Hardware-aware Low Light Image Enhancement on Edge

Sowmya Vajrala[1,✉], Sravanth Kodavanti[1], Srinivas Soumitri Miriyala[1]

[1]Samsung Research Institute Bangalore, India

Abstract

Low light image enhancement is a critical task in computer vision and photography that is often entangled with noise and blur. This renders the traditional image signal processing ineffective compared to the advances in deep learning. However, as deep learning continues to evolve, its migration towards edge and embedded systems necessitates the design of networks that are not only accurate but also efficient for deployment on resource constrained devices. To this end, this work presents novel mobile-friendly networks for low-light image enhancement obtained with network surgery, hardware-aware search space design and a training-free, search-strategy agnostic scheme for fast neural architecture search. The best model resulted in 3-fold improvement in on-device latency when profiled on Galaxy S24, with marginal drop in image quality for low-light image enhancement tasks.

1. Introduction

Mobile imaging systems rely on a tightly engineered Image Signal Processing (ISP) pipeline to transform raw sensor measurements into visually pleasing photographs under strict latency, memory, and power constraints [1, 8]. In modern smartphones, features such as night photography [18] operate under extremely low illumination, where photon shot noise, motion blur, and limited dynamic range severely degrade image quality [15] at the time of capture. To compensate, the ISP performs a sequence of deterministic operations: multi-frame fusion, denoising, demosaicing, color correction, tone mapping, and sharpening, many of which must execute within a few hundred milliseconds and within a tightly bounded memory footprint [1, 9]. In this regime, even modest increases in model complexity can translate into unacceptable delays, thermal throttling, or battery drain.

Although deep neural networks have demonstrated remarkable capability for low-light image enhancement (LLIE) in academic benchmarks, their direct integration into commercial ISP pipelines remains non-trivial [11, 23]. The strong need for hardware-aware design, therefore, arises from a practical gap: models optimized purely for accuracy often fail to meet deployment requirements, while conventional ISP blocks alone struggle to match the perceptual improvements delivered by learning-based methods [1, 10]. Bridging this gap between algorithmic sophistication and deployment feasibility is essential for translating advances in low-light enhancement into reliable, real-time mobile photography [8, 25].

Within this broader context, our work focuses on a specific and practically constrained setting: a commercial mobile ISP pipeline in which the overall processing pipeline [13] is fixed, and only a single enhancement module positioned after demosaicing and before final tone and color refinement admits learning-based optimization. This module operates as a proprietary enhancement block whose objective is implicitly defined by the surrounding ISP stages and perceptual tuning requirements [1].

In this work, we operate within a fixed commercial ISP and do not redesign the end-to-end imaging pipeline or replace traditional stages with a fully neural alternative. The enhancement module is optimized under proprietary data constraints, without relying on publicly curated low-light datasets. Consequently, our objective is not to report gains on academic LLIE benchmarks under standardized protocols. Rather, we address the realistic industrial problem of improving and deploying a pre-trained enhancement network within a rigid, legacy ISP framework, where task definition, data distribution, and evaluation protocols are governed by product constraints. Our scenario involves a fixed deployment target and tight time-to-market constraints that preclude multi-month search-and-train cycles. Consequently, there exists no established comparison protocol for this class of deployment-centric optimization problems. The contribution of this work is thus methodological: we reformulate architecture adaptation as a post-training, hardware-aware optimization problem tailored to an industrial ISP, rather than as a model discovery exercise in the

✉v.lahari@samsung.com

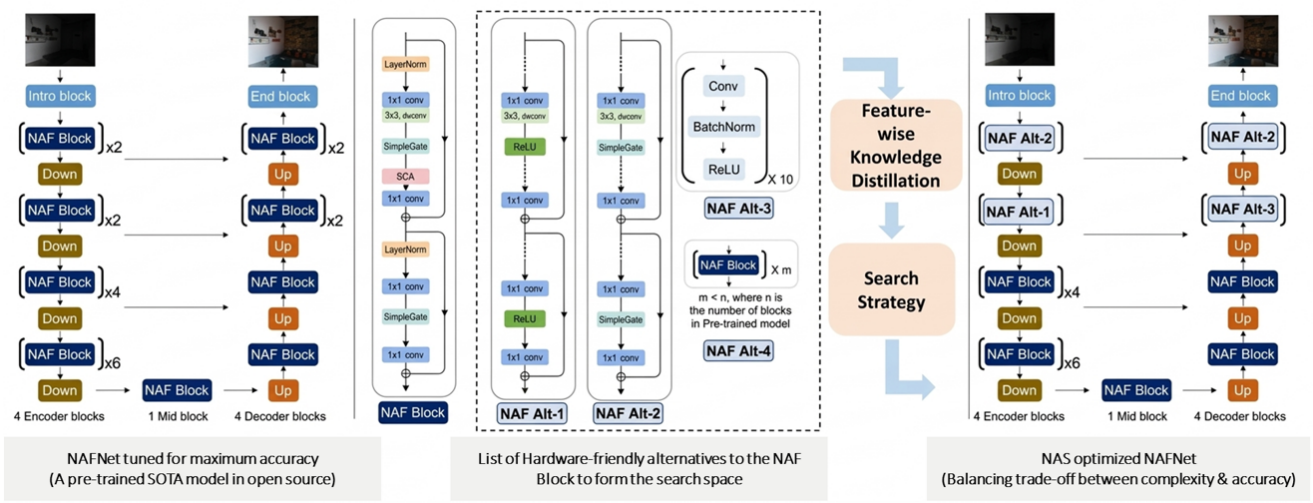

Figure 1. Proposed method for optimizing the pre-trained foundation models. In this case, we optimize the NAFNet [6] for low-light enhancement which is performed with hardware-aware search space design and training-free search strategy.

conventional academic sense. In this process,

- First, we formalize the deployment of an AI-based enhancement block within a commercial mobile ISP as a constrained optimization problem, explicitly delineating the boundaries between fixed pipeline stages and the learnable module.
- Second, we introduce a hardware-aware, post-training network reconfiguration strategy (see Figure 1) that combines sensitivity-driven network surgery, feature-wise knowledge distillation for block-level alternatives, and multi-objective Bayesian optimization [7] to balance image quality and on-device latency.
- Third, we provide an empirical analysis of latency–quality trade-offs on a commercial mobile NPU, including block-level profiling, search space design under compiler constraints, and Pareto-optimal configurations validated on-device.

We emphasize that we do not claim a new state-of-the-art LLIE algorithm, a universal NAS framework, or superiority on public benchmarks. Instead, we present a principled and reproducible methodology for bridging high-quality low-light enhancement models with real-world mobile deployment constraints, an increasingly critical yet underexplored problem at the intersection of computer vision and embedded systems.

2. Related Works

Low-light image enhancement. There are many significant works [17] that focus on resolving the noise and blur artefacts together with LLIE using an end-to-end model. While such an approach has the advantage of end-to-end deep learning, it reduces the human-interpretability in the

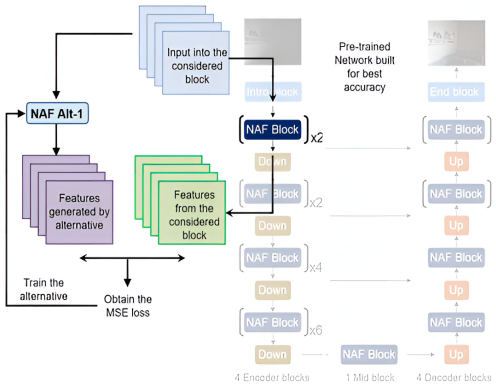

Figure 2. Feature-wise Knowledge Distillation, where the knowledge learnt by a single NAF block is distilled into its alternative (Alt) instead of end-to-end distillation of NAFNet [6].

ISP pipeline. LLNet proposed by Lore et al (2017) is a prominent example of this approach [20]. Other important methods that follow the suit are attention guided enhancement [21], Learning to see in the dark [3], kindling the darkness [35], and EnglightenGAN [12]. To improve the understanding of the end-to-end approach, networks based on Retinex theory [16] were introduced, where the networks first learn to decompose the image into reflectance and illumination and then focus on improving the illumination component. Deep Retinex Decomposition by Wei et al (2018) [27] and Retinexformer by Cai et al (2023) [2] are the representatives for the Retinex theory.

Deployment on mobile devices. Deploying deep learning models for low-light enhancement on smartphones in-

troduces strict latency, memory, and power constraints. Model compression and post-training quantization are commonly used to enable real-time inference, but for perceptually sensitive tasks such as LLIE, aggressive simplification can amplify noise, distort color, and degrade fine details. While efficient architectures and hand-tuned designs alleviate some of these issues, their effectiveness remains task-dependent under extremely low-light conditions. Neural Architecture Search (NAS) has been widely adopted to design efficient models by jointly optimizing accuracy and computational complexity. Conventional NAS frameworks define a search space, apply a search strategy (e.g., gradient-based, evolutionary, or Bayesian), and repeatedly train candidate architectures for evaluation. Although techniques such as supernets and proxy models reduce the computational burden, most NAS approaches still require substantial retraining and are primarily geared toward model discovery rather than post-training deployment adaptation. In contrast, our setting assumes a pre-trained enhancement network within a fixed ISP pipeline and focuses on hardware-aware reconfiguration under proprietary constraints, rather than designing a new architecture from scratch.

3. Formulation

This section formalizes the proposed deployment-centric optimization strategy. We begin by describing the MobileISP pipeline, baseline NAFNet [6] configuration, and its forward inference characteristics, followed by sensitivity-driven block analysis, construction of hardware-aware surrogates via feature-wise distillation, and finally the multi-objective Bayesian optimization [7] framework used to select the optimal configuration under device constraints.

3.1. Mobile ISP and Deployment Constraints

Mobile ISP pipeline includes multi-frame acquisition, sensor-domain corrections, demosaicing, color processing, tone mapping, and detail refinement. These stages are hardware-accelerated, validated across product generations, and organized with fixed interfaces. Unlike academic prototypes, this production pipeline offers limited architectural flexibility: most stages execute on CPU/GPU/DSP units with predefined numerical formats, while selected components are offloaded to dedicated accelerators. This rigidity ensures robustness and real-time guarantees, but constrains how learning-based models can be integrated.

Low-light operation further amplifies these constraints. Brightness amplification increases noise and accentuates motion artifacts, while deployment must remain latency-critical. Although physics-inspired approaches such as Retinex-based decomposition have shown promise in benchmarks, their iterative structure, global receptive fields, or dynamic computation patterns are difficult to reconcile with tile-based execution and static NPU graphs. Commercial systems therefore, adopt a hybrid strategy: the classical ISP backbone is preserved, and a single learning-based enhancement block is inserted at a designated stage.

In our setting, the enhancement module operates after demosaicing and before final tone and color refinement. A full-resolution 4K frame (3840×2160) is partitioned into overlapping 384×384 tiles to satisfy memory limits. Each tile is processed sequentially through legacy ISP stages and then forwarded to the NPU for enhancement. The per-tile latency accumulates across approximately 60 tiles per frame, making the runtime highly sensitive to the latency of the enhancement block. For a base network with 1.7M parameters (approximately 6.8 MB in FP32), the measured latency is about 150 ms per tile, which scales unfavorably for 4K inference. Even with pipeline overlap, such scaling reveals a fundamental bottleneck: per-tile efficiency directly determines user-perceived responsiveness.

The enhancement backbone considered here is NAFNet [6], selected for its convolutional structure, depth-wise operations, and compiler-friendly design relative to heavier attention-based models. Nevertheless, even this lightweight architecture becomes computationally demanding under tiled 4K deployment. Straightforward quantization often introduces visible color distortions and noise amplification, while retraining or conventional NAS approaches require substantial compute budgets and repeated optimization cycles, which is impractical under proprietary data and industrial time-to-market constraints.

An additional challenge arises from the behavior of mobile NPUs. Latency does not correlate monotonically with parameter count or FLOPs; operator fusion, tensor alignment, and compiler optimizations significantly influence runtime. In certain cases, modest structural changes or channel reallocation can reduce latency despite similar nominal complexity. Therefore, theoretical metrics alone are insufficient, and hardware profiling must be incorporated into the optimization loop.

Finally, deployment occurs in an agile industrial environment where model refinement and inference optimization proceed concurrently. Image-quality improvements must respect strict latency budgets, and hardware-aware modifications must preserve perceptual fidelity across large validation sets. This co-design requirement differentiates commercial ISP optimization from purely academic model development.

In summary, integrating AI-based low-light enhancement within a commercial mobile ISP entails rigid pipeline boundaries, tile-based high-resolution inference, hardware-specific runtime behavior, and strict time-to-market constraints. These factors motivate a deployment-centric, hardware-aware reconfiguration framework tailored specifically to the enhancement block, which we formalize in the subsequent section.

3.2. Baseline Architecture and its Inference

The base enhancement model is derived from NAFNet [6], a U-Net style encoder–decoder architecture composed of repeated NAF blocks. Each NAF block consists of Layer Normalization (LN), depth-wise convolution, channel attention, and a lightweight gated linear unit approximation. Let the input tile be denoted by $x \in \mathbb{R}^{H\times W\times 3}$, where $H = W = 384$. The network implements a mapping

$$\hat{y} = f(x;\theta), \tag{1}$$

where θ denotes the set of learned parameters and $\hat{y}$ is the enhanced RGB tile.

The custom configuration considered in this work follows the block distribution

$$[2\text{-}2\text{-}4\text{-}6] - 1 - [1\text{-}1\text{-}2\text{-}2],$$

corresponding to four encoder stages, a single middle stage, and four decoder stages. Specifically, encoder stages 1–4 contain 2, 2, 4, and 6 NAF blocks respectively; the bottleneck contains 1 block; and decoder stages contain 1, 1, 2, and 2 blocks respectively. For deployment reconfiguration, we treat each stage as a modular unit, resulting in a 9-dimensional structural decision space. Let B_i denote the block group at stage i, with $i \in \{1,\dots,9\}$.

During forward inference on device, each tile is propagated through these stages sequentially. For a given stage i, the transformation is

$$h_i = B_i(h_{i-1};\theta_i), \tag{2}$$

where $h_0 = x$ and $\theta_i \subset \theta$. The cumulative latency of the network on the NPU is not purely proportional to FLOPs but depends on tensor shapes, operator fusion opportunities, and memory reuse. Let $\mathcal{L}_i$ denote the measured on-device latency of stage i. The total tile latency is

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{9} \mathcal{L}_i + \mathcal{L}_{\text{overhead}}, \tag{3}$$

where $\mathcal{L}_{\text{overhead}}$ accounts for memory transfers and scheduling overhead.

Profiling revealed that latency contributions vary significantly across stages due to resolution differences and channel widths. Therefore, block-level intervention is both necessary and non-uniform in impact.

3.3. Sensitivity-Guided Structural Diagnosis

To identify blocks amenable to structural modification, we perform sensitivity analysis using SNIP (Single-shot Network Pruning). For each stage i, we compute the saliency score

$$S_i = \left\| \frac{\partial \mathcal{L}_{\text{task}}}{\partial \theta_i} \odot \theta_i \right\|_1, \tag{4}$$

Algorithm 1 Deployment-Aware Multi-Objective Bayesian Optimization

Require: Pre-trained base network $f(x;\theta)$, calibration set $\mathcal{D}_{\text{cal}}$, search space $\mathcal{Z}$, maximum iterations T
Ensure: Pareto-optimal configuration set $\mathcal{P}$

```
Compute SNIP sensitivity scores {S_i}_{i=1}^9
Profile stage-level latency {L_i}_{i=1}^9 on device
Construct surrogate sets {B̃_{i,j}} and distill via Eq. 5
Prune low-fidelity surrogates to define Z
Initialize dataset D_0 with random configurations
for t = 0 to T − 1 do
    Fit GP models for L(z) and ΔPSNR(z)
    Select z_{t+1} via EHVI maximization
    Assemble network f(x; θ(z_{t+1}))
    Measure latency L(z_{t+1}) on device
    Evaluate ΔPSNR(z_{t+1})
    Update dataset D_{t+1} = D_t ∪ {z_{t+1}}
end for
Extract Pareto front P from evaluated configurations
```

where $\mathcal{L}_{\text{task}}$ is the enhancement loss and $\odot$ denotes element-wise multiplication. High values indicate accuracy-critical stages, while low values suggest structural redundancy.

In parallel, each stage is independently profiled on the target NPU to measure its latency contribution at the specific feature resolution corresponding to that stage. The joint consideration of sensitivity S_i and latency $\mathcal{L}_i$ allows us to prioritize stages that are latency-dominant yet accuracy-insensitive for structural modification.

3.4. Hardware-Compatible Surrogate Construction via Feature Distillation

For each candidate stage i, we construct a set of hardware-friendly surrogate blocks $\{\tilde{B}_{i,j}\}_{j=1}^{n_i}$ by modifying normalization, convolutional structure, or channel allocation while preserving input–output dimensionality. For example, Layer Normalization may be replaced by Batch Normalization to enable compiler-level fusion. However, such replacements induce distribution shifts and accuracy degradation.

To mitigate this, we perform feature-wise knowledge distillation at the block level as show in Figure 2. For a given original block B_i and surrogate $\tilde{B}_{i,j}$, we optimize

$$\min_{\tilde{\theta}_{i,j}} \mathbb{E}_{x\sim\mathcal{D}_{\text{cal}}} \left\| B_i(h_{i-1}) - \tilde{B}_{i,j}(h_{i-1}) \right\|_2^2, \tag{5}$$

where $\mathcal{D}_{\text{cal}}$ is a small calibration subset and no ground truth targets are required. This produces a digital twin of the original block that approximates its feature transformation while being hardware-efficient.

Surrogates failing to meet a predefined PSNR tolerance relative to the base network are pruned from the search

space. The resulting candidate set defines a discrete alternative pool per stage.

3.5. Multi-Objective INLP Formulation

Let decision variable $z_i \in \{0, 1, \ldots, n_i\}$ indicate the choice of base block (0) or one of its n_i surrogates. The architecture configuration is denoted by

$$\mathbf{z} = (z_1, z_2, \ldots, z_9). \tag{6}$$

For a given configuration $\mathbf{z}$, the enhanced output is

$$\hat{y}_{\mathbf{z}} = f(x; \theta(\mathbf{z})), \tag{7}$$

where $\theta(\mathbf{z})$ represents the parameter set induced by the selected surrogates.

We define two objectives:

$$\Delta\text{PSNR}(\mathbf{z}) = \text{PSNR}_{\text{base}} - \text{PSNR}(\mathbf{z}), \tag{8}$$

$$\mathcal{L}(\mathbf{z}) = \sum_{i=1}^{9} \mathcal{L}_{i,z_i}. \tag{9}$$

The problem becomes a multi-objective integer nonlinear programming (INLP) problem:

$$\min_{\mathbf{z}\in\mathcal{Z}} \left(\Delta\text{PSNR}(\mathbf{z}),\ \mathcal{L}(\mathbf{z})\right), \tag{10}$$

where $\mathcal{Z} = \prod_{i=1}^{9}\{0, \ldots, n_i\}$.

3.6. Bayesian Optimization with EHVI

We adopt multi-objective Bayesian Optimization [7] (BO) with Gaussian Process (GP) surrogates modeling both objectives:

$$\mathcal{L}(\mathbf{z}) \sim \mathcal{GP}(\mu_L, k_L), \tag{11}$$

$$\Delta\text{PSNR}(\mathbf{z}) \sim \mathcal{GP}(\mu_A, k_A). \tag{12}$$

At iteration t, the next configuration is selected by maximizing the Expected Hypervolume Improvement (EHVI):

$$\mathbf{z}_{t+1} = \arg\max_{\mathbf{z}\in\mathcal{Z}} \mathbb{E}\left[\text{HVI}(\mathbf{z} \mid \mathcal{D}_t)\right], \tag{13}$$

where $\mathcal{D}_t$ is the set of evaluated configurations.

The overall optimization procedure is presented in Algorithm 1. This formulation transforms deployment-aware architecture adaptation into a structured, integer multi-objective optimization problem with hardware-in-the-loop evaluation, eliminating the need for repeated end-to-end retraining while preserving image enhancement fidelity.

4. Experiments and Results

We evaluate the proposed deployment-aware optimization framework on the commercial low-light enhancement pipeline described in Section 3.1, using the baseline NAFNet [6] configuration introduced in Section 3.2. All latency measurements correspond to on-device profiling on the target mobile NPU (Qualcomm's Snapdragon sm8550), and image quality is evaluated using PSNR relative to the proprietary ground truth used during training of the base model. Throughout this section, we refer to the multi-objective formulation in Eq.9, where the two objectives are the accuracy drop $\Delta\text{PSNR}(\mathbf{z})$ and the cumulative latency $\mathcal{L}(\mathbf{z})$.

Table 1. Network Sensitivity analysis for LLIE's base model.

Block	Difference	Latency (ms)
Enc 1	Illuminated areas are affected	68
Enc 2	Block artifacts	19
Enc 3	Textures are affected	11.7
Enc 4	Illuminated areas are affected	8.5
Mid	No visible differences	2
Dec 4	No visible differences	3
Dec 3	Very close to baseline	5
Dec 2	Over exposed	19
Dec 1	Over exposed	68

4.1. Sensitivity and Latency Profiling

We first analyze the structural importance of each stage B_i in the 9-dimensional configuration space. Table 1 reports the inferences drawn from SNIP-based sensitivity scores S_i alongside the measured stage-level latency $\mathcal{L}_i$ on the target NPU. To further validate the saliency interpretation, we replaced the learned weights of each stage independently with random weights while keeping the remaining network fixed, and visually inspected the resulting enhanced images. The qualitative degradations observed (see Figure 3), ranging from texture loss and over-exposure to minimal perceptual change, are consistent with the quantitative SNIP scores. Stages exhibiting low S_i yet high $\mathcal{L}_i$ were identified as prime candidates for structural modification.

Notably, early encoder stages operating at higher spatial resolutions contribute disproportionately to latency due to larger feature maps, while certain decoder stages exhibit low sensitivity despite measurable runtime cost. As the latency of middle blocks was minimal, they were untouched during the optimization. At the same time, the rest of the blocks were considered in the search space as all of them had similar sensitivity to accuracy. This joint analysis confirms that latency impact and accuracy importance are not uniformly correlated, thereby justifying the block-wise reconfiguration strategy rather than global compression.

4.2. Hardware-Aware Surrogate Space

For the selected stages, we constructed hardware-compatible surrogates $\tilde{B}_{i,j}$ as described in Section 3.3. Fig-

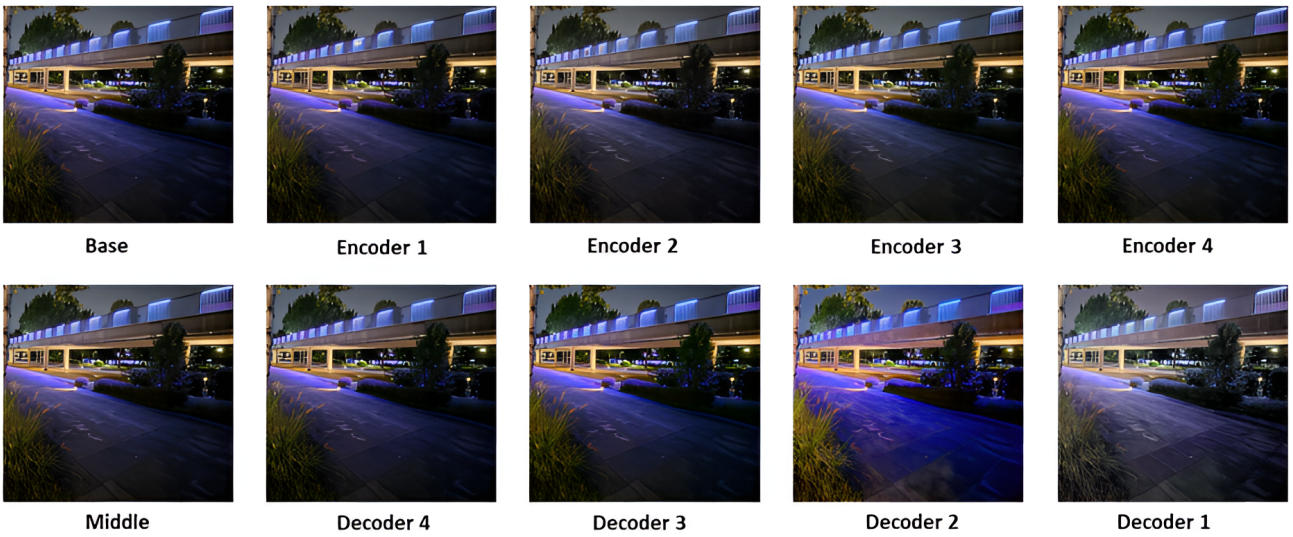

Figure 3. Qualitative degradations observed during Sensitivity analysis.

Table 2. Latency Comparison (measured on GS24) of Alternatives Across Different Encoder-Decoder Configurations.

Block Name	Base Block	Base Latency (ms)	Alt.	Alt. Latency (ms)
Encoder1-Decoder1	2 NAF	68	2xAlt1	25
			3xAlt1	26
			4xAlt1	45
			4xAlt2	20
			5xAlt2	28
			6xAlt2	31
			7xAlt2	42
			10xAlt3	17
			20xAlt3	39
			1xAlt4	20
			1xAlt5	8
Encoder2-Decoder2	2 NAF	19	1xNAF	10.0
			2xAlt1	6.5
			3xAlt1	7.5
			4xAlt1	10.0
			6xAlt2	11.0
			10xAlt3	8.5
			20xAlt3	11.3
Encoder3	4 NAF	11.7	6xAlt1	6.7
			6xAlt2	6.6
			15xAlt3	8.0
Encoder4	6 NAF	8.5	6xAlt1	4.4
			6xAlt2	4.6
			10xAlt3	4.8

ure 4 illustrates representative alternatives to the original NAF block, including normalization substitutions and convolutional restructuring designed to improve compiler fusion and memory access patterns. Table 2 enumerates the

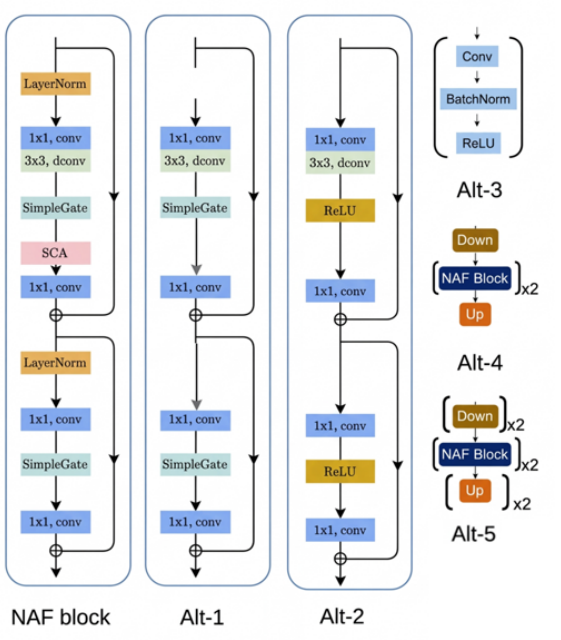

Figure 4. Hardware aware alternatives considered to replace the NAF Block. Exact alternatives considered for each encoder/decoder/middle block is specified in Table 2.

feasible alternatives retained after feature-wise distillation and PSNR tolerance filtering.

The distillation process ensured that each surrogate approximates the original block's feature transformation under the calibration distribution. Surrogates that exceeded a predefined ΔPSNR threshold when integrated into the network were discarded. The resulting search space $\mathcal{Z}$ thus contains only high-fidelity, hardware-aware candidates, reducing the combinatorial complexity while preserving structural diversity.

4.3. Multi-Objective Search and Pareto Analysis

Using the EHVI-based Bayesian Optimization [7] procedure outlined in Section 3.6, we explored the discrete space $\mathcal{Z}$ to approximate the Pareto frontier of Eq.8. Figure 5 presents the resulting Pareto front in the latency-accuracy plane. Each point corresponds to a configuration $\mathbf{z}$ evaluated on-device, with $\mathcal{L}(\mathbf{z})$ measured directly and $\Delta\text{PSNR}(\mathbf{z})$ computed relative to the base model.

The Pareto set reveals a clear trade-off: modest increases in ΔPSNR yield substantial reductions in latency, particularly when high-resolution encoder stages are replaced by hardware-friendly surrogates. Importantly, the optimization process required significantly fewer evaluations than exhaustive enumeration, demonstrating the sample efficiency.

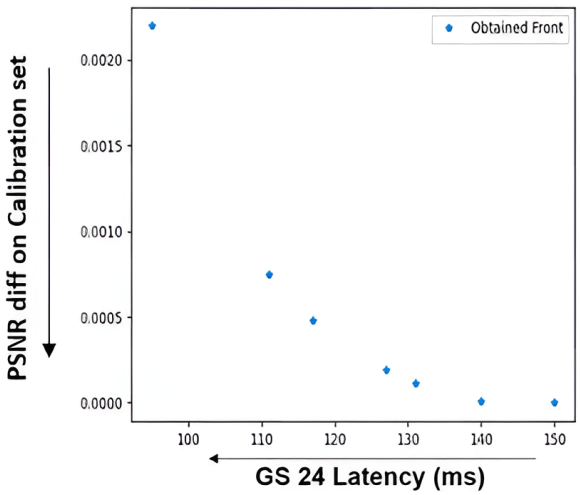

Figure 5. Pareto front obtained upon searching. PSNR diff is measured on a calibration set and latencies are measured on GS24

Table 3. Comparison of our best models with open source models.

Methods	Params (M)	PSNR	SSIM
SID [4]	7.76	14.35	0.436
3DLUT [34]	0.59	14.35	0.445
DeepUPE [24]	1.02	14.38	0.446
RF [14]	21.54	15.23	0.452
DeepLPF [22]	1.77	15.28	0.473
IPT [5]	115.31	16.27	0.504
UFormer [26]	5.29	16.36	0.771
RetinexNet [27]	0.84	16.77	0.56
Sparse [31]	2.33	17.2	0.64
EnGAN [12]	114.35	17.48	0.65
RUAS [19]	0.003	18.23	0.72
FIDE [28]	8.62	18.27	0.665
DRBN [30]	5.27	20.13	0.83
KinD [35]	8.02	20.86	0.79
Restormer [33]	26.13	22.43	0.823
MIRNet [32]	31.76	24.14	0.83
SNR-Net [29]	4.01	24.61	0.842
Retinexformer [2]	1.61	25.16	0.845
NAFNet (base) [6]	1.79	27.983	0.87
Iter1-best	1.72	18.161	0.613
Iter2-best	1.81	24.281	0.82
Iter3-best	1.79	14.05	0.308
Iter4-best	1.8	17.987	0.498

4.4. Comparison Across Architectures

Table 3 summarizes the quantitative comparison among (i) the base NAFNet [6] model, (ii) selected Pareto-optimal configurations from our search, and (iii) alternative SOTA

Table 4. Architecture details of best models obtained in each iteration along with latencies measured on GS24.

	Enc 1	Enc 2	Enc 3	Enc 4	Mid	Dec 4	Dec 3	Dec 2	Dec 1	Latency(ms)
Base model	**2xNAF**	2xNAF	4xNAF	6xNAF	1xNAF	1xNAF	1xNAF	2xNAF	**2xNAF**	150
Iter1-best	**1xAlt1**	2xNAF	4xNAF	6xNAF	1xNAF	1xNAF	1xNAF	2xNAF	**2xAlt1**	80
Iter2-best	**2xAlt1**	2xNAF	4xNAF	6xNAF	1xNAF	1xNAF	1xNAF	2xNAF	2xNAF	120
Iter3-best	**1xAlt4**	**2xAlt1**	4xNAF	6xNAF	1xNAF	1xNAF	1xNAF	**10xAlt3**	**1xAlt4**	55
Iter4-best	**4xAlt1**	2xNAF	**1xAlt1**	6xNAF	1xNAF	1xNAF	1xNAF	2xNAF	**4xAlt1**	100

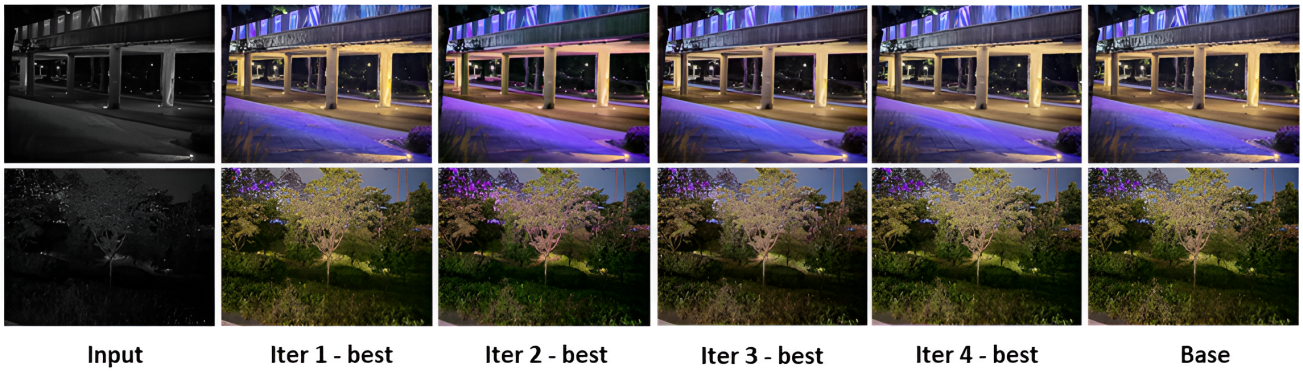

Figure 6. Visual comparison of images generated from Base and best model obtained in each iteration.

restoration backbones integrated into the same ISP pipeline. Beyond a single optimization run, we performed multiple iterative search cycles to incorporate feedback from scenario experts. Table 4 presents the architectural designs obtained from multiple iterations along with latencies.

The results indicate that hardware-aware reconfiguration consistently reduces on-device latency relative to the base model, even in cases where parameter count slightly increases. This observation reinforces the hardware-driven argument presented in Section 3.1 that parameter count and FLOPs alone are insufficient predictors of runtime on mobile NPUs. Compiler fusion opportunities and operator alignment can yield latency reductions despite marginally higher nominal complexity.

Compared to directly substituting the enhancement block with other SOTA architectures, the optimized configurations achieve a superior latency-quality trade-off within the fixed ISP pipeline. Architectures designed primarily for benchmark performance exhibit unfavorable latency scaling under tile-based inference, underscoring the necessity of deployment-aware adaptation.

4.5. Visual Quality Assessment

Figure 6 presents qualitative comparisons between the base model and representative Pareto-optimal configurations. The visual inspection confirms that latency-reduced models preserve structural details, illumination consistency, and color fidelity in low-light regions, with only minor degradations corresponding to small ΔPSNR values. The qualitative consistency across diverse low-light scenes indicates that the feature-wise distillation effectively preserves the functional behavior of the original NAF blocks, validating the surrogate construction strategy.

5. Conclusion

We presented a deployment-centric framework for optimizing low-light enhancement models within a commercial mobile ISP under strict hardware constraints. Instead of redesigning the enhancement algorithm, we formulated deployment as a constrained, hardware-aware reconfiguration problem applied to a pre-trained backbone. By integrating sensitivity-guided block analysis, feature-wise surrogate distillation, and multi-objective Bayesian Optimization with hardware-in-the-loop evaluation, the proposed approach enables efficient adaptation without repeated end-to-end retraining.

Experimental results on a commercial mobile NPU demonstrate that latency can be substantially reduced while preserving perceptual quality, and that runtime behavior is governed more by hardware characteristics than by parameter count or FLOPs alone. Although evaluated on a specific NPU, our hardware-agnostic methodology generalizes to diverse GPUs and accelerators. This work provides a practical methodology for bridging high-quality restoration networks with real-world ISP deployment constraints.

References

[1] Mark Buckler, Suren Jayasuriya, and Adrian Sampson. Reconfiguring the imaging pipeline for computer vision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 975–984, 2017. 1

[2] Yuanhao Cai, Hao Bian, Jing Lin, Haoqian Wang, Radu Timofte, and Yulun Zhang. Retinexformer: One-stage retinex-based transformer for low-light image enhancement. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12504–12513, 2023. 2, 7

[3] C. Chen, Q. Chen, J. Xu, and V. Koltun. Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3291–3300, 2018. 2

[4] Chen Chen, Qifeng Chen, Minh N. Do, and Vladlen Koltun. Seeing motion in the dark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 7

[5] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer, 2020. 7

[6] L. Chen, X. Chu, X. Zhang, and J. Sun. Simple baselines for image restoration. In *European Conference on Computer Vision (ECCV)*, pages 17–33, 2022. 2, 3, 4, 5, 7

[7] Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-objective bayesian optimization over high-dimensional search spaces. In *Uncertainty in Artificial Intelligence*, pages 507–517. PMLR, 2022. 2, 3, 5, 7

[8] Mauricio Delbracio, Damien Kelly, Michael S Brown, and Peyman Milanfar. Mobile computational photography: A tour. *Annual review of vision science*, 7(1):571–604, 2021. 1

[9] Felix Heide, Markus Steinberger, Yun-Ta Tsai, Mushfiqur Rouf, Dawid Pajak, Dikpal Reddy, Orazio Gallo, Jing Liu, Wolfgang Heidrich, Karen Egiazarian, et al. Flexisp: A flexible camera image processing framework. *ACM Transactions on Graphics (ToG)*, 33(6):1–13, 2014. 1

[10] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. Ai benchmark: Running deep neural networks on android smartphones. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 1

[11] Andrey Ignatov, Luc Van Gool, and Radu Timofte. Replacing mobile camera isp with a single deep learning model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 536–537, 2020. 1

[12] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, et al. Enlightengan: Deep light enhancement without paired supervision. *IEEE Transactions on Image Processing*, 30:2340–2349, 2021. 2, 7

[13] Hakki Can Karaimer and Michael S Brown. A software platform for manipulating the camera imaging pipeline. In *European Conference on Computer Vision*, pages 429–444. Springer, 2016. 1

[14] Satoshi Kosugi and Toshihiko Yamasaki. Unpaired image enhancement featuring reinforcement-learning-controlled image editing software, 2019. 7

[15] Eero Kurimo, Leena Lepistö, Jarno Nikkanen, Juuso Grén, Iivari Kunttu, and Jorma Laaksonen. The effect of motion blur and signal noise on image quality in low light imaging. In *Scandinavian Conference on Image Analysis*, pages 81–90. Springer, 2009. 1

[16] Edwin H Land. The retinex theory of color vision. *Scientific american*, 237(6):108–129, 1977. 2

[17] Chongyi Li, Chunle Guo, Linghao Han, Jun Jiang, Ming-Ming Cheng, Jinwei Gu, and Chen Change Loy. Low-light image and video enhancement using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):9396–9416, 2021. 2

[18] Orly Liba, Kiran Murthy, Yun-Ta Tsai, Tim Brooks, Tianfan Xue, Nikhil Karnad, Qiurui He, Jonathan T Barron, Dillon Sharlet, Ryan Geiss, et al. Handheld mobile photography in very low light. *ACM Trans. Graph.*, 38(6):164–1, 2019. 1

[19] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10561–10570, 2021. 7

[20] K. G. Lore, A. Akintayo, and S. Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement. *Pattern Recognition*, 61:650–662, 2017. 2

[21] F. Lv, Y. Li, and F. Lu. Attention guided low-light image enhancement with a large scale low-light simulation dataset. *International Journal of Computer Vision*, 129(7): 2175–2193, 2021. 2

[22] Sean Moran, Pierre Marza, Steven McDonagh, Sarah Parisot, and Gregory Slabaugh. Deeplpf: Deep local parametric filters for image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7

[23] Eli Schwartz, Raja Giryes, and Alex M Bronstein. Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing*, 28(2):912–923, 2018. 1

[24] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7

[25] Wencheng Wang, Xiaojin Wu, Xiaohui Yuan, and Zairui Gao. An experiment-based review of low-light image enhancement methods. *Ieee Access*, 8:87884–87917, 2020. 1

[26] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration, 2021. 7

[27] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement, 2018. 2, 7

[28] Ke Xu, Xin Yang, Baocai Yin, and Rynson WH Lau. Learning to restore low-light images via decomposition-and-enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2281–2290, 2020. 7

[29] Xiaogang Xu, Ruixing Wang, Chi-Wing Fu, and Jiaya Jia. Snr-aware low-light image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17714–17724, 2022. 7

[30] Wenhan Yang, Shiqi Wang, Yuming Fang, Yue Wang, and Jiaying Liu. Band representation-based semi-supervised low-light image enhancement: Bridging the gap between signal fidelity and perceptual quality. *IEEE Transactions on Image Processing*, 30:3461–3473, 2021. 7

[31] Wenhan Yang, Wenjing Wang, Haofeng Huang, Shiqi Wang, and Jiaying Liu. Sparse gradient regularized deep retinex network for robust low-light image enhancement. *IEEE Transactions on Image Processing*, 30:2072–2086, 2021. 7

[32] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M. H. Yang, and L. Shao. Learning enriched features for real image restoration and enhancement. In *European Conference on Computer Vision (ECCV)*, pages 492–511, 2020. 7

[33] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M. H. Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5728–5739, 2022. 7

[34] Hui Zeng, Jianrui Cai, Lida Li, Zisheng Cao, and Lei Zhang. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. 2020. 7

[35] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1632–1640, 2019. 2, 7