Adaptive Pruning for Universal-Filter Image Color Correction

Aleksei Samarin, Aleksei Toropov, Artem Nazarenko
ITMO University / ISP RAS
St. Petersburg, Russia
avsamarin@itmo.ru, toropov.ag@hotmail.com, aanazarenko@itmo.ru

Egor Kotenko
St. Petersburg State University / ISP RAS
St. Petersburg, Russia
kotenkoed@gmail.com

Anastasia Mamaeva, Dmitry Nazarenko
ITMO University
St. Petersburg, Russia
asmamaeva@itmo.ru, nazarenkodmit@gmail.com

Elena Mikhailova
St. Petersburg State University
St. Petersburg, Russia
e.mikhailova@itmo.ru

Valentin Malykh
International IT University
Almaty, Kazakhstan
valentin.malykh@phystech.edu

Alexander Savelev
St. Petersburg Electrotechnical University / ISP RAS
St. Petersburg, Russia
algsavelev@gmail.com

Alexander Motyko
St. Petersburg Electrotechnical University
St. Petersburg, Russia
aamotyko@etu.ru

***Abstract*—Universal-filter-based models provide an efficient and interpretable framework for image color correction by predicting parameters of structured correction operators. However, practical deployment still benefits from additional reductions in computational cost and memory footprint, especially for on-device and real-time pipelines where color stability is perceptually critical. In this work, we study pruning for a universal-filter color correction model and benchmark representative unstructured and structured pruning baselines, including channel pruning and latency-aware structured pruning. We further propose a configuration-agnostic customized sparsity allocation strategy that adapts pruning decisions across predictor components to better preserve quality under compression. Experiments on the MIT-Adobe FiveK benchmark show that unstructured pruning yields limited wall-clock gains despite parameter reduction, whereas structured strategies provide meaningful acceleration. At 50% sparsity, the proposed method reduces latency from 8.5 ms to 4.4 ms and FLOPs from 85.4M to 36.3M while maintaining high quality (PSNR = 24.06 dB, SSIM = 0.916). At 60% sparsity, it further reduces latency to 4.0 ms with stable performance (PSNR = 23.89 dB, SSIM = 0.910). Overall, the proposed strategy achieves a more favorable quality-efficiency trade-off than baseline pruning approaches, improving the deployability of universal-filter color correction models.**

***Index Terms*—image color correction, universal filters, model pruning, efficiency optimization, MIT-Adobe FiveK**

I. Introduction

Recent progress in learned image processing has significantly improved automatic photo adjustment, yet color correction remains a particularly sensitive and practically important subproblem. Unlike texture-oriented restoration, color correction must preserve global consistency while compensating for illumination changes, sensor characteristics, and scene-dependent color casts. Failures are often perceptually salient, manifesting as unstable white balance, hue shifts, and inconsistent tone rendering across images, which makes robust color correction essential for both consumer photography and downstream imaging pipelines. Learning-based approaches have addressed parts of this problem through ranking-based color enhancement [1], color constancy estimation [2], and white-balance editing [3], while more recent work explicitly targets color shift estimation and correction as a dedicated modeling objective [4]. Despite these advances, achieving stable color correction under diverse real-world capture conditions remains challenging, especially when models must generalize across devices and acquisition settings [2]–[4].

A parallel line of work has demonstrated that deep models can approximate complex image processing operators and enable efficient learned pipelines [5], [6]. However, deployment constraints remain central to color correction because the operation is frequently performed on-device and at scale, for example, as part of camera pipelines, batch post-processing, or real-time previews. In such settings, compute, memory footprint, and energy consumption become limiting factors, and even moderately sized networks can be impractical when executed repeatedly or combined with other learned modules [5], [6]. These constraints are further amplified in challenging illumination regimes, where models often need increased capacity to avoid artifacts and preserve color stability [7]–[10].

In this context, universal-filter-based color correction provides an attractive compromise between expressivity, interpretability, and efficiency. Instead of learning an unconstrained pixel-to-pixel mapping, a compact predictor network estimates parameters for a set of predefined or learnable filters, and the final corrected image is obtained through a structured composition of filter outputs. Such a design supports trans-

parency of the correction process and can reduce the number of trainable parameters relative to fully unconstrained end-to-end approaches, while remaining competitive on standard benchmarks [5]. We recently proposed a universal-filter model that achieves strong quality-speed trade-offs on MIT-Adobe FiveK while remaining lightweight [11], and later extended this direction with an unpaired learning mode that further increases practical flexibility [12]. Nevertheless, even compact predictor-based models can benefit from additional optimization. Deployment is governed not only by parameter count but also by wall-clock latency and activation memory, and naive compression can disproportionately affect components critical to maintaining stable global color behavior, leading to visually noticeable color drift.

This motivates a systematic investigation of pruning as a direct and deployment-oriented compression method for universal-filter-based color correction. Pruning includes both unstructured sparsification and structured removal of hardware-friendly units such as channels or blocks, and it remains one of the most widely used strategies for reducing inference cost [13]–[16]. In this work, we benchmark representative pruning baselines for the predictor network and introduce a configuration-agnostic refinement that customizes sparsity allocation with respect to the functional role and sensitivity of different components. The proposed strategy is designed to reduce compute and memory while preserving visually critical properties of color correction, including stable white balance and consistent tone rendering. We evaluate all pruned variants on standard photo-adjustment benchmarks using established image-quality metrics and deployment-relevant efficiency metrics.

Finally, we note that deployment-oriented model design is a recurring theme across multiple vision domains. In addition to image enhancement and color correction [11], [12], we have studied efficiency-conscious architectures and attention mechanisms for medical imaging segmentation problems [17]–[19]. We have also explored trainable agent-based descriptors for text-rich visual recognition pipelines [20]–[22], which further motivates careful compression and adaptive pruning in modular vision systems.

II. Related work

Classical image enhancement and color correction pipelines have historically relied on hand-crafted priors and parametric transforms, which provide interpretability but are limited in their ability to generalize across diverse illumination, camera settings, and scene content. As learning-based methods matured, research increasingly shifted toward data-driven correction of exposure, tone, and color casts, where models learn mappings that better align with human perception and camera-specific color processing.

One influential direction focused on improving camera color reproduction by revisiting limitations of colorimetric mappings and proposing learned or hybrid mappings that reduce reproduction error across devices and illumination conditions [23]. While such approaches improve physical color fidelity, they often emphasize global accuracy and may still struggle with spatially varying effects, local color shifts, or non-uniform illumination patterns that arise in realistic scenes. Closely related, white-balance correction has remained a central problem because it directly controls chromatic adaptation and global color rendering. Beyond the long-standing color constancy setting, work on correcting improperly white-balanced sRGB images highlighted that post-capture correction is complicated by camera-specific nonlinear processing, and proposed learning-based correction enabled by large paired datasets [24]. However, reliance on specific camera pipelines and supervised paired data can restrict coverage, and robustness under novel capture pipelines or mixed illumination remains challenging.

Parallel to white balance, exposure correction, and low-light restoration methods, learning-based decomposition, illumination estimation, and frequency-aware processing to better preserve structure while stabilizing color appearance. For instance, illumination-driven enhancement for underexposed photography introduced intermediate illumination estimation and tailored losses to achieve more natural exposure, contrast, and color on paired retouching benchmarks [25]. Nevertheless, such methods can be computationally demanding, and their runtime cost can become a barrier when used in real-time camera or mobile pipelines. In a complementary line, local enhancement models aimed to bridge the gap between global parametric adjustment and dense pixel-to-pixel mappings by regressing interpretable local filter parameters, improving editability and sometimes reducing model size [26]. Even so, local parametric filtering may still require nontrivial computation at high resolutions and can be sensitive to deployment constraints. Frequency-based decomposition-and-enhancement frameworks further addressed the coupled nature of low-light enhancement and denoising, arguing that noise behavior differs across frequency components and leveraging this structure for improved restoration [27]. While effective, such multi-stage or multi-component designs can add architectural complexity and increase deployment overhead.

A major obstacle for practical color correction is the presence of spatially varying and mixed illumination. To support progress in this setting, large-scale datasets with pixel-level ground-truth illumination maps and mixture ratios were introduced for multi-illuminant white balancing [28]. However, the learning problem remains difficult because predicting per-pixel illumination under mixed sources can be unstable, and models may produce globally inconsistent color rendering. More recent approaches have introduced attention-based decomposition mechanisms that explicitly separate multiple illuminants and improve consistency, achieving strong results on both single- and multi-illuminant benchmarks [29]. Despite these advances, attention-centric designs tend to increase memory footprint and inference cost, and their deployment can be constrained by hardware limits in real-world applications. Related work also explored reinterpreting lighting as a style factor for auto white-balance correction, offering alternative representations and blending strategies, but these models still

require careful design choices to remain efficient and stable across diverse scenes [30].

Another recurring challenge is maintaining color consistency during enhancement, especially in low-light conditions where methods can improve brightness but introduce color deviations relative to reference images. To mitigate this, color-consistency-aware networks have been proposed to explicitly preserve chromatic statistics and reduce color difference during enhancement [31]. Such methods improve perceptual quality but can add auxiliary modules and losses that increase computational requirements. In addition, recent work has explored controllable tone adjustment using language guidance and pre-trained vision-language representations, enabling flexible edits but introducing new sources of overhead due to conditioning mechanisms and high-capacity backbones [32]. Against this background, universal-filter pipelines represent an alternative point in the design space by combining interpretability with a compact predictor network; recent results demonstrate that such models can remain competitive while being lightweight and amenable to deployment [11], [12].

This motivates compression techniques that can reduce computational cost while preserving visually critical properties of color correction. Pruning has long been used to reduce model complexity, and structured pruning in particular is attractive because it can yield hardware-friendly reductions. Channel pruning via reconstruction-based selection [33] and filter-level pruning based on data-driven statistics [34] demonstrated that substantial acceleration is possible with limited accuracy degradation, but these methods typically target classification backbones and may not transfer optimally to low-level vision predictors whose objectives prioritize perceptual fidelity and stable global color behavior. Automated and platform-aware strategies extended pruning and simplification toward direct deployment metrics, including progressive adaptation under measured constraints [35] and reinforcement-learning-based compression policy search [36]. Meanwhile, structure learning under explicit budgets proposed iterative shrink-and-expand procedures to satisfy computational constraints while retaining accuracy [37]. Despite their relevance, such approaches are not plug-and-play for domain-specific color correction models: low-level tasks often exhibit different sensitivity patterns than recognition models, and naive sparsification can degrade global color consistency or produce visually noticeable artifacts. This motivates configuration-agnostic, adaptive pruning procedures that respect the functional roles of predictor components and preserve stable color behavior under compression.

III. Problem statement

Recent color correction methods demonstrate substantial progress in handling illumination changes, white balance, and color shifts, yet practical deployment remains constrained by computational cost and memory footprint. Many state-of-the-art solutions rely on increasingly complex architectures, attention mechanisms, or multi-component pipelines, which can be difficult to run under strict latency and energy budgets in on-device and real-time imaging workflows. At the same time, generic model compression recipes, and pruning in particular, are typically developed and tuned for high-level recognition backbones and may not transfer reliably to low-level color correction models, where visually noticeable artifacts can be introduced by naive sparsification and where different components contribute unevenly to global color stability.

The goal of this work is to obtain a more resource-efficient variant of a universal-filter-based image color correction model by applying and improving pruning strategies while preserving visually critical properties of color correction, including stable white balance and consistent tone rendering. To achieve this goal, we benchmark representative pruning baselines on the predictor network, quantify their impact on both correction quality and deployment-relevant efficiency metrics, and propose an architecture-aware refinement of the pruning procedure that allocates sparsity with respect to the functional role and sensitivity of model components in order to improve the resulting accuracy-efficiency trade-off.

IV. Proposed solution

A. Baseline pruning strategies

We consider pruning baselines for a universal-filter-based color correction model, where the output image is obtained by composing a set of differentiable, physically meaningful, or visually interpretable transforms. Let $x \in [0,1]^{H\times W\times 3}$ be an input sRGB image and let a predictor network $h(\cdot;\theta)$ produce parameters for a set of N correction operators, $p_{1:N} = h(x;\theta)$. The corrected image is formed as a residual composition

$$\hat{y} = \mathrm{clip}\Big(x + \sum_{i=1}^{N} f_i(x;p_i)\Big), \tag{1}$$

where $f_i(\cdot)$ denotes the i-th filter operator and $\mathrm{clip}(\cdot)$ clamps values to $[0,1]$ per channel. This formulation is attractive for color correction because it combines learnability with explicit control of global color behavior.

The set of operators f_i typically includes global and local color transforms. A canonical global color correction is an affine transform in RGB space,

$$f_{\mathrm{aff}}(x;P,b) = Px + b, \tag{2}$$

where $P \in \mathbb{R}^{3\times 3}$ and $b \in \mathbb{R}^3$ are predicted parameters. Such transforms are widely used in practical pipelines to model device-dependent color mapping and to correct global color casts [23], [24]. Another common component is exposure or gain adjustment, which can be expressed as

$$f_{\mathrm{exp}}(x;\alpha) = \alpha x, \tag{3}$$

and is directly related to underexposure correction pipelines [25]. White balance correction can be represented by a diagonal scaling in an appropriate color space; in RGB form, a simple approximation is

$$f_{\mathrm{wb}}(x;w) = \mathrm{diag}(w)\,x, \quad w \in \mathbb{R}^3_+, \tag{4}$$

which connects to learned white-balance editing and mixed-illumination white balancing [3], [28], [29]. Nonlinear tone

mapping and gamma-like adjustments are often used to control brightness and contrast:

$$f_\gamma(x;\gamma) = x^\gamma, \tag{5}$$

applied element-wise. Local correction can be introduced by predicting spatially varying filter parameters. For example, local parametric filtering can be written as

$$f_{\text{local}}(x;\phi) = \mathcal{F}\big(x;\phi(x)\big), \tag{6}$$

where $\phi(x)$ are per-pixel (or low-resolution) parameters predicted by the network, enabling local tone and color adjustment [26]. Such local behavior is useful for difficult illumination conditions, but it also increases sensitivity to compression because local artifacts can become visually noticeable.

Given the above structure, pruning is applied to the predictor network $h(\cdot;\theta)$ rather than to the fixed functional form of the operators. Let $m \in \{0,1\}^{|\theta|}$ be a binary mask and $\theta' = \theta \odot m$. The pruned model yields

$$p_{1:N} = h(x;\theta'), \qquad \hat{y} = \text{clip}\Big(x + \sum_{i=1}^{N} f_i(x;p_i)\Big). \tag{7}$$

A generic baseline pruning objective is to reduce the number of active parameters under a constraint while maintaining correction quality:

$$\min_{\theta,m} \ \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\ell(\hat{y},y)\Big] \quad \text{s.t.} \quad \|m\|_0 \le s, \tag{8}$$

where $\ell(\cdot,\cdot)$ is a training loss consistent with color correction quality criteria and s is the sparsity budget.

We benchmark two representative baseline families. The first is unstructured magnitude pruning, where weights with small magnitude are removed, and the model is fine-tuned [13], [14]:

$$m_j = \mathbb{I}\big(|\theta_j| \ge \tau\big), \tag{9}$$

with τ chosen to match the target sparsity. The second family is structured pruning, which removes hardware-friendly units such as convolutional channels. Denoting by $\{\mathcal{G}_k\}$ groups corresponding to channels, one may rank and remove groups using norm-based or sparsity-inducing criteria [15], [16], [33], [34]. In addition, platform-aware baselines incorporate deployment constraints and can prioritize structural changes that improve measured latency [35]–[37].

Importantly, in color correction, the quality degradation induced by pruning can manifest not only as a metric drop but also as visually salient artifacts such as global hue drift, unstable white balance, or local banding. Therefore, we evaluate baseline pruning strategies in terms of both efficiency and preservation of stable global color behavior, and use them as reference points for the customized strategy introduced in the next subsection.

B. Customize pruning strategies

Baseline pruning methods reduce predictor size and runtime, but for universal-filter-based color correction, they may not preserve perceptually critical properties. The predictor controls both global operators, such as affine color transforms and white balance gains, and local adjustments that affect tone and chroma, and these components can differ substantially in pruning sensitivity. Consequently, uniform sparsification can degrade global color stability, cause hue drift, or amplify local artifacts even under moderate compression.

We therefore propose a customized pruning strategy tailored to the universal-filter pipeline. The strategy refines a baseline pruning procedure by introducing an additional mechanism for sparsity allocation and fine-tuning that targets preservation of stable global color behavior under a fixed efficiency budget. It is lightweight, compatible with the evaluated baselines, and reproducible with a small set of hyperparameters.

C. Customized pruning strategy for universal-filter color correction

Baseline pruning methods can reduce the computational footprint of the predictor network in universal-filter-based color correction, but their uniform application often fails to preserve perceptually critical properties. In color correction, the predictor controls two distinct families of operators: global transforms (affine color matrices, white-balance gains, exposure adjustments) that determine the overall chromatic appearance, and local parametric filters that refine tonal and color details at finer spatial scales. These components exhibit markedly different sensitivity to pruning: degradation in global operators can manifest as hue drift or unstable white balance across the entire image, while excessive pruning of local pathways may produce spatially inconsistent corrections or visible artifacts in detailed regions. To address this, we propose a customized pruning strategy that incorporates two architecture-aware mechanisms: (i) spatial importance weighting that accounts for the non-uniform distribution of diagnostically (or perceptually) relevant information across the image field, and (ii) component-specific sparsity allocation that distinguishes between global and local processing pathways.

1) Spatial importance weighting via activation norms: A key observation in image correction pipelines is that not all spatial regions contribute equally to the final perceptual quality. In typical photographic compositions, the central region of the image often contains the primary subject, while peripheral areas may include background elements where minor color deviations are less noticeable. This spatial non-uniformity provides an opportunity for targeted pruning: we can allocate computational resources preferentially to regions where accurate processing matters most, while reducing capacity in less critical areas.

We operationalize this insight by introducing a spatial importance map derived from activation statistics. Let $F \in \mathbb{R}^{C\times H\times W}$ denote a feature map at a given depth in the predictor network. For each spatial position (i,j), we compute an

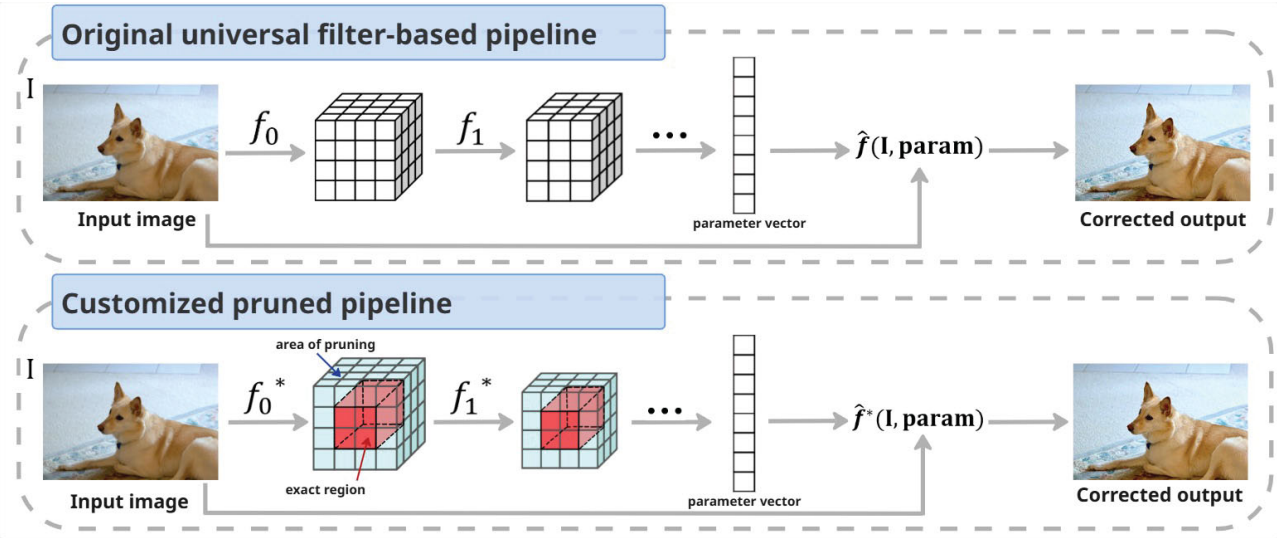

Fig. 1. Original and customized pruned universal-filter pipelines. The predictor generates correction parameters via transformations $\{f_k\}$, while the pruned variant replaces them with $\{f_k^*\}$ and reduces computations in the pruned feature region, preserving the core representation used for color correction.

importance weight based on the average activation magnitude across channels and across a calibration set $\mathcal{D}_{\text{cal}}$:

$$W(i,j) = \frac{1}{N \cdot C} \sum_{n=1}^{N} \sum_{c=1}^{C} \left| F_{c,i,j}^{(n)} \right|, \tag{10}$$

where N is the number of calibration samples. This map captures regions that consistently produce strong activations, which typically correspond to image areas with complex structure or significant color variation requiring fine-grained adjustment. Empirically, we observe that such regions are concentrated near the image center, reflecting the compositional bias in natural photography.

To leverage this spatial importance for pruning, we incorporate $W(i,j)$ into the importance scoring for structured units (e.g., convolutional channels). For a channel c, we compute a spatially weighted activation norm:

$$I_{\text{spatial}}(c) = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{H} \sum_{j=1}^{W} W(i,j) \cdot \left| F_{c,i,j}^{(n)} \right|. \tag{11}$$

Channels with low $I_{\text{spatial}}(c)$ are those that contribute primarily to regions of low perceptual importance; such channels can be pruned with minimal impact on perceived correction quality. This approach preserves model capacity for processing image centers where accurate color and tone are most critical, while allowing aggressive pruning in peripheral regions.

2) Spatially-adaptive feature computation: Beyond channel-level pruning, we introduce a finer-grained mechanism that dynamically suppresses computation on spatially localised background regions. For each spatial position (i,j), we compute a binary mask indicating whether to retain or skip processing:

$$M(i,j) = \mathbb{I}\left(\frac{1}{C} \sum_{c=1}^{C} |F_{c,i,j}| \geq \tau_{\text{spatial}} \right), \tag{12}$$

where τ_{spatial} is a threshold calibrated to achieve a desired level of spatial sparsity. Positions with activation magnitude below the threshold are considered low-importance and are masked out in subsequent layers:

$$\tilde{F}_{:,i,j} = M(i,j) \cdot F_{:,i,j}. \tag{13}$$

This operation is implemented efficiently by skipping computation at masked positions in convolutional and element-wise operations, yielding proportional reductions in FLOPs. Importantly, the masking is applied identically across all channels for a given position, preserving the representational capacity in retained regions while eliminating unnecessary computation in background areas.

3) Component-sensitive sparsity allocation: The universal-filter architecture comprises distinct components with different functional roles: the global parameter predictor (which estimates affine transforms, white-balance gains, and exposure adjustments) and the local refinement network (which predicts spatially varying filter parameters). These components exhibit asymmetric sensitivity to pruning. Global parameters affect the entire image uniformly; errors in these estimates cannot be compensated for by local adjustments and lead to global color casts. Local parameters, while important for fine detail, operate at smaller spatial scales, and minor inaccuracies may be less perceptible.

We quantify this sensitivity using a Taylor-based importance score computed separately for each component. Let $\mathcal{L}$ denote the training loss (e.g., L2 or perceptual loss). For a parameter group $\mathcal{G}$ corresponding to a specific component (e.g., all weights in the global predictor head), we estimate its contribution as:

$$S_{\mathcal{G}} = \sum_{\theta \in \mathcal{G}} \left| \theta \cdot \frac{\partial \mathcal{L}}{\partial \theta} \right|, \tag{14}$$

evaluated on the calibration set. Components with higher $S_{\mathcal{G}}$ are deemed more critical and are assigned lower target sparsity. We then solve a constrained optimization to allocate sparsity budgets across components:

$$\min_{\{s_k\}} \sum_k S_k \cdot s_k \quad \text{s.t.} \quad \frac{\sum_k p_k \cdot (1 - s_k)}{\sum_k p_k} = t, \qquad (15)$$

where s_k is the target sparsity for component k, p_k is its parameter count, and t is the overall parameter retention target. This formulation preferentially prunes less sensitive components, preserving capacity where it matters most for perceptual quality.

4) Integration with discriminator and regressor pruning: In our universal-filter pipeline, the predictor network $h(\cdot;\theta)$ can be viewed as having dual roles: it acts as a regressor producing continuous filter parameters, and its intermediate representations may be used by an auxiliary discriminator in adversarial training setups. Pruning must account for both functions. The spatial importance weighting described above naturally preserves representations needed for accurate regression in perceptually critical regions. For discriminator-related capacity, we observe that discriminator sensitivity is also spatially non-uniform: adversarial feedback primarily targets regions where realistic texture and color are essential, which align with the high-importance areas identified by $W(i, j)$. Thus, the same spatial importance map serves to protect features needed for adversarial training without additional complexity.

5) Pruning schedule and fine-tuning: The proposed pruning strategy is implemented within an iterative pruning-fine-tuning framework. Starting from a pretrained baseline, we alternate between pruning steps (removing a small fraction of channels or spatial positions based on the criteria above) and fine-tuning steps to recover quality. This gradual approach allows the model to adapt to the changing architecture and mitigates capacity loss. For spatial masking, we introduce the masks gradually, starting with conservative thresholds and increasing spatial sparsity over multiple iterations to avoid abrupt changes in the feature distribution.

6) Summary: The proposed customized pruning strategy integrates three complementary mechanisms tailored to universal-filter color correction: (i) spatial importance weighting that prioritizes channels contributing to perceptually critical image regions, (ii) spatially-adaptive feature computation that suppresses processing in low-importance areas, and (iii) component-sensitive sparsity allocation that distinguishes between global and local processing pathways. Together, these refinements are designed to preserve SSIM and perceptual fidelity under aggressive pruning by focusing computational resources where they contribute most to final image quality. The following section evaluates this strategy against standard pruning baselines on the MIT-Adobe FiveK benchmark.

V. Experiments

A. Dataset description

We evaluate baseline and customized pruning strategies on the MIT-Adobe FiveK benchmark, a widely used dataset for learned photo adjustment and color correction [38]. The dataset contains 5,000 photographs and, for each original, five expert-retouched versions produced by trained photographers (A, B, C, D, and E). Following the common protocol in automatic enhancement and color correction research, we adopt the expert C rendition as the target reference for supervised training and evaluation [38]. Fig. 2 shows representative examples of input images from the dataset.

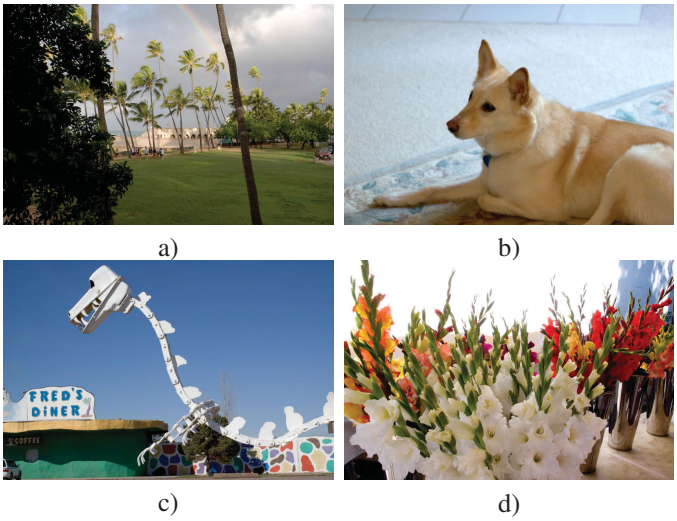

Fig. 2. Example input photographs from the MIT-Adobe FiveK dataset [38] used in our experiments.

To ensure comparability with prior work on universal-filter-based pipelines, we use the standard RANDOM250 subset for testing, while the remaining images are used for training [39]. All training, validation, and evaluation operations are performed in the RGB color space. During preprocessing, images are padded to a spatial resolution of 500×500 and normalized to the range $[0, 1]$. This setup allows us to measure how pruning affects both correction quality and deployment-relevant efficiency on a canonical color correction benchmark.

B. Training details

All models were implemented in TensorFlow 2 [40]. Inference latency was measured using a PyTorch-based timing script to ensure consistent wall-clock evaluation across pruned variants [41]. Training was performed with the Adam optimizer using $\beta_1 = 0$ and $\beta_2 = 0.9$, and a batch size of 40. We initialized the learning rate to 10^{-3} and applied exponential decay with a factor of 0.95 every 1200 optimization steps. Unless stated otherwise, training was run for 10,000 steps with the same optimization settings for the unpruned baseline and all pruning configurations, followed by fine-tuning of pruned models under the identical schedule. All experiments were executed on a workstation equipped with a single NVIDIA

GeForce RTX 3060 GPU and an Intel Core i5-10400 CPU (2.90 GHz).

C. Experimental results

We evaluate pruning effects for universal-filter-based color correction on the MIT-Adobe FiveK benchmark. All pruning strategies are applied to the same baseline predictor architecture, enabling a direct comparison of sparsification effects. We report PSNR and SSIM for quality assessment, and we additionally report deployment-relevant efficiency measures, including the number of trainable parameters, FLOPs, and inference latency under a fixed hardware setup.

PSNR is computed from the mean squared error (MSE) between the predicted image $\hat{y}$ and the reference image y as

$$\mathrm{MSE}(\hat{y}, y) = \frac{1}{3HW} \sum_{c=1}^{3} \sum_{u=1}^{H} \sum_{v=1}^{W} (\hat{y}_{u,v,c} - y_{u,v,c})^2, \quad (16)$$

$$\mathrm{PSNR}(\hat{y}, y) = 10 \log_{10} \left(\frac{L^2}{\mathrm{MSE}(\hat{y}, y)} \right), \quad (17)$$

where H and W denote image height and width, and L is the maximum possible pixel value. Since all images are normalized to $[0, 1]$, we set $L = 1$.

SSIM measures structural similarity by comparing local statistics of $\hat{y}$ and y over image windows. In its standard form, SSIM is defined as

$$\mathrm{SSIM}(\hat{y}, y) = \frac{(2\mu_{\hat{y}}\mu_y + c_1)(2\sigma_{\hat{y}y} + c_2)}{(\mu_{\hat{y}}^2 + \mu_y^2 + c_1)(\sigma_{\hat{y}}^2 + \sigma_y^2 + c_2)}, \quad (18)$$

where $\mu_{\hat{y}}$ and μ_y are local means, $\sigma_{\hat{y}}^2$ and σ_y^2 are local variances, $\sigma_{\hat{y}y}$ is the local covariance, and c_1, c_2 are small constants for numerical stability. We report SSIM averaged over all test images.

The results are summarized in Table I for quality and Table II for efficiency. The tables are provided with placeholders and will be populated after completing the full experimental sweep.

TABLE I. QUALITY RESULTS

Method	Sparsity	PSNR (dB)	SSIM
Base	0%	24.38	0.921
GMP	30%	24.11	0.916
GMP	50%	23.59	0.907
GMP	60%	23.24	0.899
Structured Channel	30%	24.19	0.918
Structured Channel	50%	23.82	0.911
Structured Channel	60%	23.52	0.904
Latency-aware Structured	30%	24.23	0.919
Latency-aware Structured	50%	23.94	0.913
Latency-aware Structured	60%	23.68	0.905
Customized sparsity allocation	30%	24.31	0.920
Customized sparsity allocation	50%	24.06	0.916
Customized sparsity allocation	60%	23.89	0.910

TABLE II. EFFICIENCY RESULTS

Method	Sparsity	Params (K)	FLOPs (M)	Latency (ms)
Base	0%	420	85.4	8.5
GMP	30%	296	85.0	8.4
GMP	50%	188	84.6	8.2
GMP	60%	170	84.3	8.1
Structured Channel	30%	307	63.5	7.0
Structured Channel	50%	212	48.7	5.7
Structured Channel	60%	171	41.2	5.1
Latency-aware Structured	30%	301	58.9	6.5
Latency-aware Structured	50%	203	44.1	5.0
Latency-aware Structured	60%	173	37.5	4.7
Customized sparsity allocation	30%	302	54.8	6.1
Customized sparsity allocation	50%	198	36.3	4.4
Customized sparsity allocation	60%	169	31.6	4.0

VI. CONCLUSION

In this work, we investigated pruning as a practical approach to improve the deployability of universal-filter-based image color correction models. The focus was on a compact predictor-driven pipeline that composes interpretable correction operators, where efficiency constraints are central for on-device and real-time usage and where compression-induced color artifacts can be perceptually salient.

Our experiments on the MIT-Adobe FiveK benchmark confirm that pruning can substantially reduce the computational footprint of the predictor, but the achieved quality-efficiency trade-off depends strongly on the pruning strategy. Unstructured GMP reduces the parameter count (from 420K to 188K at 50% sparsity) but provides only marginal latency improvements (8.5 ms to 8.2 ms), indicating limited practical acceleration without structured sparsity. Structured channel pruning and latency-aware structured pruning yield meaningful reductions in FLOPs and latency, but they introduce a larger drop in PSNR and SSIM as sparsity increases.

In contrast, the proposed customized sparsity allocation consistently provides a more favorable trade-off at moderate-to-high sparsity levels. At 50% sparsity, it improves latency from 8.5 ms to 4.4 ms and reduces FLOPs from 85.4M to 36.3M while preserving higher image quality than the baseline pruning strategies (PSNR = 24.06, SSIM = 0.916). At 60% sparsity, the proposed method further reduces latency to 4.0 ms and FLOPs to 31.6M while maintaining strong quality (PSNR = 23.89, SSIM = 0.910), outperforming the compared baselines at similar compression levels. These results support the conclusion that configuration-agnostic, adaptive sparsity allocation is important for universal-filter color correction, enabling substantial acceleration while preserving stable color behavior.

The proposed approach improves the practical deployability of universal-filter-based color correction models without sacrificing visually critical performance. Future work will validate the strategy across additional datasets and camera pipelines and will explore integration with complementary compression techniques and device-specific optimization objectives.

REFERENCES

[1] J. Yan, S. Lin, S. Bing Kang, and X. Tang, "A learning-to-rank approach for image color enhancement," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[2] Y. Hu, B. Wang, and S. Lin, "Fc4: Fully convolutional color constancy with confidence-weighted pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[3] M. Afifi and M. S. Brown, "Deep white-balance editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[4] Y. Li, K. Xu, G. P. Hancke, and R. W. Lau, "Color shift estimation-and-correction for image enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 25 389–25 398.

[5] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully-convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[6] A. Ignatov, N. Kobyshev, R. Timofte, K. Vanhoey, and L. Van Gool, "Dslr-quality photos on mobile devices with deep convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[7] 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)*, June 2018.

[8] C. Guo, C. Li, J. Guo, C. C. Loy, J. Hou, S. Kwong, and R. Cong, "Zero-reference deep curve estimation for low-light image enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[9] R. Liu, L. Ma, J. Zhang, X. Fan, and Z. 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 (CVPR)*, June 2021, pp. 10 561–10 570.

[10] H. Fu, W. Zheng, X. Meng, X. Wang, C. Wang, and H. Ma, "You do not need additional priors or regularizers in retinex-based low-light image enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 18 125–18 134.

[11] A. Samarin, A. Nazarenko, A. Savelev, A. Toropov, A. Dzestelova, E. Mikhailova, A. Motyko, and V. Malykh, "A model based on universal filters for image color correction," *Pattern Recognition and Image Analysis*, vol. 34, no. 3, pp. 844–854, 2024. [Online]. Available: https://doi.org/10.1134/S1054661824700731

[12] A. Samarin, A. Nazarenko, A. Toropov, E. Kotenko, A. Dzestelova, E. Mikhailova, V. Malykh, A. Savelev, and A. Motyko, "Universal filter-based lightweight image enhancement model with unpaired learning mode," in *2024 36th Conference of Open Innovations Association (FRUCT)*, 2024, pp. 711–720.

[13] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," 2016. [Online]. Available: https://arxiv.org/abs/1510.00149

[14] T. Gale, E. Elsen, and S. Hooker, "The state of sparsity in deep neural networks," 2019. [Online]. Available: https://arxiv.org/abs/1902.09574

[15] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," in *International Conference on Learning Representations (ICLR)*, 2017, poster. [Online]. Available: https://openreview.net/forum?id=rJqFGTslg

[16] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[17] A. Samarin, A. Toropov, A. Dzestelova, A. Nazarenko, E. Kotenko, E. Mikhailova, A. Savelev, and A. Motyko, "Specialized non-local blocks for recognizing tumors on computed tomography snapshots of human lungs," in *2024 35th Conference of Open Innovations Association (FRUCT)*, 2024, pp. 659–664.

[18] A. Samarin, A. Savelev, A. Toropov, A. Nazarenko, A. Golovatiuk, P. Dmitriev, A. Dzestelova, E. Mikhailova, A. Motyko, and V. Malykh, "Segmentation of the iris and pupil of the human eye in images from an infrared camera," *Pattern Recognition and Image Analysis*, vol. 34, no. 3, pp. 855–862, 2024. [Online]. Available: https://doi.org/10.1134/S1054661824700743

[19] A. Samarin, A. Toropov, and O. Egorova, "Self-attention based approach to iris segmentation," in *2025 International Russian Smart Industry Conference (SmartIndustryCon)*, 2025, pp. 200–205.

[20] A. Samarin, A. Savelev, A. Toropov, A. Dzestelova, V. Malykh, E. Mikhailova, and A. Motyko, "Trainable agents movement strategies for advertising sign visual descriptors," *Pattern Recognition and Image Analysis*, vol. 32, no. 3, pp. 651–657, 2022. [Online]. Available: https://doi.org/10.1134/S1054661822030373

[21] A. Samarin, A. Savelev, A. Toropov, A. Dzestelova, V. Malykh, A. Motyko, and E. Mikhailova, "Predictors based on convolutional neural networks for the movement strategy of trainable agents for building customized image descriptors," *Pattern Recognition and Image Analysis*, vol. 33, no. 2, pp. 139–146, 2023. [Online]. Available: https://doi.org/10.1134/S105466182302013X

[22] A. Samarin, A. Savelev, A. Toropov, A. Dzestelova, V. Malykh, E. Mikhailova, and A. Motyko, "The complete study of the movement strategies of trained agents for visual descriptors of advertising signs," in *Pattern Recognition, Computer Vision, and Image Processing. ICPR 2022 International Workshops and Challenges*, J.-J. Rousseau and B. Kapralos, Eds. Cham: Springer Nature Switzerland, 2023, pp. 571–585.

[23] H. C. Karaimer and M. S. Brown, "Improving color reproduction accuracy on cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[24] M. Afifi, B. Price, S. Cohen, and M. S. Brown, "When color constancy goes wrong: Correcting improperly white-balanced images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[25] R. Wang, Q. Zhang, C.-W. Fu, X. Shen, W.-S. Zheng, and J. Jia, "Underexposed photo enhancement using deep illumination estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[26] S. Moran, P. Marza, S. McDonagh, S. Parisot, and G. Slabaugh, "Deeplpf: Deep local parametric filters for image enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[27] K. Xu, X. Yang, B. Yin, and R. W. Lau, "Learning to restore low-light images via decomposition-and-enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[28] D. Kim, J. Kim, S. Nam, D. Lee, Y. Lee, N. Kang, H.-E. Lee, B. Yoo, J.-J. Han, and S. J. Kim, "Large scale multi-illuminant (lsmi) dataset for developing white balance algorithm under mixed illumination," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 2410–2419.

[29] D. Kim, J. Kim, J. Yu, and S. J. Kim, "Attentive illumination decomposition model for multi-illuminant white balancing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 25 512–25 521.

[30] F. Kınlı, D. Yılmaz, B. Özcan, and F. Kıraç, "Modeling the lighting in scenes as style for auto white-balance correction," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 4903–4913.

[31] Z. Zhang, H. Zheng, R. Hong, M. Xu, S. Yan, and M. Wang, "Deep color consistent network for low-light image enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 1899–1908.

[32] H. Lee, K. Kang, J. Ok, and S. Cho, "Cliptone: Unsupervised learning for text-based image tone adjustment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 2942–2951.

[33] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[34] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[35] T.-J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, M. Sandler, V. Sze, and H. Adam, "Netadapt: Platform-aware neural network adaptation for mobile applications," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[36] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "Amc: Automl for model compression and acceleration on mobile devices," in *Proceedings*

of the European Conference on Computer Vision (ECCV), September 2018.

[37] A. Gordon, E. Eban, O. Nachum, B. Chen, H. Wu, T.-J. Yang, and E. Choi, "Morphnet: Fast simple resource-constrained structure learning of deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[38] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input / output image pairs," in *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

[39] J. Park, J.-Y. Lee, D. Yoo, and I. S. Kweon, "Distort-and-recover: Color enhancement using deep reinforcement learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[40] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2016. [Online]. Available: https://arxiv.org/abs/1603.04467

[41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf