Efficient Real-Time Raw-to-Raw Denoising for Extreme Low-Light Ultra HD Video on Mobile Devices

Charantej Reddy Pochimireddy Subhasmita Sahoo Apoorva Verma Palavalli Shyam
Swapnil Malviya Sarvesh Raj Narayana Gadde
Samsung R&D Institute, Bangalore
{charan.tej, s.subhasmita, apoorva.v, p.shyam, malviya.s, sarvesh.s, raj.gadde}@samsung.com

Abstract

Recent advancements in deep neural networks (DNNs) have significantly improved visual quality of camera captures under low-light (<10lx) conditions. Yet, visual quality in extreme low-light (<1lx) remains inadequate. Existing DNN models are computationally intensive and suffer from large processing times, making them impractical for real-time enhancement of high-resolution videos. Consequently, Ultra HD (UHD) videos (4K/8K) captured in extreme low-light environments exhibit elevated noise and diminished details. Developing DNN-based solutions for UHD video enhancement faces challenges including paired dataset creation, temporal consistency, and efficient deployment under strict latency (<33ms) and power constraints (<250mA for 30fps video). We present a comprehensive methodology for developing a real-time raw-to-raw denoising solution for UHD videos in extreme low-light, designed for seamless integration into existing Image Signal Processor (ISP) pipelines. Unlike ISP-replacement approaches, our solution enhances commercial camera stacks across sensor platforms. Our framework comprises: (1) diverse dataset creation methodology, (2) a low-complexity model architecture optimized for mobile compute elements, and (3) efficient training and post-training optimizations (reparameterization, restructuring, quantization) to meet latency constraints while ensuring high-quality outputs. The result is a power-efficient real-time raw-to-raw video denoiser that improves extreme low-light video quality while preserving downstream ISP behavior.

1. Introduction

Smartphone cameras dominate modern video content creation, and users increasingly expect high-quality Ultra HD (4K/8K) videos across a wide range of lighting conditions. Meeting this expectation on mobile devices is challenging; each frame must be processed under strict latency and power[1] constraints, typically within <33 ms and <250 mA for 30 fps video. Short exposure times at high frame rates amplifies sensor noise and suppresses fine details, thus degrading visual quality. Under these limitations, traditional ISPs often fail to adequately process UHD streams, leading to noisy, low-detail videos with poor color reproduction and temporal flicker. These issues are further exacerbated in extreme low-light environments (<1lx), where sensor readout noise increases while visual quality diminishes even further (Figure 2). In such conditions, noise reduction techniques in conventional ISP pipelines are insufficient, making advanced AI-based ISP techniques critical for acceptable visual quality.

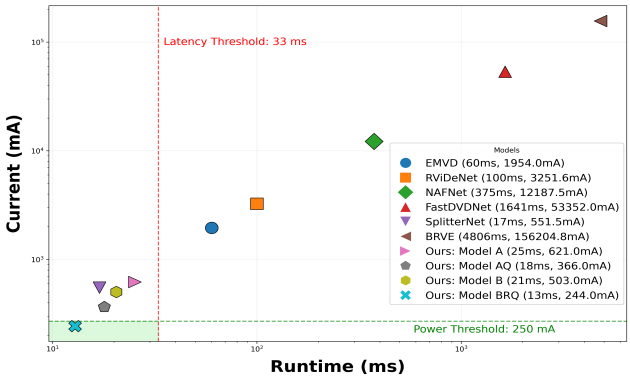


Figure 1. Latency and power efficiency comparison of raw to raw denoising networks (for measurement details refer section 3.4).

Compared to the sRGB domain, noise statistics in the raw sensor domain are far more tractable, and the raw measurements retain the fullest fidelity to the scene since they precede all ISP operations [38]. However, raw-domain video denoising under extreme low-light remains relatively underexplored [38, 41], primarily due to the difficulty in

[1]Power consumption during model execution on mobile device is proportional to current drawn (fixed voltage); thus, current (mA) is reported. Project page: link

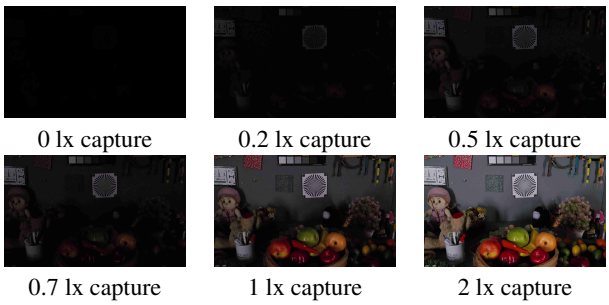


Figure 2. **Default ISP output under 0–2lx:** Degraded scene details and elevated noise with reducing illumination

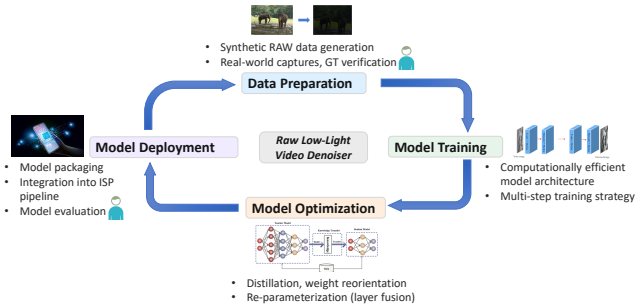


Figure 3. **End-to-End framework for real-time raw denoising in extreme low-light**: Integrates an ISP-compatible denoiser, hybrid real/synthetic data strategy, mobile-optimized architecture, and deployment optimizations for reduced latency/power. The loop depicts the *iterative refinement cycle* post-deployment (collect failure cases → dataset updates → fine-tune/re-train).

collecting paired datasets. Also, existing raw video denoising methods, including[38, 39] and top entries in recent challenges [36], often rely on complex architectures that are effective in terms of restoration quality, but impractical for real-time mobile deployment (Figure 1).

In this work, we address these gaps and target real-time raw-to-raw denoising for UHD videos in extreme low-light on mobile devices, with a design that is explicitly compatible with existing ISP pipelines. Our solution operates post sensor readout and pre-demosaic, preserving the raw CFA pattern so that downstream ISP behavior is minimally perturbed. We identify three key challenges:

- **Data scarcity:** Capturing paired low-light raw videos with realistic motion is challenging [27]. Synthetic methods [2, 34] often lack generalization.
- **Lack of low-complexity model:** Mobile-friendly architectures for real-time UHD video at 30/60 fps are scarce. Existing solutions [13, 33] are optimized for images (8MP in 0.5–1s), but not for real-time video processing.
- **Deployment constraints:** Meeting latency (<33ms) and power (<250mA) targets requires training low-complexity models and post-training optimizations (reparameterization, quantization), complicated by limited learnability and detail preservation trade-offs. State-of-the-art denoisers [9, 16, 22] are computationally intensive or ISP-integrated, hindering modular deployment.

We present a *comprehensive methodology* for developing real-time raw-to-raw denoising solution for UHD videos in extreme low-light, designed for seamless integration into existing ISP pipelines by operating on post-sensor readout. Our framework integrates: 1) dataset creation strategy: combines real low-light raw captures and synthetic sequences via unprocessing [2] and noise calibration [15, 26, 34]. 2) low-complexity video processing model: an optimal, scalable, modular model structure for UHD video processing. 3) deployment optimizations: post training optimizations such as distillation, structural re-parameterization [10], efficient model restructuring, and quantization [31, 37], aligned with mobile design principles [17, 18] to meet power and latency constraints. This paper is the first to holistically address mobile on-device deployment, extending beyond denoising model creation in state-of-the-art (SOTA) literature. Our contributions include:

1. **End-to-end framework:** From dataset curation to mobile deployment (Figure 3).
2. **Existing hardware ISP-compatibility:** Plug and play denoiser preserving the raw CFA pattern for downstream ISP compatibility.
3. **Mobile-friendly architecture:** Modular scalable architecture to process UHD videos.
4. **Deployment path:** Latency and power reduction via model restructuring along with post training optimizations such as re-parameterization, and quantization.

Paper organization. Section 2 reviews related work. Section 3 details the data preparation strategy and synthetic pipeline, base model architecture design, training strategy, and deployment optimizations. Section 4 reports results and ablations, and Section 5 concludes the paper.

2. Related Work

Raw image denoising has emerged as a key strategy for preserving sensor signal statistics and mitigating Color Filter Array (CFA) artifacts, driven by high-quality benchmarks such as [1, 3]. In the context of extreme low-light raw video denoising for mobile UHD capture, we review three relevant threads:

2.1. Raw denoising dataset

For videos, RViDeNet [38] introduced the CRVD dataset (55 sequences at ISO 1600–25600), simulating noise via ISO adjustments but lacking real-world low-light fidelity. The AIM 2025 challenge [36] later proposed a low-light raw video dataset (unreleased). On the other hand, synthetic raw generation bridges data gaps: unprocessing [2] inverts sRGB to raw, while physics-based ELD [34] and

Poisson-Gaussian models [15] calibrate noise. Normalizing flows [26] enable camera-specific noise learning, aiding cross-sensor generalization.

2.2. Raw denoising models and efficiency gaps

Standard RGB datasets (SID [3], SIDD [1], and MIDD [12]) progressed video enhancement but prioritize temporal consistency over efficiency. Models like V-BM4D [24], VNLnet [8], VideNN [6], FastDVDnet [30], and NAFNet [4] are computationally prohibitive for UHD videos. CRVD spurred innovations like recurrent fusion [25] and transformers [39], but these prioritize quality over deployability. Binary neural network-based architectures like BRVE [41] require specialized hardware, ignoring mobile NPU/GPU constraints.

2.3. Integrated training strategy and deployment

Training Challenges: Preserving CFA patterns while avoiding artifacts is critical. Top methods [7] use rotation invariance loss to minimize edge artifacts. Temporal consistency is enforced via deformable alignment [32], warping losses [40], or multi-step denoising [30].

Deployment Optimizations: Placement of denoising modules in ISP pipeline affects efficacy; pre-demosaic processing reduces color zippering [28]. While Joint Demosaic and Denoising (JDD) methods [9, 16, 22] improve accuracy, they compromise modularity.For mobile deployment, structural re-parameterization [10] fuses multi-branch blocks into 3×3 convolutions, Ghost modules [17] expand features efficiently, and quantization [31, 37] optimizes for target devices. Skip connection shortening [35] further refines resource efficiency.

3. Proposed Method

We propose a comprehensive end-to-end methodology for real-time extreme low-light raw video denoising, encompassing dataset creation strategy, model development, and mobile deployment. The framework integrates three pivotal components:

- **Dataset creation strategy**: Our dataset creation strategy employs a hybrid approach, combining synthetic pairs and real captures tailored for extreme low-light conditions. Synthetic data is generated via a custom degradation pipeline inspired by unprocessing [2], enhanced with color blob cut-mix to simulate intensity gradient around small light sources and mitigate color bleed artifacts. Real captures provide domain adaptation, while realistic motion synthesis [29] applied to static frames ensures temporal coherence, bridging the gap between synthetic and real-world dynamics.
- **Architectural design**: We design a computationally efficient feature refinement architecture inspired by [19, 21]. We replace the attention modules proposed in [19] with a re-parameterizable structure. We also introduce multi-stage skip connections within the re-parametrizable structure to ensure effective information flow. This architecture supports single and multi-frame configurations to balance receptive field coverage and efficiency.
- **Training and deployment**: We use a multi-term loss to enforce raw fidelity (L1 loss) and color consistency (chromatic aberration penalty). We also perform post-training optimizations like structural re-parameterization, distillation, model restructuring, and quantization for minimal power consumption.

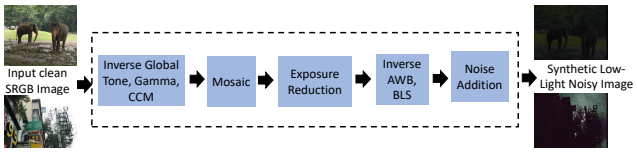


Figure 4. **Synthetic data generation pipeline.** Simulates $<$1lx low-light conditions via exposure reduction [27], converting sRGB to pseudo-raw through: 1) Inverse tone mapping, gamma, CCM, 2) Bayer mosaicing, 3) Exposure reduction, 4) Inverse AWB, BLS, and 5) Noise addition for realistic sensor noise synthesis.

3.1. Dataset preparation

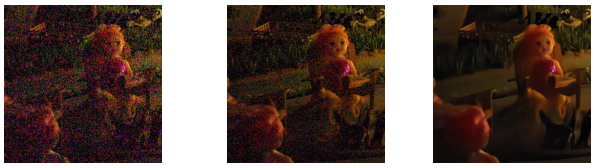
Acquiring aligned noisy-clean low-light raw video pairs is challenging due to scene variability constraints (e.g., motion) and limited access to raw sensor data. Prior work [27] identifies two core issues: (1) *data capture difficulty:* large-scale low-light acquisition is inherently constrained, and (2) *ground truth preparation:* motion complicates clean reference generation. To address these, we create three sets of data:

- **Synthetic data**: Diverse pseudo-raw scenes synthesized from high-quality sRGB images using unprocessing, exposure reduction, and noise augmentation.
- **Tripod-captured real data**: Static low-light raw videos captured on mobile camera, augmented with synthetic motion to simulate dynamic scenarios.
- **Benchmarking set**: 10-video evaluation suite (5 static tripod-mounted, 5 synthetic-motion induced on static) to assess model performance under static and dynamic conditions.

This strategy ensures diversity, reduces domain mismatch, and enables practical deployment. Synthetic data is used to train the initial model, while real-world captures (static + motion-augmented) are used to fine-tune it.

3.1.1. Synthetic dataset Preparation:

We generate synthetic data using sRGB images instead of video frames due to the former's superior texture and luminance fidelity. We employ two complementary sets for balanced representation: **Set 1** (texture emphasis): captured 1500 sRGB images in well-lit indoor and outdoor conditions to preserve high-frequency details and natural pat-



Noisy frame (cropped) Burst average (cropped) final GT (cropped)

Figure 5. **Ground truth preparation.** Controlled low-light raw capture with tripod stabilization, followed by two-stage denoising: burst averaging + residual noise removal via synthetic-trained large mRLFB model.

terns, augmented with color blob cut-mix to simulate intensity gradient around small light sources and mitigate color bleed artifacts; **Set 2** (luminance range emphasis): includes 1200 sRGB images with strong backlighting to capture silhouette scenes and ensure wide luminance coverage.

Low-light conditions ($<$1lx) are simulated via exposure reduction [27] to produce darkened images (I_{dark}). The pipeline (Figure 4) converts clean sRGB to clean pseudo-raw (B_{GT}): 1) Inverse tone, gamma ($\gamma = 2.12$), CCM, 2) Bayer mosaicing, 3) Exposure reduction, and 4) Inverse AWB, BLS. ISP model parameters were extracted from the metadata stored during RGB image captures on a Samsung Galaxy S25. Across our captures, *WB* gains have mean = [1.81, 1, 1.89] with per channel variance = [0.01, 0, 0.03] for [R,G,B]. The CCM has element-wise mean = [1.54, -0.49, -0.05;-0.34, 1.49, -0.15;0.07, -0.94, 1.87] with element-wise variance = [0.006, 0.005, 0.007;0.0006, 0.008, 0.009;0.0009, 0.006, 0.009], where columns correspond to (R,G,B).

Noise augmentation: We use heteroscedastic Gaussian noise [14, 23] model to mimic noise under low-light:

$$B_{\text{in}} = \mathcal{N}(0, \beta_1 B_{GT} + \beta_2), \tag{1}$$

where β_1, β_2 parameterize shot/read noise. Parameters are calibrated using 0 lx (dark) and $<$ 1lx captures to estimate noise statistics, followed by a grid search with perceptual/human validation.

3.1.2. Real world data preparation

Static scenes for training: Low-light raw sequences were captured on mobile camera in a controlled lab environment with tripod stabilization. To ensure stability, the first 90 frames were excluded. Under extreme low-light ($<$1lx), burst averaging alone is insufficient for noise removal (Figure 5). Thus, clean Ground Truth (GT) was generated via a two-stage process:

1. **Burst averaging:** 90 consecutive frames averaged to remove additive zero-mean noise.
2. **Residual denoising:** A synthetic-trained 16 mRLFB model (Section 3.2) eliminated residual noise. Resulting denoised GT-noisy pairs fine-tuned the model for real-world noise adaptation.

Using a model to generate ground truth (GT) offers a practical compromise for supervision in ill-posed problems. However, this approach introduces model-specific constraints, such as inherent bias and a performance ceiling on GT quality, yet remains sufficiently effective for practical applications.

A total of 160 videos were captured, with all GTs manually verified for quality. However, 60 videos were excluded due to blur caused by local motion or lighting changes.

Synthetic motion incorporation: Ground-truth for real motion is not possible under $<$1lx due to trails/artifacts after averaging; hence we use controlled tripod captures and synthetic motion for temporal evaluation. Static noisy-GT pairs were augmented with synthetic motion from a predefined dictionary, simulating camera motion scenarios. This ensured frame-wise alignment and addressed GT generation challenges for dynamic videos.

3.2. Model architecture Design

We propose a computationally efficient raw video denoising model, partly inspired by the Residual Local Feature Network (RLFN) [19, 21]. We make two key modifications: attention modules are replaced with re-parameterizable structures and optimally placed skip connections. The architecture (Figure 6) is designed for deployment on resource-constrained mobile devices.

For UHD video processing, feature resolution critically impacts computation time. We employ a $k \times k$ Space-to-Depth (S2D) operation, to downsample input resolution by k while expanding channels to k^2, preserving color channel relationships. Subsequent convolutions operate on this reduced resolution, balancing computational efficiency and output quality.

Mobile-Optimized Feature Processing: The model begins with a 3×3 convolution layer to extract shallow features. The core processing component uses a cascade of $N = 4$ mobile-optimized Residual Local Feature Blocks (mRLFBs), modified from RLFN by removing the attention module, thus eliminating global pooling/upsampling overhead. Each mRLFB comprises: three sequential 3×3 convolutional layers with ReLU activations, followed by a 1×1 convolution, wrapped in a residual connection:

$$F_{\text{out}} = F_{\text{in}} + W_1 * (F_{\text{in}} + \sigma(W_3 * \sigma(W_2 * \sigma(W_1 * F_{\text{in}})))), \tag{2}$$

where W_i denote convolutional kernels, and $\sigma(\cdot)$ is ReLU. This structure enables feature refinement similar to RLFN while avoiding the computational complexity of distillation-based modules (RFDN). During training, a reparameterizable block replaces the 3×3 convolutions (Section 3.4).

Feature fusion and output: Deep features from mRLFBs are aggregated via 3×3 convolution and concatenated with shallow features through a 1×1 convolution, preserving

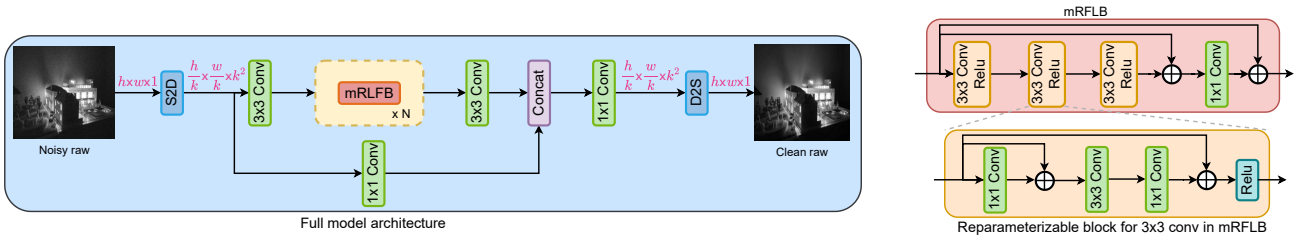


Figure 6. **Denoising base model architecture.** Integrates Space-to-Depth (S2D) for spatial reduction, mobile-optimized Residual Local Feature Blocks (mRLFB), and residual learning to enable real-time UHD video processing on resource-constrained devices.

critical high-frequency details like texture. The fused features undergo a final 1×1 convolution and Depth-to-Space (D2S) to reconstruct denoised output.

3.3. Model training strategy

3.3.1. Training Objective

We adopt a composite loss to enforce raw fidelity and inter-channel consistency while mitigating color artifacts:

Raw Reconstruction Loss: Primary supervision via L1 loss between predicted (B_{pred}) and ground-truth (B_{GT}) raw frames:

$$L_{\text{raw}} = \frac{1}{M} \sum \left| B_{\text{GT}} - B_{\text{pred}} \right|. \tag{3}$$

where M is the batch size, and the loss is averaged over all samples in the batch. This loss is optimized iteratively across the dataset during training.

Chromatic Aberration Loss: Penalizes cross-channel misalignments by comparing differences in green-averaged channels. Given the predicted Bayer channels $\hat{R}, \hat{G}_1, \hat{G}_2, \hat{B}$ in B_{pred} and ground-truth channels R, G_1, G_2, B in B_{GT}, we define the green reference as the average: $G_{\text{avg}} = \frac{1}{2}(G_1 + G_2)$, and $\hat{G}_{\text{avg}} = \frac{1}{2}(\hat{G}_1 + \hat{G}_2)$. We then compute the difference features: $D_{BG} = (B - G_{\text{avg}})$, and $D_{RG} = (R - G_{\text{avg}})$, with corresponding predictions $\hat{D}_{BG}, \hat{D}_{RG}$. The chromatic loss is computed as

$$L_{\text{chromatic}} = \frac{1}{M} \sum \left(\|D_{BG} - \hat{D}_{BG}\|_1 + \|D_{RG} - \hat{D}_{RG}\|_1 \right). \tag{4}$$

Final Objective: Combined loss with tunable weights:

$$L = w_r L_{\text{raw}} + w_c L_{\text{chromatic}}, \tag{5}$$

where $w_r = 0.6$ and $w_c = 0.4$ are hyperparameters balancing raw fidelity and color consistency.

3.3.2. Multi-frame processing

The single-frame architecture (Figure 6) lacks inherent temporal consistency, which is critical for perceived video quality. SOTA models often rely on compute-intensive complex temporal blocks for consistency. However, they introduce prohibitive latency and power overhead, making them impractical for mobile deployment. To address this issue, we modify the initial 3×3 convolution layer to accept Space-to-Depth (S2D) outputs from both previous and current frames (32-channel input) and generate denoised output for the current frame while preserving the remaining architecture. Trained with the same loss function, this two-frame approach balances temporal coherence and latency, enhancing consistency without compromising on per-frame denoising performance or exceeding mobile constraints.

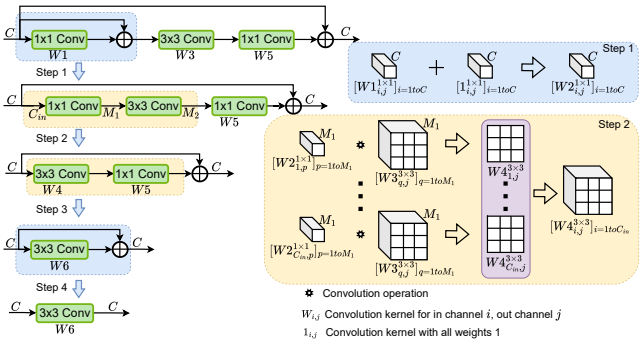


Figure 7. **Structural re-parameterization.** A multi-branch block is used in place of 3×3 convolution in mRLFB (Figure 6) for training which are fused together post-training.

3.4. Model optimization and deployment

Step 1: Knowledge distillation. A high-capacity *teacher* model $\mathcal{T}$ (Model A, Section 4) with N = 4, k = 4, and intermediate feature depth d = 32, is trained and distilled into a compact *student* model $\mathcal{S}$ (Model B, Section 4) N = 4, k = 4, and d = 16. Distillation uses output-space and intermediate-feature guidance loss to transfer knowledge while maintaining efficiency.

$$L_{KD} = \|\mathcal{T}(y) - \mathcal{S}(y)\|_1 \; + \; \beta \, \|\phi(\mathcal{T}) - \phi(\mathcal{S})\|_2^2, \tag{6}$$

where $\phi(\cdot)$ extracts shallow/mid-level features. The student is co-trained with L (base loss) and L_{KD}.

Step 2: Structural re-parameterization (layer fusion). While skip connections enhance information flow and gradient propagation in ConvNets, they incur significant memory overhead by retaining feature maps across layers—a critical limitation for real-time mobile deployment. To address this, we adopt re-parameterization inspired by [42].

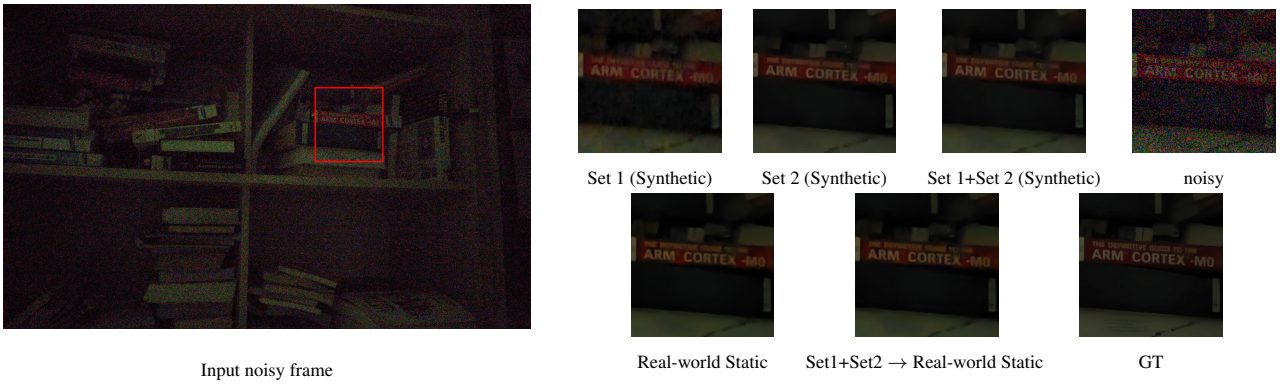


Figure 8. **Impact of Synthetic Data.** Visual comparison of base model outputs trained on Synthetic (Set1,Set2, combined), Real-World Static, and Hybrid (pre-trained on Synthetic + fine-tuned on Real-World Static)

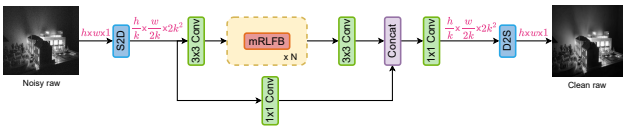


Figure 9. **Spatial resolution reduction.** Restructured model with halved spatial resolution and doubled channel depth via weight reconfiguration, leveraging NPU latency equivalence for both.

During training, a multi-branch block is employed: a 1×1 convolution with a skip for local feature distillation, followed by a 3×3 convolution and subsequent 1×1 convolution, followed by a global skip connection. Post-training, this block is transformed into a single 3×3 convolution via kernel fusion as shown in Figure 7. This preserves mathematical equivalence and representational capacity while eliminating skip-induced memory overhead, reducing runtime compared to the multi-branch design.

Step 3: Spatial Resolution Reduction (restructuring). To further optimize the model, we reduce the spatial resolution of feature maps by half during inference. We reshape the model weights and modify the S2D operation to halve the spatial dimension along width, as shown in Figure 9. This adjustment doubles the intermediate channel depth through weight reconfiguration. We avoid training in this configuration to prevent directional bias, as the receptive field along one dimension becomes larger. Conversely, reducing the dimension along both directions (height and width) limits the channel depth to a maximum of 64, beyond which NPU parallelization is hindered, resulting in increased latency.

Step 4: Quantization. Integer operations are inherently less power-intensive than their floating-point counterparts. However, quantizing floating-point model weights to integers often results in quality degradation due to precision loss. We increase quantization granularity by employing post-training int16 quantization with per-channel symmetric weights and per-tensor activations. Calibration uses a few hundred frames spanning the ISO/motion distribution. If the quality drop exceeds a small threshold, we perform quantization-aware fine-tuning for a few epochs with fake quantization. The final I/O remains 10/12-bit integer Bayer, with internal int16 accumulators promoted to int32 wherever required.

Step 5: Smartphone deployment & measurement protocol. The model is deployed within the ISP pipeline, operating post-sensor readout and optimized for 4K@30fps on the Qualcomm® Hexagon NPU. For real-time models, such as our approach and SplitterNet, runtime is measured within the ISP pipeline post-deployment. For non-real-time models like NAFNet and BRVE, runtime is assessed via standalone executables as real time processing in ISP is not feasible due to large runtimes. Power consumption is evaluated through differential measurements between the default ISP and the model-integrated pipelines using the Monsoon HV for real-time scenarios. For non-real-time cases, power is linearly scaled based on runtime, as direct measurement within the ISP is not feasible. Testing is conducted on Samsung Galaxy S25 Plus (Snapdragon® 8 Gen 3 Elite) with a 5V supply at room temperature (25°C). Precision levels are set to INT16 for quantized models and FP16 for others.

4. Experiments

In this section, we evaluate the proposed raw to raw low-light video denoiser across (i) restoration quality, (ii) temporal stability, and (iii) on-device efficiency, following the settings described in Section 3. We report metrics in the raw domain (linear intensity) and, show sRGB frames by running a fixed ISP post-process and gain application for qualitative perceptual comparison.

4.1. Experiment Settings

Datasets: We use the *sensor-aware synthetic* raw videos and *real mobile captures* (Section 3) to evaluate the model's performance under both static and motion conditions. We

also submit the mobile captured videos of the model deployed in camera pipeline in supplementary material.

Training details: We use a batch size of 16 and packed-RAW patches of 256×256. We optimize with the loss mentioned in Section 3 using the Adam optimizer and a cosine-annealing schedule with initial learning rate 1×10^{-4} in TensorFlow using one NVIDIA A100 GPU.

Proposed models: We evaluate four variants of the single-frame and multi-frame architecture (Figure 6):

- **Model A:** $N = 4$, $k = 4$, feature depth $d = 32$.
- **Model AQ:** Quantized version of Model A for mobile deployment.
- **Model B:** Distilled from Model A ($N = 4$, $k = 4$, $d = 16$).
- **Model BRQ:** Model B with spatial resolution reduction and quantization for mobile deployment.

We denote the single frame architecture with * (for example Model A*). Ablation studies (Table 5) compare visual quality, runtime, and current consumption across these variants.

Metrics: We report frame-averaged Peak Signal-to-Noise Ratio (PSNR) in dB and Structural Similarity Index Measure (SSIM) on linear raw (12 bit normalized to [0,1]) full frame (no ROI) to assess restoration fidelity; for device evaluation we report per-frame inference time (ms) and current drawn (mA). Higher PSNR/SSIM and lower latency/current are better. We also report temporal stability metrics- tOF [5], tLP [5], Avg. flicker [11], and video quality metric VMAF [20].

4.2. Impact of the Synthetic Dataset

We demonstrate the effectiveness of different components of the training dataset (synthetic + real-world static) through the following ablation. We create models trained on Set1 only, Set2 only, Set1+Set2 (synthetic), and real-world static. We also take a model pretrained on synthetic data and fine-tune it on real-world static data. Key findings from evaluating these models (Table 1) are: **1) Synthetic Data Synergy**: Combining Set1 and Set2 yields higher PSNR than individual sets, underscoring the need for diverse synthetic data covering texture and luminance variations. **2) Domain Alignment**: Training solely on Real-World static data improves PSNR, highlighting the importance of domain alignment for practical performance. **3) Fine-tuning Advantage**: The model pre-trained on Set1+Set2 and fine-tuned on Real-World static achieves the highest PSNR, demonstrating the efficacy of synthetic-to-real transfer learning. Visual comparisons in Fig. 8 further validate these trends.

4.3. Comparison with State of the Art

While SOTA models often face deployment constraints due to computational and power limitations, we benchmark our method against three low-complex models: BRVE [41] (trained with authors' setup for raw-to-raw enhancement) and frame-wise NAFNet [4] (single frame raw processing), and SplitterNet [13]. We assess all models under two training paradigms: 1) trained solely on real-world data, and 2) pre-trained on synthetic data then fine-tuned on real-world data. As shown in Table 2, our approach achieves superior runtime and power with on par PSNR in both settings. Visual comparisons in Figure 10 further corroborate these quantitative gains.

[2]Model B* is distilled from the optimized version of Model A* (trained on synthetic and fine-tuned on real-world data). Consequently, real-world static evaluation results for Model B are marked as 'empty' in the table.

Table 1. **Impact of Synthetic Data.** PSNR and SSIM comparison of models trained on synthetic (Set1 only, Set2 only, Set1+Set2), and real datasets (Real-World static, Set1+Set2 pre-training + fine-tuning on Real-World static.

Training Regime	PSNR↑	SSIM↑
Set1 (Synthetic)	56.97	0.9885
Set2 (Synthetic)	56.53	0.9879
Set1 + Set2 (Synthetic)	58.05	0.9885
Real-World Static	60.18	0.9883
Set1+Set2 → Real-World Static	**60.54**	**0.9886**

Table 2. **Comparison with SOTA models BRVE [41], NAFNet [4], and SplitterNet [13].** Evaluated under two training regimes: 1) real-world, 2) synthetic pre-training + real-world fine-tuning[2].

Method	Runtime (ms)↓	Current (mA)↓	Real-World Static		Synthetic → Real	
			PSNR↑	SSIM↑	PSNR↑	SSIM↑
NAFNet	375	12187	59.09	0.9838	61.49	0.9857
BRVE	4806.3	156204	59.55	0.9993	60.42	0.9994
SplitterNet	16.97	551	53.83	0.9735	55.74	0.9860
Ours: Model A*	22.95	565	60.18	0.9883	60.54	0.9886
Ours: Model B*	19.30	475	-	-	58.63	0.9880

4.3.1. Evaluation on Public Datasets

Due to the absence of extreme low-light raw video datasets, we evaluate generalization using the closest available moderately low-light datasets. Our robustness analysis employs CRVD and SRVD validation sets [38], adhering to their methodology: indoor scenes 7–11 from CRVD and scenes [2,9,10,11] from SRVD. As shown in Table 3, our approach outperforms baselines across both datasets, demonstrating robust generalization.

In extreme low light (<1lx) the pixel intensities are very small in *RAW*; thus absolute MSE is small in RAW domain, yielding high PSNR (55-60 DB) in Table 2. In higher lx datasets (SIDD/ELD/CRVD/SRVD), the pixel intensities are higher hence the reported PSNR is 30-45 DB. The baseline PSNR (between GT and the Noisy) for the proposed bench mark are (46.45 DB) and for the CRVD/SRVD are (27.01/27.36 DB).

4.4. Temporal Noise Reduction and Motion Compensation

While the single-frame model delivers good per-frame quality, it introduces temporal flicker in extreme low-light con-

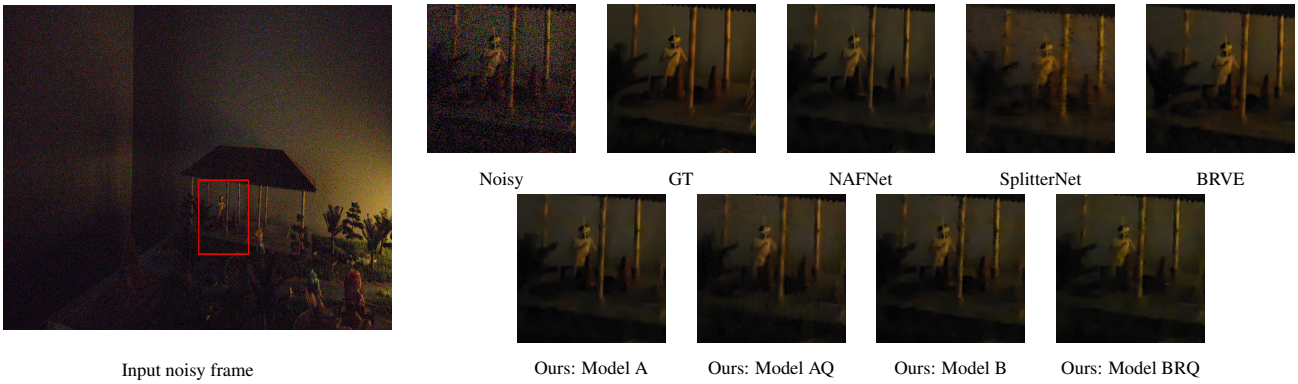


Figure 10. Visual comparison of different low-light video denoising methods on Real world captures

Table 3. **Performance evaluation results on the public dataset.** Evaluated on CRVD and SRVD from [38]

Method	CRVD		SRVD	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
NAFNet	33.47	0.9517	33.37	0.9500
BRVE	34.43	0.9601	33.42	0.9543
SplitterNet	35.33	0.9735	31.62	0.9572
Ours: Model A*	35.76	0.9707	34.74	0.9701
Ours: Model B*	37.21	0.9798	34.46	0.9730

ditions (<11x). To mitigate this issue, we adapted the model to process multi-frame inputs (two frames in, one frame out), enhancing temporal coherency. Multi-frame model is trained on synthetic data adapted for two-frame input, and fine-tuned on real world data. Comparisons of temporal performance between the single-frame and multi-frame models are presented in Tables 4. We refer the readers to videos provided in supplementary material.

Table 4. **Temporal evaluation.** Temporal stability metrics and VMAF scores on Benchmarking set (Section 3).

Metric	NAFNet	BRVE	SplitterNet	Model A*	Model A
tOF ↓ ×10	0.81	0.61	1.56	0.63	0.55
tLP ↓ ×100	1.70	1.26	3.99	1.83	0.70
Avg. flicker ↓ ×10000	1.20	2.46	2.90	1.26	0.78
VMAF ↑	67.11	66.87	40.74	49.06	70.43

4.4.1. On device optimization

Table 5 presents an ablation study on post-training optimizations (restructuring/spatial resolution reduction and quantization) for on-device deployment. Distillation from Model A to Model B reduces runtime and power by 20%, while restructuring further cuts Model B's run time by 37% and power consumption by 16% leveraging NPU parallelization. Conversely, restructuring Model A increases intermediate channel depth $d$from 32 to 64 beyond NPU efficient regime (32) resulting in higher runtime and power. The table quantifies trade-offs in visual quality, latency, and power across optimized variants.

Table 5. **Ablations on proposed Multi-frame model optimizations.** Runtime, current/power and IQ comparison.

Method	Runtime (ms) ↓	Current (mA) ↓	PSNR ↑	SSIM ↑
Model A	25.32	621	60.54	0.9886
Model A (restructured)	43.54	1405	60.54	0.9886
Model AQ	17.94	366	57.46	0.9992
Model B	20.53	503	58.63	0.9880
Model B (restructured)	12.96	419	58.63	0.9880
Model BRQ	12.91	244	57.17	0.9986

5. Conclusion

We presented an end-to-end comprehensive methodology for developing and deploying a real-time raw UHD (4K/8K) video denoiser compatible with commercial ISP pipelines. Our solution operates pre-demosaic, enhancing existing systems without requiring full ISP replacement. This paper is the first to holistically address mobile on-device deployment, extending beyond denoising model creation prevalent in SOTA literature.

Key contributions include hybrid dataset creation strategy merging sensor-aware synthetic and real mobile captures; a mobile-optimized architecture with reparameterizable blocks for compute efficiency; and deployment optimizations (distillation, spatial restructuring, quantization) enabling real-time execution on mobile device.

Experiments demonstrate real-time performance on mobile devices with high visual quality while preserving downstream ISP behavior. Ablation studies validate synthetic-real data synergy and model efficacy. The raw-in/raw-out design ensures plug-and-play integration into camera pipelines, bridging research and practical deployment to advance extreme low-light video quality on mobile devices.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1692–1700, 2018. 2, 3

[2] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11036–11045, 2019. 2, 3

[3] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3291–3300, 2018. 2, 3

[4] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European conference on computer vision*, pages 17–33. Springer, 2022. 3, 7

[5] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transactions on Graphics (TOG)*, 39(4):75–1, 2020. 7

[6] Michele Claus and Jan Van Gemert. Videnn: Deep blind video denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 3

[7] Marcos Conde, Radu Timofte, Zihao Lu, Xiangyu Kong, Xiaoxia Xing, Fan Wang, Suejin Han, MinKyu Park, Tianyu Hao, Yuhong He, et al. Ntire 2025 challenge on raw image restoration and super-resolution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1148–1171, 2025. 3

[8] Axel Davy, Thibaud Ehret, Jean-Michel Morel, Pablo Arias, and Gabriele Facciolo. A non-local cnn for video denoising. In *2019 IEEE international conference on image processing (ICIP)*, pages 2409–2413. IEEE, 2019. 3

[9] Valéry Dewil, Adrien Courtois, Mariano Rodríguez, Thibaud Ehret, Nicola Brandonisio, Denis Bujoreanu, Gabriele Facciolo, and Pablo Arias. Video joint denoising and demosaicing with recurrent cnns. In *Proceedings of the ieee/cvf winter conference on applications of computer vision*, pages 5108–5119, 2023. 2, 3

[10] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742, 2021. 2, 3

[11] Pontus Ebelin, Gyorgy Denes, Tomas Akenine-Möller, Kalle Åström, Magnus Oskarsson, and William H McIlhagga. Estimates of temporal edge detection filters in human vision. *ACM Transactions on Applied Perception*, 21(2):1–25, 2024. 7

[12] Roman Flepp, Andrey Ignatov, Radu Timofte, and Luc Van Gool. Real-world mobile image denoising dataset with efficient baselines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22368–22377, 2024. 3

[13] Roman Flepp, Andrey Ignatov, Radu Timofte, and Luc Van Gool. Real-world mobile image denoising dataset with efficient baselines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22368–22377, 2024. 2, 7

[14] Alessandro Foi. Clipped noisy images: Heteroskedastic modeling and practical denoising. *Signal Processing*, 89 (12):2609–2629, 2009. 4

[15] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE transactions on image processing*, 17(10):1737–1754, 2008. 2

[16] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. pages 1–12. ACM New York, NY, USA, 2016. 2, 3

[17] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1580–1589, 2020. 2, 3

[18] Andrey Ignatov, Kim Byeoung-Su, Radu Timofte, and Angeline Pouget. Fast camera image denoising on mobile gpus with deep learning, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2515–2524, 2021. 2

[19] Fangyuan Kong, Mingxi Li, Songwei Liu, Ding Liu, Jingwen He, Yang Bai, Fangmin Chen, and Lean Fu. Residual local feature network for efficient super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 766–776, 2022. 3, 4

[20] Zhi Li. On vmaf's property in the presence of image enhancement operations, 2021. 7

[21] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In *European conference on computer vision*, pages 41–55. Springer, 2020. 3, 4

[22] Lin Liu, Xu Jia, Jianzhuang Liu, and Qi Tian. Joint demosaicing and denoising with self guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2240–2249, 2020. 2, 3

[23] Xinhao Liu, Masayuki Tanaka, and Masatoshi Okutomi. Practical signal-dependent noise parameter estimation from a single noisy image. *IEEE Transactions on Image Processing*, 23(10):4361–4371, 2014. 4

[24] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian. Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. *IEEE Transactions on image processing*, 21(9): 3952–3966, 2012. 3

[25] Matteo Maggioni, Yibin Huang, Cheng Li, Shuai Xiao, Zhongqian Fu, and Fenglong Song. Efficient multi-stage video denoising with recurrent spatio-temporal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3466–3475, 2021. 3

[26] Ali Maleky, Shayan Kousha, Michael S Brown, and Marcus A Brubaker. Noise2noiseflow: Realistic camera noise

modeling without clean images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, pages 17632–17641, 2022. 2

[27] Abhijith Punnappurath, Abdullah Abuolaim, Abdelrahman Abdelhamed, Alex Levinshtein, and Michael S Brown. Day-to-night image synthesis for training nighttime neural isps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10769–10778, 2022. 2, 3, 4

[28] Marco Sánchez-Beeckman, Antoni Buades, Nicola Brandonisio, and Bilel Kanoun. Combining pre-and post-demosaicking noise removal for raw video. *IEEE Transactions on Image Processing*, 2025. 3

[29] Yong Shu, Liquan Shen, Xiangyu Hu, Mengyao Li, and Zihao Zhou. Towards real-world hdr video reconstruction: A large-scale benchmark dataset and a two-stage alignment network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2879–2888, 2024. 3

[30] Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1354–1363, 2020. 3

[31] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8612–8620, 2019. 2, 3

[32] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 3

[33] Yuzhi Wang, Haibin Huang, Qin Xu, Jiaming Liu, Yiqun Liu, and Jue Wang. Practical deep raw image denoising on mobile devices. In *European Conference on Computer Vision*, pages 1–16. Springer, 2020. 2

[34] Kaixuan Wei, Ying Fu, Jiaolong Yang, and Hua Huang. A physics-based noise formation model for extreme low-light raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2758–2767, 2020. 2

[35] Olivia Weng, Gabriel Marcano, Vladimir Loncar, Alireza Khodamoradi, Abarajithan G, Nojan Sheybani, Andres Meza, Farinaz Koushanfar, Kristof Denolf, Javier Mauricio Duarte, et al. Tailor: Altering skip connections for resource-efficient inference. *ACM Transactions on Reconfigurable Technology and Systems*, 17(1):1–23, 2024. 3

[36] Alexander Yakovenko, George Chakvetadze, Ilya Khrapov, Maksim Zhelezov, Dmitry Vatolin, Radu Timofte, Youngjin Oh, Junhyeong Kwon, Junyoung Park, Nam Ik Cho, et al. Aim 2025 low-light raw video denoising challenge: Dataset, methods and results. *arXiv preprint arXiv:2508.16830*, 2025. 2

[37] Tien-Ju Yang, Andrew Howard, Bo Chen, Xiao Zhang, Alec Go, Mark Sandler, Vivienne Sze, and Hartwig Adam. Netadapt: Platform-aware neural network adaptation for mobile applications. In *Proceedings of the European conference on computer vision (ECCV)*, pages 285–300, 2018. 2, 3

[38] Huanjing Yue, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu Yang. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2301–2310, 2020. 1, 2, 7, 8

[39] Huanjing Yue, Cong Cao, Lei Liao, and Jingyu Yang. Rvideformer: Efficient raw video denoising transformer with a larger benchmark dataset. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 2, 3

[40] Fan Zhang, Yu Li, Shaodi You, and Ying Fu. Learning temporal consistency for low light video enhancement from single images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4967–4976, 2021. 3

[41] Gengchen Zhang, Yulun Zhang, Xin Yuan, and Ying Fu. Binarized low-light raw video enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25753–25762, 2024. 1, 3, 7

[42] Yuantong Zhang, Baoxin Teng, Daiqin Yang, Zhenzhong Chen, Haichuan Ma, Gang Li, and Wenpeng Ding. Learning a single convolutional layer model for low light image enhancement. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7):5995–6008, 2023. 5