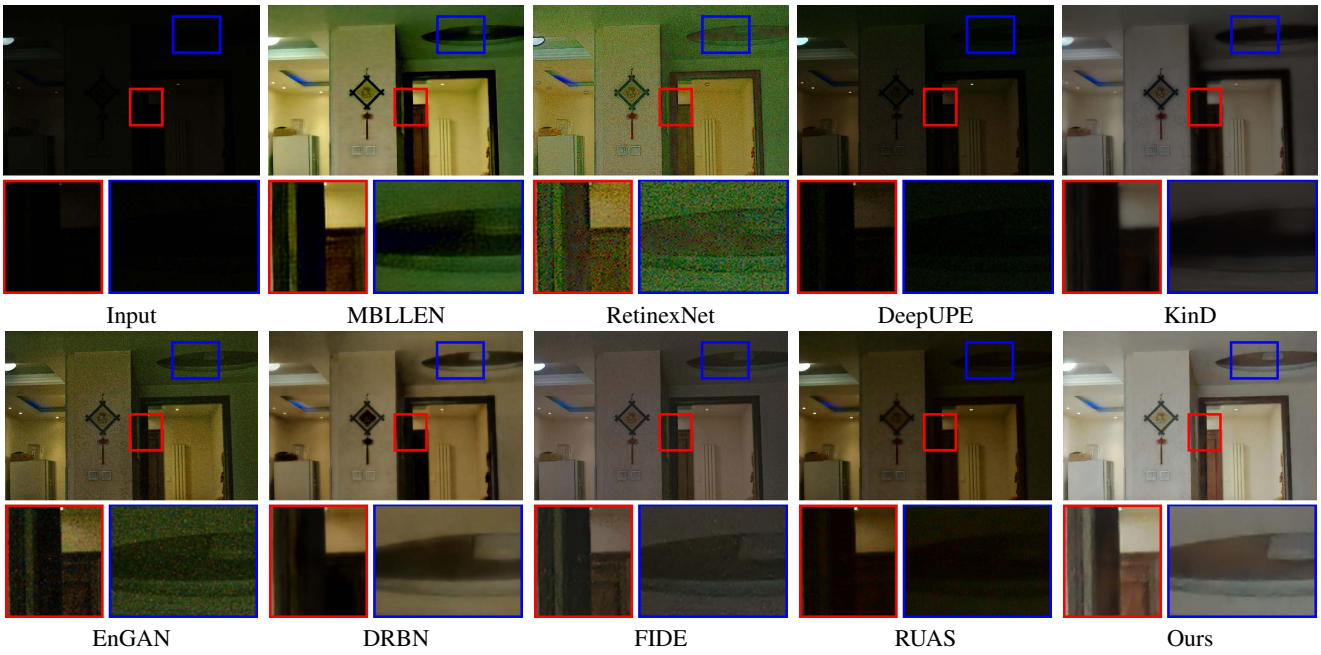


Figure 3: Visual comparison among state-of-the-art LLIE methods on LOL dataset.

	Methods	MBLLEN	RetinexNet	DeepUPE	KinD	EnGAN	DRBN	FIDE	RUAS	Ours
MIT	PSNR	17.4947	12.5591	20.8741	15.7552	15.1014	16.3702	16.2580	17.0056	**23.2378**
	SSIM	0.6218	0.5673	0.6958	0.6318	0.6001	0.5967	0.6095	0.6182	**0.8184**
	LPIPS	0.1787	0.2234	0.0937	0.1387	0.1218	0.1734	0.1769	0.1164	**0.0519**
LOL	PSNR	18.6313	16.3076	11.8956	20.2552	17.6844	18.2622	21.5797	14.9663	**24.1640**
	SSIM	0.7705	0.4362	0.4418	0.8382	0.6081	0.6688	0.7935	0.4986	**0.8663**
	LPIPS	0.2161	0.4997	0.2793	0.1079	0.2975	0.1508	0.1307	0.2432	**0.0751**

Table 2: Qualitative results of state-of-the-art methods and ours on the MIT-Adobe 5K and LOL datasets.

another kind of variable which is the selection of the hidden layers of the VGG network. We divide the layers of the VGG network into four groups in terms of feature scale, and the goal is to select one layer from each group. By employing a new variable for each group, we added the selection of VGG layers into loss search space. The detailed definition of each loss is shown in Table 1.

Architecture Search Space. According to different characteristics, we define search spaces for HNU and FCM respectively. The candidate operations include 1×1, 3×3 and 5×5 Convolution (1-C, 3-C and 5-C), 1×1 and 3×3 Residual Convolution (1-RC and 3-RC),3×3 Dilation Convolution with dilation rate of 2 (3-2-DC), 3×3 Residual Dilation Convolution with dilation rate of 2 (3-2-RDC), 1×1 and 3×3 Dense Convolution (1-DC and 3-DC), and Skip Connection (SC). The division info for HNU and FCM is shown in Figure 2.

3 Experimental Results

3.1 Implementation Details

For LLIE task, we used MIT-Adobe 5K [Bychkovsky *et al.*, 2011] and LOL Dataset [Wei *et al.*, 2018]. We sampled 600 low-light images randomly from the MIT-Adobe 5K dataset for searching and training, and 100 images for testing. As for the LOL Dataset, we randomly sampled 690 image pairs for searching and training, and the remaining 99 pairs for testing. For SIHR task, we used the dataset proposed by [Pan *et al.*, 2020]. And for UIE task, we sampled 712 image pairs from UIEB dataset [Li *et al.*, 2019a] for searching and training and the remaining 178 image pairs for testing. All the experiments were performed on a PC with a single TITAN X GPU. We used ResNet-50 to train our classifier. In searching phase, the maximum epoch was 30, the batch size was 1 and the initial learning rate was 0.0005. The momentum was 0.9 and the weight decay was 0.0003. The initiate weight of each loss in

Methods	PSNR	SSIM	LPIPS
DAD	20.2702	0.8050	0.1438
EPDN	20.6930	0.8232	0.0732
FFA	22.6119	0.8830	0.0726
GridDehazeNet	23.8632	0.8965	0.0677
MSBDN	23.8819	0.8965	0.0677
PSD	25.1914	0.9107	0.0675
PhysicsGAN	24.4496	0.8806	0.0623
Ours	**27.0509**	**0.9370**	**0.0484**

Table 3: Quantitative SIHR result of state-of-the-art methods.

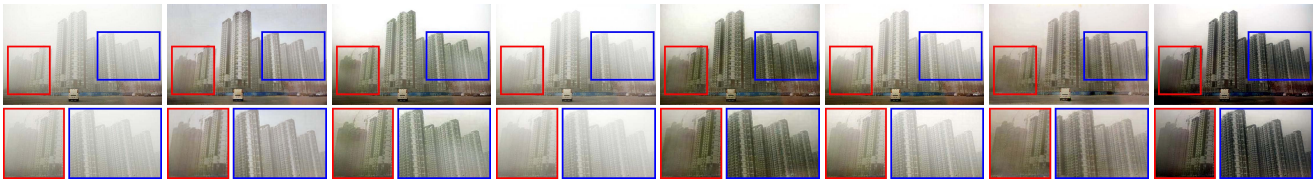


Figure 4: Visual comparison among state-of-the-art SIHR methods on the real hazy image.

Methods	UDCP	OCM	TSA	Fusion	AIO	WaterNet	UWCNN	FGAN	GLN	Ours
PSNR	11.6973	22.0781	14.2882	21.8875	13.3344	19.5888	17.5703	17.6067	21.0797	**23.0009**
SSIM	0.5060	0.5838	0.4851	0.8200	0.4850	0.8238	0.6347	0.5880	0.7548	**0.8630**
LPIPS	0.2635	0.2040	0.3187	0.1390	0.2851	0.1135	0.1911	0.2287	0.1179	**0.0822**

Table 4: Qualitative comparison of UIE task on UIEB dataset between our methods and state-of-the-art methods.

Figure 5: Visual comparison among state-of-the-art UIE methods on UIEB dataset.

loss search space was 1.0. As for the training phase (with the searched architecture and searched training loss), we set the learning rate as 0.0001 and the batch size was 1.

3.2 Comparison with State-of-the-arts

To evaluate our method roundly, we adopted PSNR, SSIM and LPIPS as our evaluation metrics for three tasks. To fully demonstrate the superiority of our method, we compared our method with recently proposed state-of-the-art approaches on LLIE, SIHR and UIE tasks respectively.

Evaluations on LLIE. The compared methods include MBLLEN [Lv *et al.*, 2018], RetinexNet [Wei *et al.*, 2018], DeepUPE [Wang *et al.*, 2019], KinD [Zhang *et al.*, 2019], EnGAN [Jiang *et al.*, 2021], DRBN [Yang *et al.*, 2020], FIDE [Xu *et al.*, 2020] and RUAS [Liu *et al.*, 2021e]. As shown in Figure 3, the compared methods such as DeepUPE and RUAS either couldn't remove noise or caused color shift. By contrast, our method obtained the highest performance in removing noise and enhancing the image and was superior to others in both the evaluation metrics and visual results.

Evaluations on SIHR. The compared methods contain DAD [Shao *et al.*, 2020], EPDN [Qu *et al.*, 2019], FFA [Qin *et al.*, 2020], GridDehazeNet [Liu *et al.*, 2019], MSBDN [Dong *et al.*, 2020], PSD [Chen *et al.*, 2021] and PhysicsGAN [Pan *et al.*, 2020]. The quantitative result was in Table 3 and the qualitative result was in Figure 4. As shown, most methods failed on the real hazy image. The reason is that these methods can't learn the inherent characteristics of the real hazy image. By comparison, our method was superior to others in both haze removal and detail recovery.

Evaluations on UIE. The compared methods include UDCP [Drews *et al.*, 2016], OCM [Li *et al.*, 2016], TSA [Fu *et al.*, 2017], Fusion [Ancuti *et al.*, 2012], AIO [Uplavikar *et al.*, 2019], WaterNet [Li *et al.*, 2019a], UWCNN [Li *et al.*, 2020], FGAN [Islam *et al.*, 2020] and GLN [Fu and Cao, 2020]. As showed in Table 4 and Figure 5, our method gained the highest scores and could enhance the challenging underwater image well while others failed to recover clear details.

4 Algorithmic Analyses

4.1 The Effect of Training Loss Search

To demonstrate the effect of training loss search. We conducted three groups of contrast experiments on LLIE task. (1): Employ MBLLEN as the basic model and its proposed loss as loss space. Use the experienced loss setting proposed by the author (MBLLEN) and compare it with the searched version (MBLLEN$^+$). (2): Employ UNET as the basic model and our proposed loss space as loss space. Set the weights as 1.0 for each candidate (UNET) and compare it with the searched version (UNET$^+$). (3): Using our searched architectures, set the weights as 1.0 for each candidate (Ours) and compare it with the searched version (Ours$^+$). The result is shown in Figure 6. As showed, the searched version obtained a higher performance on both PSNR and SSIM than another. And our proposed method obtained the highest performance.

4.2 The Effect of Architecture Search

We analyzed the performance among our model and different heuristically-designed architectures on LLIE task and they were trained using the searched loss. The result was shown in Table 5. As can be seen, the performance of the supernet and heuristically designed architectures was either unideal or couldn't reach the searched architectures. Briefly, these architectures may not effective enough. It is because these architectures did not integrate the task cues/principles. By comparison, our searched architecture realized the highest performance in all metrics. In a word, this experiment indicated the

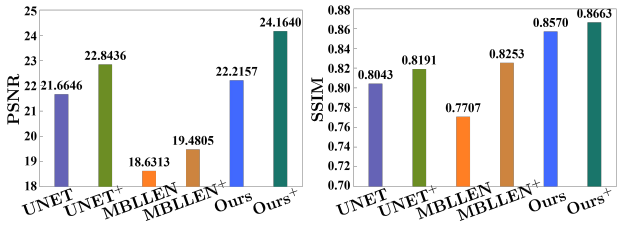


Figure 6: Quantitative results between naively determined and searched training loss on LOL dataset.

Model	PSNR	SSIM	LPIPS
Supernet	17.4412	0.7410	0.2447
1-C	21.3163	0.7708	0.1431
3-C	21.9472	0.8573	0.0947
1-RC	20.3591	0.6253	0.1938
3-RC	22.4196	0.8612	0.0816
Ours$_a$	22.9998	0.8186	0.0858
Ours	**24.1640**	**0.8663**	**0.0751**

Table 5: Quantitative results between naively determined architectures and our searched architecture on LOL dataset.

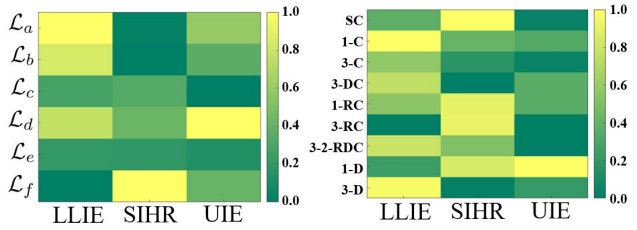
necessity of searching the architecture and the superiority of our searched architecture.

4.3 Analyzing the Search Strategy

Actually, the search strategy is a decisive factor for the searched architecture. Here, we made an evaluation on the search strategy. Ours$_a$: search training loss and architecture synchronously using the same classification loss $\mathcal{L}_{\texttt{val}}^{\texttt{C}}$ as the upper loss. In this way, architecture ($\boldsymbol{\alpha}$) and training loss ($\boldsymbol{\beta}$) are searched by the same optimal goal at the same time. Ours: divide the triple-level problem into two bilevel problems. Specifically, firstly fix the architecture ($\boldsymbol{\alpha}$) as the supernet and use the classification loss ($\mathcal{L}_{\texttt{val}}^{\texttt{C}}$) to search training loss, then use this searched loss ($\mathcal{L}_{\texttt{val}}^{\texttt{Q}}$) to search the architecture. As shown in Table 5, the second strategy achieved higher performance and gained a large advantage.

4.4 The Searched Results of Different Tasks

To better analyze our triple-level model for different tasks and explore the differences between tasks. We plotted the searched architectures and training loss of LLIE, SIHR and UIE tasks in Figure 7. For simplicity, we just selected one searched layer of the architectures from the HNU to show. It should be noted that the searched weights of TV and Laplacian loss are small to zero and don't show great differences among tasks. To focus on more useful information, we didn't draw them in the heatmap. And for easier observation, we softmax the searched loss weights of each task. As can be seen, the searched architectures were different in tasks. With the help of Hadamard principle, 1-C or 1-D was enough in this layer for LLIE and UIE task, while SC was searched in SIHR tasks. And the dominant loss also differed greatly in tasks, which were ℓ_1, VGG and SSIM loss for LLIE, SIHR and UIE respectively. As shown in Figure 8, we showed the feature processed by the searched layer in second group of the



(a) Losses (b) Architectures

Figure 7: The searched losses and architectures of different tasks.

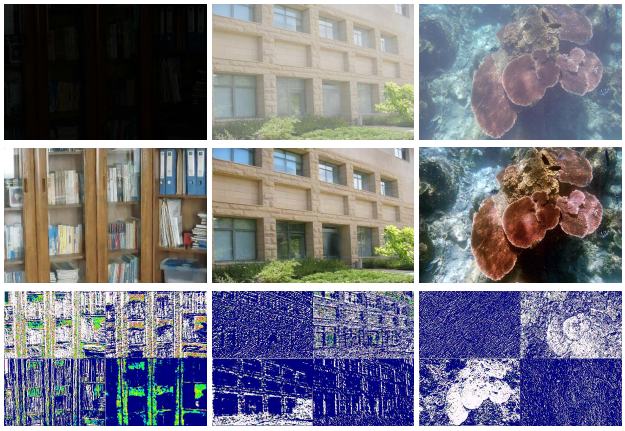


Figure 8: Comparing features of different tasks processed by their searched VGG layers. The first to third rows are the inputs, the restored results of our method and the corresponding features.

VGG network. And present the 34-th, 42-th,51-th and 60-th channel feature to a colormap. As can be seen, the perceptual features required of SIHR and UIE were similar, while that of LLIE focused more on architectural characteristics.

5 Concluding Remarks

We proposed a new framework to combine domain knowledge with a new search strategy to perform architecture and loss search for Hadamard-based image restoration. We first developed a Hadamard-inspired neural unit to explore task-related knowledge. And by cascading multiple units into feature convert modules, we can obtain the whole holistic structure of our enhancement network. And we designed an architecture search space and an image quality-oriented loss search space. Then we designed a hierarchical bilevel learning strategy to solve the triple-level problems including the architecture, loss and network parameters. Our experiments were performed on three image restoration tasks and we obtained new state-of-the-art results. In the future, we will apply our built triple-level model technique to different vision tasks.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Nos. 61922019, 61733002 and 62027826), the major key project of Peng Cheng Laboratory (No. PCL2021A12) and the Fundamental Research Funds for the Central Universities.

References

[Ancuti *et al.*, 2012] Cosmin Ancuti, Codruta Orniana Ancuti, Tom Haber, and Philippe Bekaert. Enhancing underwater images and videos by fusion. In *CVPR*, pages 81–88, 2012.

[Bychkovsky *et al.*, 2011] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR*, pages 97–104, 2011.

[Chen *et al.*, 2021] Zeyuan Chen, Yangchao Wang, Yang Yang, and Dong Liu. Psd: Principled synthetic-to-real dehazing guided by physical priors. In *CVPR*, pages 7180–7189, 2021.

[Dong *et al.*, 2020] Hang Dong, Jinshan Pan, Lei Xiang, Zhe Hu, Xinyi Zhang, Fei Wang, and Ming-Hsuan Yang. Multi-scale boosted dehazing network with dense feature fusion. In *CVPR*, pages 2157–2167, 2020.

[Drews *et al.*, 2016] Paulo LJ Drews, Erickson R Nascimento, Silvia SC Botelho, and Mario Fernando Montenegro Campos. Underwater depth estimation and image restoration based on single images. *IEEE CGA*, 36(2):24–35, 2016.

[Fu and Cao, 2020] Xueyang Fu and Xiangyong Cao. Underwater image enhancement with global–local networks and compressed-histogram equalization. *SPIC*, 86:115892, 2020.

[Fu *et al.*, 2017] Xueyang Fu, Zhiwen Fan, Mei Ling, Yue Huang, and Xinghao Ding. Two-step approach for single underwater image enhancement. In *ISPACS*, pages 789–794, 2017.

[Islam *et al.*, 2020] Md Jahidul Islam, Youya Xia, and Junaed Sattar. Fast underwater image enhancement for improved visual perception. *IEEE RAL*, 5(2):3227–3234, 2020.

[Jiang *et al.*, 2021] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE TIP*, 30:2340–2349, 2021.

[Johnson *et al.*, 2016] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016.

[Li *et al.*, 2016] ChongYi Li, JiChang Guo, RunMin Cong, YanWei Pang, and Bo Wang. Underwater image enhancement by dehazing with minimum information loss and histogram distribution prior. *IEEE TIP*, 25(12):5664–5677, 2016.

[Li *et al.*, 2019a] Chongyi Li, Chunle Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong, and Dacheng Tao. An underwater image enhancement benchmark dataset and beyond. *IEEE TIP*, 29:4376–4389, 2019.

[Li *et al.*, 2019b] Chuming Li, Xin Yuan, Chen Lin, Minghao Guo, Wei Wu, Junjie Yan, and Wanli Ouyang. Am-lfs: Automl for loss function search. In *ICCV*, pages 8410–8419, 2019.

[Li *et al.*, 2020] Chongyi Li, Saeed Anwar, and Fatih Porikli. Underwater scene prior inspired deep underwater image and video enhancement. *Pattern Recognition*, 98:107038, 2020.

[Li *et al.*, 2022] Hao Li, Tianwen Fu, Jifeng Dai, Hongsheng Li, Gao Huang, and Xizhou Zhou. Autoloss-zero: Searching loss functions from scratch for generic tasks. In *CVPR*, 2022.

[Liu *et al.*, 2018] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *ICLR*, 2018.

[Liu *et al.*, 2019] Xiaohong Liu, Yongrui Ma, Zhihao Shi, and Jun Chen. Griddehazenet: Attention-based multi-scale network for image dehazing. In *ICCV*, pages 7314–7323, 2019.

[Liu *et al.*, 2021a] Risheng Liu, Jiaxin Gao, Jin Zhang, Deyu Meng, and Zhouchen Lin. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE TPAMI*, 2021.

[Liu *et al.*, 2021b] Risheng Liu, Zi Li, Xin Fan, Chenying Zhao, Hao Huang, and Zhongxuan Luo. Learning deformable image registration from optimization: perspective, modules, bilevel training and beyond. *IEEE TPAMI*, 2021.

[Liu *et al.*, 2021c] Risheng Liu, Xuan Liu, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A value-function-based interior-point method for non-convex bi-level optimization. In Marina Meila and Tong Zhang, editors, *ICML*, volume 139, pages 6882–6892, 2021.

[Liu *et al.*, 2021d] Risheng Liu, Yaohua Liu, Shangzhi Zeng, and Jin Zhang. Towards gradient-based bilevel optimization with non-convex followers and beyond. *NeurIPS*, 2021.

[Liu *et al.*, 2021e] Risheng Liu, Long Ma, Jiaao Zhang, Xin Fan, and Zhongxuan Luo. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In *CVPR*, pages 10561–10570, 2021.

[Liu *et al.*, 2022a] Risheng Liu, Long Ma, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. Task-oriented convex bilevel optimization with latent feasibility. *IEEE TIP*, 2022.

[Liu *et al.*, 2022b] Risheng Liu, Pan Mu, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A general descent aggregation framework for gradient-based bi-level optimization. *IEEE TPAMI*, 2022.

[Lv *et al.*, 2018] Feifan Lv, Feng Lu, Jianhua Wu, and Chongsoon Lim. Mbllen: Low-light image/video enhancement using cnns. In *BMVC*, volume 220, page 4, 2018.

[Ma *et al.*, 2021] Long Ma, Risheng Liu, Jiaao Zhang, Xin Fan, and Zhongxuan Luo. Learning deep context-sensitive decomposition for low-light image enhancement. *IEEE TNNLS*, 2021.

[Ma *et al.*, 2022] Long Ma, Tengyu Ma, Risheng Liu, Xin Fan, and Zhongxuan Luo. Toward fast, flexible, and robust low-light image enhancement. In *CVPR*, 2022.

[Osher *et al.*, 2005] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

[Pan *et al.*, 2020] Jinshan Pan, Jiangxin Dong, Yang Liu, Jiawei Zhang, Jimmy Ren, Jinhui Tang, Yu-Wing Tai, and Ming-Hsuan Yang. Physics-based generative adversarial models for image restoration and beyond. *IEEE TPAMI*, 43(7):2449–2462, 2020.

[Qin *et al.*, 2020] Xu Qin, Zhilin Wang, Yuanchao Bai, Xiaodong Xie, and Huizhu Jia. Ffa-net: Feature fusion attention network for single image dehazing. In *AAAI*, volume 34, pages 11908–11915, 2020.

[Qu *et al.*, 2019] Yanyun Qu, Yizi Chen, Jingying Huang, and Yuan Xie. Enhanced pix2pix dehazing network. In *CVPR*, pages 8160–8168, 2019.

[Rosenfeld, 1976] Azriel Rosenfeld. *Digital picture processing*. 1976.

[Shao *et al.*, 2020] Yuanjie Shao, Lerenhan Li, Wenqi Ren, Changxin Gao, and Nong Sang. Domain adaptation for image dehazing. In *CVPR*, pages 2808–2817, 2020.

[Tan *et al.*, 2020] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, pages 10781–10790, 2020.

[Uplavikar *et al.*, 2019] Pritish M Uplavikar, Zhenyu Wu, and Zhangyang Wang. All-in-one underwater image enhancement using domain-adversarial learning. In *CVPRW*, pages 1–8, 2019.

[Wang *et al.*, 2019] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *CVPR*, pages 6849–6857, 2019.

[Wei *et al.*, 2018] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *BMVC*, 2018.

[Xu *et al.*, 2020] Ke Xu, Xin Yang, Baocai Yin, and Rynson WH Lau. Learning to restore low-light images via decomposition-and-enhancement. In *CVPR*, pages 2281–2290, 2020.

[Yang *et al.*, 2020] Wenhan Yang, Shiqi Wang, Yuming Fang, Yue Wang, and Jiaying Liu. From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement. In *CVPR*, pages 3063–3072, 2020.

[Zhang *et al.*, 2018] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

[Zhang *et al.*, 2019] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. In *ACM MM*, pages 1632–1640, 2019.

[Zoph and Le, 2016] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.