ISP Meets Deep Learning: A Survey on Deep Learning Methods for Image Signal Processing

CLAUDIO FILIPI GONCALVES DOS SANTOS, Computer Science, UFSCar, Sao Carlos, Brazil and Eldorado Institute, Campinas, Brazil
RODRIGO REIS ARRAIS, Eldorado Institute, Campinas, Brazil
JHESSICA VICTORIA SANTOS DA SILVA, Eldorado Institute, Campinas, Brazil
MATHEUS HENRIQUE MARQUES DA SILVA, Eldorado Institute, Campinas, Brazil
WLADIMIR BARROSO GUEDES DE ARAUJO NETO, Eldorado Institute, Campinas, Brazil
LEONARDO TADEU LOPES, Eldorado Institute, Campinas, Brazil
GUILHERME AUGUSTO BILEKI, Eldorado Institute, Campinas, Brazil
IAGO OLIVEIRA LIMA, Eldorado Institute, Campinas, Brazil
LUCAS BORGES RONDON, Eldorado Institute, Campinas, Brazil
BRUNO MELO DE SOUZA, Eldorado Institute, Campinas, Brazil
MAYARA COSTA REGAZIO, Eldorado Institute, Campinas, Brazil
RODOLFO COELHO DALAPICOLA, Eldorado Institute, Campinas, Brazil
ARTHUR ALVES TASCA, Eldorado Institute, Campinas, Brazil

The entire Image Signal Processor (ISP) of a camera relies on several processes to transform the data from the Color Filter Array (CFA) sensor, such as demosaicing, denoising, and enhancement. These processes can be executed either by some hardware or via software. In recent years, Deep Learning (DL) has emerged as one solution for some of them or even to replace the entire ISP using a single neural network for the task. In this work, we investigated several recent pieces of research in this area and provide deeper analysis and comparison among them, including results and possible points of improvement for future researchers.

Matheus Henrique Marques Da Silva, Jhessica Victoria Santos Da Silva, and Rodrigo Reis Arrais contributed equally to this research.

Authors' Contact Information: Claudio Filipi Goncalves dos Santos, Computer Science, UFSCar, Sao Carlos, Brazil and Eldorado Institute, Campinas, São Paulo, Brazil; e-mail: cfsantos85@gmail.com; Rodrigo Reis Arrais, Eldorado Institute, Campinas, São Paulo, Brazil; e-mail: rodrigo.arrais@eldorado.org.br; Jhessica Victoria Santos da Silva, Eldorado Institute, Campinas, São Paulo, Brazil; e-mail: jhessica.silva@eldorado.org.br; Matheus Henrique Marques da Silva, Eldorado Institute, Campinas, São Paulo, Brazil; e-mail: matheus.marques@eldorado.org.br; Wladimir Barroso Guedes de Araujo Neto, Eldorado Institute, Campinas, São Paulo, Brazil; e-mail: wladimir.neto@eldorado.org.br; Leonardo Tadeu Lopes, Eldorado Institute, Campinas, São Paulo, Brazil; e-mail: leonardo.lopes@eldorado.org.br; Guilherme Augusto Bileki, Eldorado Institute, Campinas, São Paulo, Brazil; e-mail: bilekig@eldorado.org.br; Iago Oliveira Lima, Eldorado Institute, Campinas, Brazil; e-mail: iagolima@eldorado.org.br; Lucas Borges Rondon, Eldorado Institute, Campinas, São Paulo, Brazil; e-mail: lucas.rondon@eldorado.org.br; Bruno Melo de Souza, Eldorado Institute, Campinas, São Paulo, Brazil; e-mail: brunobms@eldorado.org.br; Mayara Costa Regazio, Eldorado Institute, Campinas, São Paulo, Brazil; e-mail: mayara.regazio@eldorado.org.br; Rodolfo Coelho Dalapicola, Eldorado Institute, Campinas, São Paulo, Brazil; e-mail: rodolfo.dalapicola@eldorado.org.br; Arthur Alves Tasca, Eldorado Institute, Campinas, São Paulo, Brazil; e-mail: atasca@eldorado.org.br.

ACM 0360-0300/2025/01-ART127
https://doi.org/10.1145/3708516

CCS Concepts: • **Computing methodologies → Image and video acquisition;**

Additional Key Words and Phrases: Image signal processing, deep learning, convolutional neural networks

ACM Reference Format:
Claudio Filipi Goncalves dos Santos, Rodrigo Reis Arrais, Jhessica Victoria Santos da Silva, Matheus Henrique Marques da Silva, Wladimir Barroso Guedes de Araujo Neto, Leonardo Tadeu Lopes, Guilherme Augusto Bileki, Iago Oliveira Lima, Lucas Borges Rondon, Bruno Melo de Souza, Mayara Costa Regazio, Rodolfo Coelho Dalapicola, and Arthur Alves Tasca. 2025. ISP Meets Deep Learning: A Survey on Deep Learning Methods for Image Signal Processing. *ACM Comput. Surv.* 57, 5, Article 127 (January 2025), 44 pages. https://doi.org/10.1145/3708516

1 Introduction

The **Image Signal Processor (ISP)** is a component of digital cameras capable of performing various tasks to improve image quality, such as demosaicing, denoising, and white balance. The set of tasks performed by the ISP is called ISP pipeline, divided in preprocessing and postprocessing steps, and may differ from manufacturer to manufacturer [90]. Nowadays, Machine Learning can be used to replace partial or the entire ISP pipeline. Particulary, Deep Learning is employed to replace ISP tasks, working on noise removal or some image feature that hinders processing over the network. Deep Learning network provides an improvement in relation to computational efficiency and processing time. This survey paper aims to analyze recent studies that implemented Deep Learning based ISP pipeline.

1.1 Image Signal Processing

Traditionally, ISPs are digital signal processors that reconstruct **RGB (Red-Green-Blue)** images from RAW images. In traditional camera pipelines, complex and proprietary hardware processes are used to perform image signal processing [140]. It consists of several processing steps, including noise reduction, white balance, demosaicing, and more. Each step with loss functions in the ISP is usually performed sequentially, the residual error accumulates over the runtime [91]. Parameter adjustments in later stages correct the accumulated errors.

Most of the traditional methods use heuristic approaches to derive the solution at each step of the ISP pipeline [140], so numerous parameters need to be adjusted. Moreover, multiple ISP processes execute sequentially with module-based algorithms leading to cumulative errors at each execution step. To minimize those errors, new techniques are researched and, among them, algorithms related to Deep Learning started to get more focus.

1.2 Deep Learning

Even though the research of Machine Learning dates back to the decade of 1950 [94], it was only in the decade of 2010 that the advancements in technology have allowed its more complex fields to be extensively explored. The rapid evolution of computational power, coupled with the growing amount of data being produced daily, caused a subtle renewal of interest in the usage of Machine Learning techniques. For that reason, several areas such as Chemistry [104], Medicine [22], Economics [106], and Physics [5], could harness its capacities to accelerate or improve their work, directly impacted by this evolution in the field known as Deep Learning.

Deep Learning, as a subset of Machine Learning, is comprised of algorithms based on **Artificial Neural Networks (ANN)** that use several layers of neurons to extract higher level features from the raw data being provided to it [14, 32, 99]. This class of algorithms requires a huge amount

of computational power that have become available only in recent years. In parallel to the high demand of computational power, the capacity of learning of Deep Learning algorithms also rises with the amount of data provided to the system. For this reason, areas that have a great influx of data in their operation saw in Deep Learning an interesting way to find and understand hidden information.

1.3 The ISP and Deep Learning Relation

The intention of using an ANN to replace the hardware-based ISP is justified by the fact that an ANN can compensate for the loss of information in the input images making it more reliable than the traditionally implemented ISP, as traditional ISP is known to accumulate errors at each step [90]. Ignatov et al. [49] was one of the first to propose a CNN, a subgroup of ANN, in place of a smartphone ISP camera and provided a RAW-to-RGB dataset using the PyNet network. These demonstrated the potential of CNN for image processing as a replacement for even the most sophisticated ISPs.

CNNs not only show significant advantages in low-level vision tasks [140], they also show good results in high-level tasks such as object detection and segmentation [15]. With these advantages, the use of CNNs for transforming RAW images into RGB images became possible. Despite the good results, there are few works using CNN as a replacement for ISP. Ratnasingam [91] showed the difficulties in performing the necessary adjustments in traditional ISP pipelines and developed a CNN that performs the ISP pipeline.

1.4 Comparative Analysis: Step-by-Step vs. End-to-End Approaches

While the discussion highlights the advantages and disadvantages of single-step and end-to-end approaches, a more detailed empirical analysis would provide deeper insights. For instance, the study by Ignatov et al. [49] utilized the PyNET network to replace the entire ISP pipeline, achieving superior results in end-to-end optimization with a PSNR of 27.12 dB on the Zurich Dataset. In contrast, single-step approaches like SGNet, which focus on specific tasks such as demosaicing and denoising, achieved a PSNR of 26.85 dB while offering better interpretability in noise control. This comparison underscores that end-to-end models can achieve superior global optimization, while single-step models excel in fine-tuning specific stages, particularly in scenarios requiring domain-specific adjustments.

In the domain of ISP using Deep Learning, two primary methodologies are often considered: the step-by-step (or modular) approach and the end-to-end approach. Each of these methods has its own distinct advantages and disadvantages, which are critical to consider depending on the application and the specific challenges being addressed.

1.4.1 Step-by-Step Approach. The step-by-step approach involves decomposing the ISP pipeline into multiple stages, where each stage is handled by a separate model or a set of models. This method offers several advantages:

- **Flexibility and Control:** Each stage of the pipeline can be individually optimized and adjusted, allowing for a high degree of control over the processing of specific image features. This can be particularly beneficial in scenarios where domain-specific knowledge is available for fine-tuning each step.
- **Interpretability:** By breaking down the process into distinct stages, it becomes easier to interpret the effects of each stage on the final output. This interpretability can be crucial for debugging and for understanding the model's behavior in detail.
- **Modularity:** Different models or algorithms can be used for different stages, allowing for modular updates or changes without affecting the entire system.

However, the step-by-step approach also comes with certain disadvantages:

— **Complexity:** Managing and training multiple models for different stages can significantly increase the complexity of the system. This can lead to longer development times and greater computational resource requirements.
— **Suboptimal Global Performance:** Since each stage is optimized individually, the overall performance may not be globally optimal. The local optimization of each stage might not necessarily lead to the best end-to-end results.
— **Data Dependency:** The approach often requires a large amount of annotated data for each stage, making it challenging to implement in data-scarce environments.

1.4.2 End-to-End Approach. In contrast, the end-to-end approach involves training a single model that learns the entire ISP pipeline as a unified process. The key advantages of this method include:

— **Simplicity:** The pipeline is streamlined into a single model, simplifying the implementation and reducing the complexity associated with managing multiple models.
— **Global Optimization:** The model is trained to optimize the final output directly, potentially leading to better overall performance as the optimization is global rather than stage-specific.
— **Reduced Manual Intervention:** This approach requires less manual design and fine-tuning of individual stages, as the model learns the optimal processing automatically from the data.

Despite its advantages, the end-to-end approach also presents certain challenges:

— **Diagnostic Challenges:** Since the entire pipeline is encapsulated within a single model, diagnosing specific issues can be more difficult. If the model fails, it may be challenging to pinpoint which part of the process is responsible.
— **Data Requirements:** End-to-end models often require large amounts of labeled data to achieve high performance, which can be a limitation in scenarios with limited data availability.
— **Overfitting Risk:** The model may overfit to specific datasets, especially if there is a lack of diversity in the training data, which could reduce its generalizability to new data [98].

While the discussion highlights the advantages and disadvantages of single-step and end-to-end approaches, a more detailed empirical analysis would provide deeper insights. For instance, the PyNET network [46] exemplifies the end-to-end approach by replacing the entire ISP pipeline, achieving a PSNR of 27.12 dB on the Zurich Dataset, optimizing all stages simultaneously for global performance. In contrast, DRDN [83], a single-step approach focused on demosaicing, demonstrated a PSNR of 25.34 dB while significantly reducing computational complexity through the use of residual and densely connected layers. This comparison illustrates that end-to-end models like PyNET achieve superior global optimization, while single-step models such as DRDN provide more efficient and interpretable solutions for specific ISP tasks, particularly in resource-constrained environments.

In conclusion, the choice between the step-by-step and end-to-end approaches should be guided by the specific needs of the application, the availability of data, and the desired balance between flexibility and simplicity. While the step-by-step approach offers fine-grained control and interpretability, the end-to-end approach provides a more streamlined and globally optimized solution. A careful consideration of these factors will lead to the most effective implementation for ISP tasks using deep learning.

1.5 Selection of Works for Review

The selection of papers for this survey was guided by the goal of providing a comprehensive overview of state-of-the-art deep learning approaches applied to ISP. We carefully selected 30 papers that represent key advancements in the field, focusing on both general ISP tasks and specialized subtasks such as joint demosaicing-denoising, resolution enhancement, raw-to-RGB mapping, and image augmentation.

The criteria for inclusion in this survey were as follows:

- **Relevance to ISP tasks:** The selected works address critical ISP subtasks that are fundamental to the image processing pipeline, including demosaicing, denoising, and resolution enhancement. These methods aim to improve the quality of images captured by modern sensors, which is essential for both consumer-grade cameras and specialized imaging systems.
- **Diversity of approaches:** To ensure a broad representation of techniques, we included methods that utilize different types of neural network architectures, such as CNNs, residual learning-based models, and transformer-based architectures. This selection highlights the variety of ways in which deep learning is being applied to ISP problems.
- **Performance and innovation:** The papers chosen for this survey have reported significant improvements over traditional ISP methods, either through innovative architectural design or by addressing specific challenges in the ISP pipeline. We prioritized works that demonstrated clear advancements in terms of performance metrics or novel approaches to processing raw sensor data.
- **Application range:** The selected works cover a wide range of applications, from real-time processing for mobile and edge devices to high-performance solutions for professional photography. This ensures that the survey is relevant to both academic research and practical industry needs.

By applying these criteria, we have curated a set of papers that not only highlight the current state of the art in ISP using deep learning but also provide a comprehensive view of how different architectural choices and techniques can impact the performance of ISP tasks.

1.6 Comparison with Other Works

In the field of Deep Learning, a great number of surveys is already available in many different areas, such as agriculture [58], biometrics [27], cyber security [7], regularization [98], autonomous driving [33], medical imaging [71], and also on more technical areas, such as on CNN [69]. But for more recent fields, such as using **Deep Learning to replace ISPs**, it might still be hard to find this kind of gathered information. There are some surveys about individual steps of an ISP, such as demosaicing [68] and denoising [25], but not other works about end-to-end ISP via Deep Learning. As far as we could search, this is the first survey targeting the entire ISP using Deep Learning techniques. This paper summarizes many of these approaches, bringing some of the best ANNs to replace entire (or part of) ISP pipeline.

1.7 Scope of This Work

For this study, the articles were studied according to two main points:

- **Novelty:** to introduce the most recent and significant works comprising strategies for replacing parts or the entire ISP pipeline through deep learning approaches; and
- **Recently developed:** all studies considered were published between the years 2019 and 2024, making this study very up-to-date.

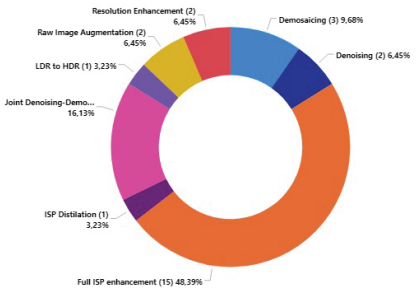

Fig. 1. Reviewed ISP tasks distribution.

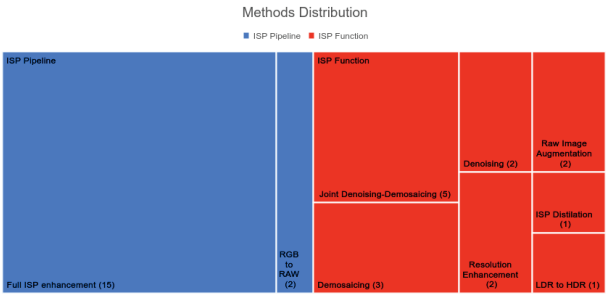

Fig. 2. Mapped ISP tasks.

The reviewed papers did not focus on the same ISP tasks and improvements. Figure 1 illustrates the distribution of ISP functions analyzed across the reviewed articles, while Table 2 lists all methods discussed in this study.

Around 30% of the papers proposed an entire ISP pipeline framework using an end-to-end Deep Learning approach. Others developed deep learning solutions for specific stages of an ISP pipeline, such as denoising tasks, joint denoising-demosaicing tasks, resolution enhancement tasks, among others. Some of them also proposed extra and distinct ISP deep learning techniques, like RAW data augmentation and RAW data generation from RGB images by using inversed ISP procedure.

Figure 2 shows how we mapped those studied ISP tasks. We labeled them into two groups: ISP function, when the proposed solution strikes specifics ISP operations, and ISP pipeline, when the proposed solution settles a RAW to RGB or RGB to RAW operation and an entire ISP procedure.

1.8 Work Structure

The rest of this work is structure as follows: Section 2 contains the resumes for the works collected through this research. Section 3 covers the results of the collected works. We provide a discussion about the methods and results in Section 4, and the Section 5 concludes this work. Along with the mentioned section, in the appendix we provide more details about how ISP works, the methodology, and more tables with comparison and details about the methods analyzed in this research.

2 Known Approaches

In this section, we review the applications of deep learning-based methods to substitute intermediary steps or the full pipeline of an ISP solution. It covers operations such as demosaicing, denoising, white balance, tone mapping, and super-resolution. Also, we briefly summarize some works that use similar approaches to increase the performance in other computer vision tasks. After introducing a set of works that address a similar problem, we stress their similarities and differences.

2.1 Denoising

Denoising is a common problem in image formation, and it has been addressed through DL-based strategies. Such strategies diverge not only on the architecture proposed, but also on the datasets used and even in the underlying assumption of the noise model.

[130] proposes the CycleISP, a framework that models camera imaging pipeline and produces realistic image pairs for denoising both in RAW and sRGB spaces. The authors trained a new image denoising network on synthetic data and achieved state-of-the-art performance on real camera benchmark datasets. CycleISP is a compound of two stages. First, the framework models the camera ISP in forward and reverse directions. Second, it synthesizes realistic noise datasets for the RAW

and sRGB images denoising tasks. The CycleISP model introduces the RGB2RAW network branch, the RAW2RGB network branch, and an auxiliary color correction network branch. The RGB2RAW and RAW2RGB modules are trained independently, followed by a joint fine-tuning.

The RGB2RAW branch converts sRGB images to RAW data without requiring any camera parameters. This module network is composed of convolutional layers, proposed recursive residual groups, dual attention blocks, and a final Bayer sampling function, generating a mosaiced RAW output. The RAW2RGB network maps clean RAW images to clean sRGB images. First, the noise injection module is set as 'OFF', followed by a 2×2 block packaging into four channels (RGGB) and an image resolution reduction block. Additionally, the authors proposed a color correction branch to the RAW2RGB network, which receives an sRGB image input and generates a color-encoded deep feature tensor. Furthermore, there is a Joint Fine-Tuning in which the RGB2RAW's output becomes the RAW2RGB's input. In this case, RGB2RAW branch receives gradients from both sub-losses, reconstructing the final sRGB image. To synthesize realistic noise image pairs for denoising in RAW space, the noise injection module is turned 'ON' and includes shot and read noise of different levels to the RGB2RAW's output. After this, CycleISP can generate a clean and its noisy image pair from any sRGB image.

The main model is trained using MIT-Adobe FiveK dataset [11]. Afterwards, a fine-tuning is done over the **Smartphone Image Denoising Dataset (SIDD)** [2]. They evaluated CycleISP performance with state-of-the-art RAW and sRGB denoising methods, using the DND [112] and SIDD [2] benchmarks. Besides that, in contrast to the other evaluated models, the proposed method provided clean and artifact-free results, also preserving image details.

Liu et al. [72] present two new techniques for DNN-based RAW image denoising. The first is a **Bayer pattern unification (BayerUnify)** method, trained to be able to handle different CFA patterns. The second method, **Bayer preserving augmentation (BayerAug)**, allows proper RAW images augmentation. Further details on BayerAug are given in Section 2.3. Associating these two techniques with a modified U-Net, the proposed method achieved a SOTA PSNR of 52.11 dB and an SSIM of 0.9969 in NTIRE 2019 Real Image Denoising Challenge.

BayerUnify consists of two stages. In the training phase, the study unified RAW data with different Bayer patterns via cropping. The technique maps the BGGR Bayer format, for example, within the other formats (RGGB, GRBG, and GBRG) and crops the selected area, converting any Bayer pattern to BGGR (or any other chosen pattern). In the testing phase, as the image pixels need to be processed, the technique unified the Bayer patterns via padding. Subsequently, there is the network denoising and the extra pixels removal, disunifying the output images and reversing the pattern conversion. The study evaluated the proposed method on SIDD [2]. The network was trained with L1 loss, AdamW [74] optimizer, and 200,000 iterations.

The two methods approach the problem in very distinct ways. BayerUnify [72] focus on bayer image denoising with a pixel-wise correspondence between input and ground-truth, giving special care to data augmentation so as to produce a unified model for all CFA patterns. CycleISP [130] argues that a deep-learning denoiser should not be trained on a dataset using a simplistic additive white Gaussian noise. A comparative summary between the training schemes is available in Table 3.

2.2 Demosaicing

Demosaicing is the process of interpolating pixel values in a single sensor that are exposed to different wavelengths. Such wavelength selection is normally achieved through a CFA. As most consumer cameras rely on CFA's for generating color images, demosaicing is almost omnipresent when it comes to ISP for consumer goods. Below is detailed different DL-based studies for performing this interpolation.

The interpolation process corresponding to demosaicing is intimately related to noise formation and reduction. For this reason, several classic and DL-based methods may combine both steps into a single one. We now present recent DL-based approaches for performing both steps at once. This problem is often referred to as **Joint Demosaicing and Denoising (JDD)**.

DRDN [83] proposes a convolutional neural network to color filter array demosaicing. Using a mosaiced image as input, the proposed model is trained in an end-to-end manner to generate demosaiced images outputs. Compared to other conventional convolutional neural network-based demosaicing models, the proposed model requires less computational complexity, because it does not require the initial interpolation step for mosaiced input images. It also solved the vanishing-gradient problem experienced by combining residual [56] and densely connected layers [29]. Moreover, the proposed model applied block-wise convolutional layers to consider local features and a sub-pixel interpolation layer, generating demosaiced output images more efficiently and accurately.

In [89], authors discussed the challenges of implementing deep learning-based demosaicing algorithms on edge the devices, especially on low-end ones. They provided an extensive search of deep neural network architectures, obtaining a Pareto front of **Color Peak Signal to Noise Ratio (CPSNR)** as the loss versus the number of parameters as the model complexity. The article proposes a classification of demosaicing methods in six groups: Edge-sensitive methods, Directional interpolation and decision methods, Frequency domain approaches, Wavelet-based methods, Statistical reconstruction techniques, and Deep learning-based methods [79]. Likewise, authors reviewed other relevant demosaicing aspects, like the unwanted presence of image artifacts, and performance evaluation methods.

The study comes up with an exhaustive search of architectures, based on discrete and well-constructed hyper-parameters, like the number of filters and blocks, the skip connections length, and the use of depthwise separable convolutions. It presented a proper methodology and mathematical theory about the neural architecture search and the Pareto building. The highlights were five brand-new theorems in respect to the neural architecture search convergence. Lastly, the designed space with a simple exhaustive search outperformed the state-of-the-art works and brought a range of loss versus complexity for edge implementation with varying resource constraints, overcoming drawbacks related to the number of evaluations and the search algorithm complexity.

This study proposed a **duplex pyramid network** (**DPN** [62]), an efficient deep neural network architecture for Quad Bayer CFA demosaicing adopted in submicron sensors.The proposed architecture consisted of two connected feature map pyramids. The architecture follows an encode-decode scheme, combining it with dense skip connections (U-Net [93]) and residual learning (ResNet [56]). DPN also implemented a Linear Feature Map Growth, reducing parameter counts when compared to the traditional exponential method, making it more suitable for mobile applications with memory constraints.

The results are compared against conventional ISP algorithm implemented in a Samsung mobile phone, observing improvement in sharpness, color moiré, edges, texture preservation, and visual artifacts reduction. DPN achieved better CPSNR values when compared with other Deep Learning based methods at that frame reference. As a limitation in the proposed network architecture, the input image width and height must be a multiple of $2^{(L+1)}$, where "L" is the resolution level.

In this paper [1], the authors proposed a deep network to work with joint demosaicing and denoising in Quad Bayer CFA and Bayer CFA patterns. The proposed network uses attention mechanisms and is oriented by an objective function that includes a novel perceptual loss function, aiming more subjective quality on a pixel-bin image sensor. This network, defined as a **pixel-bin image processing network (PIPNet)**, uses UNet as a framework and traverses different feature

depths through downscaling and upscaling operations. The authors also extended the method to reconstruct and enhance perceptual images captured with a smartphone camera.

The results were validated on the MSR demosaicing dataset [60], BSD100 [77], McMaster [134], Urban 100 [42], Kodak [28], and WED [76] datasets and compared with Deepjoint[1] [30], Kokkinos [65], Dong [24], DeepISP [102], and DPN [62] methods at three different noise levels (5, 15, 25). In all comparisons, PIPNet performed better over the PSNR, SSIM, and DeltaE2000 metrics. However, the network was tested only with data collected by traditional Bayer sensors, which may hinder network performance in other scenarios.

In this paper [108], authors proposes a method based learning the selection of the most suitable patches from an image for the training step without. To do this, the method, called PatchNet, assigns a weight to each patch that defines whether it will be used or ignored during training. This method is a feed-forward network with multiple stages where each stage is composed by several convolutions blocks and a down-sampling operator. Then gradually the stages transform the image into a set of trainability scalars that are finally binarized to obtain the network output.

In addition to PatchNet, authors also propose RestoreNet, an architecture that applies the structural knowledge extracted from PatchNet and is responsible for restoring the original image.

The results were validated on Vimeo-90k [111], MIT Moire [30], and Urban 100 [42] datasets and compared with the Kokkinos [65], SGNet [73], CDM [127], and DeepJDD[1] [30] methods. The methods were compared at three different noise levels (5db, 15db, 25db). In all comparisons, the proposed method achieved better PSRN values in the JDD task.

Ablation studies analyzed the effects of different patch sizes when PatchNet is evaluated on Demosaicing and shown that performance improves as patch size increases. The study continued with experiments of PatchNet on JDD and compared it with the methods mentioned above. Only the PSNR metric was used for comparison.

Guo and Zhang [105] studied a CNN-based Joint Denoising and Demosaicing method for real-world burst images. For this task, since the green channel has twice the sampling rate and better quality than the red and blue channels in CFA RAW data, the authors proposed a green channel prior neural network - the GCP Net. This model extracted the GCP features from green channels to conduct the deep feature modeling, upsampling the image and evaluating the frames offset, relieving the noise impact. The given work also sought out realistic noise models [6, 9], and a set of burst images instead of a single CFA image.

The GCP-Net structure is composed of two branches - a GCP branch and a reconstruction branch. By using several convolutional and Leaky ReLu blocks [126], the GCP branch extracts the green features from the noisy green channels concatenation and their noise level map.

The reconstruction branch estimates the clean full-color image. The branch consists of three blocks - the intra-frame module (IntraF), the inter-frame module (InterF), and the merge module - and utilizes the burst images, the noise maps, and the GCP features as the guided information. The IntraF block models the deep features of each frame and guides the feature extraction using the GCP features. The InterF uses a deformable convolution [20] in the feature domain to make up for the shift between frames. A pyramidal processing is applied to handle possible large motions, just like EDVR [125] and RViDeNet [43]. Furthermore, InterF includes an LSTM regularization in the offset estimation, providing the temporal constraint. The merge module provides adaptive upsampling for the full-resolution image reconstruction.

For the comparison experiments, the authors tested the proposed model on synthetic data and real-world data. In both scenarios, the GCP-Net achieved superior quantitative and qualitative

[1]Method from paper Deep Joint Demosaicing and Denoising [30], called by [1, 73] as Deepjoint, by [88] as DemosaicNet, and by [108] as DeepJDD.

performance to other state-of-the-art Joint Denoising-Demosaicing algorithms, such as FlexISP [38] and ADMM [109].

Many methods have joined highly correlated tasks and have success, decreasing the accumulated error in the several process units in the ISP pipeline. Thus, the SGNet [73] joins demosaicing and denoising in a unique network.

As the correction of high-frequency regions in images is more complicated, the authors propose extracting a density map representing the frequency of areas of the picture. This density map can help the network know each region's difficulty level and adapt better than other models in areas with high frequency. Furthermore, half of the Bayer pattern comprises green pixels; subsequently, it is easier to recover the missing pixels from this channel. For this reason, the network has a branch to reconstruct the green channel, where it consequently helps reconstruct other channels. Besides, SGNet uses the **Residual-in-Residual Dense Block (RRDB)** to feature extraction in both branches. Additionally, this network was trained in a set of loss functions, which consider the reconstruction fidelity of the green channel, full image, the objects, textures edges, and noise removal.

SGNet outperforms state-of-the-art methods in terms of PSNR and SSIM in datasets aimed at the super-resolution, denoising, and demosaicing tasks. Furthermore, compared with the ADMN [109], CDM [127], Kokkinos [65], Deepjoint[1] [30], and FlexISP [38], the SGNet can remove moiré artifacts more effectively and give images with more definition in high-frequency areas than ADMN and Deepjoint.

When it comes to demosaicing, PIPNet, DRDN, and DPN propose novel NN architectures to be trained on labeled data. They may differ in internal layers and training scheme, but their fundamental approach is the same: create an architecture, follow an existing training scheme, and choose a set of hyper-parameters. GCP-Net strategy is also similar, but it leverages multiple raw frames for achieving denoising as well.

On the other hand, DeepEdge approached the problem differently while focusing on resource-constrained applications. For doing so, it proposed a loss versus complexity Pareto frontier to choose a base architecture based on given edge-device constrains.

A similarity among most of these studies is the usage of sRGB images as output ground truth samples for training and validation. Since sRGB images are obtained from standard cameras, it could be argued that such models learn not only demosaicing and eventually denoising, but the full imaging pipeline, including steps such as white balance. If demosaicing was to be learned stand-alone, the model should be trained against images with minimal manipulation. Using a space such as the linear RGB space would be advisable, which is strongly dependent on the camera hardware. Among the studies evaluated, only GCP-Net and PIPNet use linear RGB as expected training output.

We also notice that PatchNet takes another complementary approach. By learning to select image patches for training a JDD network, it actually introduces a learnable augmentation approach tailored for images.

Concerning the denoising portion of JDD studies, it is worthy noticing that they differ in the approach for learning to remove noise. For instance, GCP-Net combines multi-frame, a classic ISP approach, with DL. SGNet introduces a custom loss function for evaluating image noise, biasing the optimization process towards smoother images. PIPNet adds noise to the inputs, considering that the dataset ground truth is clean.

2.3 Raw Image Augmentation

Data augmentation is a standard practice for training DL models. Nevertheless, dealing with raw Bayer images imposes a few challenges which are not available when handling RGB images. While

manipulating Bayer images with DL methods, researchers have proposed novel ways of augmenting datasets which are specially tailored for raw images.

One particularity of Bayer images is that its color channels are spatially intercalated, whereas in the RGB case they are overlapping. This fact makes theoretically wrong traditional augmentation procedures of image augmentation, such as image cropping, rotating, and flipping.

To address this issue, Liu et al. [72] have systematized the right procedure to flip, crop, and rotate images without distorting the CFA pattern. This set of rules, named BayerAug, is shown to improve PSRN in up to 0.3 dB when compared with naive augmentation of interleaved images. Such procedures are based on dropping single rows and columns when performing image transformations, such that the CFA pattern is kept constant.

PatchNet [108] is a CNN model for choosing patches (i.e., crops) of images that are well suited for a specific DL training process. It is proposed as an opposition to random crops, which may oversample solid patches, adding little to the training process. Authors have shown that this simple approach improved model PSNR in up to 0.27dB.

It is worth mentioning that the augmentation method proposed is not exclusive to Bayer images, but it could be used in other similar image processing tasks with minor adaptations.

The two methods presented complement each other, since both were shown to improve performance while modifying different aspects of the training procedure. Moreover, since both augmentation methods impact only training, they do not increase inference run-time nor model size. Apart from that, BayerAug is the simplest one, since it is a deterministic method, whereas PatchNet is actually a CNN that has to be trained alongside the neural network that performs the target task.

2.4 RGB to raw

Training DL methods on raw images can face some challenges, specially because of the low availability of raw images when compared to sRGB ones. Hence, one possible approach for developing new models is to first generate raw images from their RGB counterparts, and then train the model on this synthesized data. For achieving this inverse ISP, a couple of works have proposed DL models which are explored below.

The authors have proposed primarily a DL denoising framework whose training relies on inverting the processing pipeline to account for non-linear transformations of noise applied by the ISP [130]. While idealizing this framework, it has been proposed a novel method for achieving a DL inverse ISP. This method is based on two separate networks RGB2RAW and RAW2RGB, which are trained independently on image pairs. Notice that the implicit ISP to be learned corresponds to the camera that captured such shots. Afterwards, the two architectures are concatenated, such that the compound model should act as an identity operator. This new model is trained using the input as its expected output, fine tuning the weights of the two parts.

In InvISP [124], the authors have designed an intrinsically inversible NN by using mostly bijective functions. Even some lossy operations such as quantization and JPEG compression are approximated by inversible ones, so as to keep the characteristic property of the architecture. The model is trained in both directions (i.e., RGB to RAW and RAW to RGB).

CycleISP uses two different architectures RGB2RAW and RAW2RGB for converting images, while InvISP proposes a single inversible architecture based on bijective functions. In spite of this difference, both works train their models in both directions. The comparison between training parameters is available in Table 5.

2.5 Resolution Enhancement

A common digital ISP use-case takes place when the user wants to zoom in regions further than what the image sensor is able to capture. Traditionally, this would be solved with some sort of

interpolation (e.g., nearest-neighbor or bi-linear), even if doing so tends to over-smooth edges or introduce other sorts of artifacts. The deep learning methods presented here promise to enhance resolution (i.e., apparent pixel count) preserving edges and details, while maintaining the same hardware setup.

Details of SGNet were introduced in Section 2.5, since it is primarily a JDD method. Nevertheless, the authors have demonstrated that by training this architecture with Bayer images smaller than RGB output super-resolution can be achieved. This experiment was done using the Pixelshift200 dataset. This publication does not mention any adaption from its original form for handling the super-resolution case.

As detailed in Section 2.6, LiteISPNet is an end-to-end DL ISP. It has as its main feature robustness to misalignment between raw input and RGB output, a critical issue for super-resolution model training. If LiteISPNet is tailored for ISP, its framework is adapted for super-resolution by substituting the architecture backbone by SRResNet. This specific problem is evaluated in the SR-RAW dataset [137].

Resolution enhancement is closely related to demosaicing, since both problems are based on filling data gaps from information that is sub-sampled in space. This characteristic explains why both SGNet and LiteISPNet, architectures designed originally for JDD and Full ISP, are extended to $\times 2$ and $\times 4$ super-resolution problems with minor or no modifications. In fact, the proposed models perform resolution enhancement alongside other computational intense tasks (e.g., denoising or color correction), solving several problems at once.

2.6 Full ISP Enhancement

The approaches presented so far concentrate on solving some well established steps of image reconstruction targeted for the human visual system, one at a time. Instead of performing these steps separately, it is arguable that the whole processing pipeline could be compressed into a single DL model. Those defending this idea believe that DL models could learn to perform all operations proposed so far and adjust themselves taking into account scene specific features hard to grasp in hand-designed models. Based on this principle, several works have been published and are described bellow.

CameraNet [70] proposes an effective and general framework for a deep learning-based ISP pipeline, with two stages of CNN stacked. The motivation for this is that some subtasks of the ISP pipeline have poor correlation, then the subtasks from the ISP pipeline were divided into two stages: the first stage is the restoration stage with tasks like demosaicing, denoising, and white balance, and the enhancement stage as second stage performing tasks like exposure adjustment, tone mapping, color enhancement, and contrast adjustment. Besides, two ground truths were created for each image in the datasets HDR+ [37] and FiveK [11] using Adobe Camera Raw[2] and Adobe Lightroom.[3] Each ground truth was used to train a different stage. Preceding the DL stages, the input image is pre-processed with bad pixels removal, initial demosaicing with interpolation, and conversion from RGB to CIE XYZ due to its relation to human perception. Moreover, the U-Net is the base model for these two stages, because of the multi-scale extraction features. Some changes were made, like the addition of a fully connected layer in the lowest level of the network and the use of different processing blocks in each stage. While the restoration stage uses plain convolutional blocks, the enhancement stage uses residual connections to details improvement. Furthermore, in the experiments, the CameraNet generated images with less noise, artifacts, better color mapping, and higher qualitative scores than DeepISP [102] Network in the HDR+

[2] https://helpx.adobe.com/br/camera-raw/using/supported-cameras.html
[3] https://www.adobe.com/lightroom

and, mainly, SID [13] dataset. The explanation for this difference can be the high level of noise in SIDD and the separation of weakly related subtasks in two stages in the CameraNet. In the FiveK dataset, both methods achieve comparable results in SSIM, but CameraNet obtains superior results in PSNR and Color Error.

HighEr-Resolution Network (HERNet) [78] is a network that can learn local and global information about high-resolution image patches without excessive consumption of GPU memory. This network has two paths for local and global feature extraction and the introduction of the Pyramid Full-Image Encoder [78] that performs a regularization of the output image and helps to reduce the number of artifacts. Besides, this work proposes training the model with progressively growing the resolution of inputs, which results in performance stability and short training time.

The local information path consists of **Multi-Scale Residual Blocks (MSRBs)** [67] modules for feature extraction. These modified RIR modules were designed to reduce GPU memory usage, particularly for high-resolution images. To achieve this, the Channel Attention Units were removed, and only the remaining components were stacked. The authors trained and validated the model using the Zurich Dataset [49], employing only the L1 loss during training. However, the use of L1 loss tends to favor blurry outputs, especially in datasets with misaligned images.

The progressive training was used to train the network, where the input image resolution increased during the training, keeping the same architecture of networks all the time. As a result, this process can make the network converges more quickly. Besides, HERNet won second place in track 1 of fidelity and first place in track 2 of perceptual in AIM 2019 RAW to RGB Mapping Challenge.

As the first CNN proposed to substitute the entire ISP pipeline, the Deep Camera [91] is a small network with four parallel paths: a main path and other three short paths with a convolutional layer in your middle. The explanation for this is the model is very small compared with the ResNet, then the network does not generalize well with the copy of the input to the output of a block. Furthermore, authors created an inverse ISP to recreate RAW images from a large dataset [18] with 11,000 images and several types of scenes and illuminants, where the training and experiment stages used the resulted dataset.

The CNN model outperformed traditional methods in white balance and image reconstruction tasks, delivering many better images. However, in some images with many different colors, other algorithms are better. Besides, the Deep Camera can do defective pixel correction and be used in other color filter mosaics like the X-Trans color filter by Fujifilm.

Wu et al. [122] proposes a particular ISP method for computer vision applications. VisionISP reduces the data transmission needs without relevant information loss, optimizing the subsequent computer vision system performance. The framework consists of three processing blocks. The first block, the Vision Denoiser, reduces the input signal noise and modifies the tuning targets on an existing ISP. This study adopted the technique presented by Nishimura [55] to optimize the denoising parameters, constructing the denoiser for the computer vision task, not for image quality. This paper also highlights that demosaicing can be skipped and the color filter array image used, since it did not improve computer vision task performance. The second block, the **Vision Local Tone Mapping (VLTM)**, reduced the bit-depth, achieving similar accuracy with fewer bits per pixel. VLTM used a global non-linear transformation followed by a local detail boosting operator. Lastly, the **Trainable Vision Scaler block (TVS)** is a generic neural network that processes and downscales the input for a following computer vision engine.

VisionISP was trained and evaluated with the KITTI 2D object detection dataset [3], an autonomous driving benchmark dataset. The study measured the influence of each VisionISP block

in the **mean average precision(mAP)**. As a computer vision task sample in the experiments, the authors used the SqueezeDet [8] framework and its original code.

In this paper [34], authors combined convolutional neural networks with traditional algorithms to reverse the order of the traditional CFA pipeline (demosaicing and denoising). The method, which we called here RLDD, uses two stages for demosaicing-denoising. The first stage performs the demosaicing by composing the **gradient-based threshold-free (GBTF)** method [86] and a convolutional neural network to overcome the reduction of image resolution in the subsampling operation. The second stage performs the denoising using another convolutional neural network aiming to deal with residual noise. The properties of the residual noise were altered due to complex interpolation, and the convolutional neural network aimed to remove it without losing the details of an image.

The results were validated on the Kodak [28], McMaster [134], and Urban 100 [42] datasets and have shown that this model outperforms state-of-the-art demosaicing and joint demosaicing and denoising algorithms with higher PSNR and SSIM values on all datasets. The results of the visual comparison between the methods confirmed the quantitative values achieved.

PyNET [49] is a novel pyramidal CNN architecture designed to replace the entire ISP pipeline in smartphones. The proposed method has an inverted pyramidal shape and was composed of five different levels trained from bottom to top where each level is trained sequentially. Then, the pre-trained output is used in the above level training stage. The convolutional filters size in this method varies from 3×3 at level five up to 9×9 at level one. Therefore, lower levels learn global image manipulation while higher levels learn to reconstruct the final image recovering the missing details at lower levels. The network is trained using three different loss functions combinations. The lowest levels are trained with **mean squared error (MSE)** to learn global color and brightness correction. Intermediate levels are trained by a combination of MSE and VGG-based [54] to refine the color and shape of objects. Finally, the top level is trained with MSE, VGG, and SSIM loss [119] and performs corrections in the local color processing, noise removal, texture enhancement, and so on.

Additionally, authors present the Zurich RAW to RGB dataset, composed of 20,000 RAW-RGB image pairs where the RAW images are captured using Huawei P20 smartphone and the RGB images are captured using a professional high-end Canon 5D Mark IV camera.

To evaluate the method, three experiments were conducted. The first compared PyNET with the methods SPADE [85], DPED [45], U-Net [93], Pix2Pix [51], SRGAN [66], VDSR [63], and SRCNN [23]. In this comparison, PyNET outperformed all other methods in the PSNR and MS-SSIM metric values. The second experiment measures the quality of the generated images using the Amazon Mechanical Turk[4] platform, compared the images of PyNET, Visualized RAW, and Huawei P20 ISP with the images produced by the Canon 5D Mark IV DSLR camera. In this comparison, the image produced by PyNET reached the better MOS result in comparison to the target DSLR camera.

This work [21] proposes a method for enhancing smartphone generated images by replacing the base ISP by a U-Net [93] resembled CNN, called AWNet.

The network is divided into two branches, each using different inputs and, thus, different models. The first branch, using the RAW model, receives $224 \times 244 \times 4$ RAW images and the second branch, using the demosaiced model, receives $448 \times 448 \times 3$ demosaiced images. Both branches are trained separately and the results are averaged during test.

The structure of this network, following U-Net, applies three main modules to the inputs, for each branch: global context res-dense, residual wavelet up-sampling, and residual wavelet down-sampling. The res-dense module is applied to extract the low frequency components after the

[4] https://www.mturk.com

discrete wavelet transform (DWT), which are sent to the layer below, while the down-sampling extract all the components. After the extraction, both sets of components for each layer are up-sampled and concatenated with the layer above. Finally, a pyramid pooling module is applied, generating the output for that branch.

For testing, a self-ensemble mechanism was applied, made up of eight ensemble variants. Those variants were, then, evaluated using the PSNR (dB) values, which would be used as weights to generate the final predictions of the model. The chosen PSNR were 21.36 dB for the RAW model and 21.52 dB for the demosaiced one. By applying this tuned model to two tracks of the Zurich dataset from AIM 2020 Learned Smartphone ISP Challenge [48], the study reached the 5th and 2nd positions, respectively.

Using the results as a justification for the use of the wavelet transform and the global context blocks, the researchers compared the results of AWNet against other popular network architectures, like U-Net, RCAN [138] and PyNet [49], with the use of the Zurich Dataset. By comparing their performance, the researches found that AWNet is able to outperform U-Net, RCAN and the current state of the art, PyNet.

PyNET-CA is an end-to-end mobile ISP deep learning algorithm for RAW to RGB reconstruction [61]. This network improves PyNET [49] performance by adding channel attention and subpixel reconstruction modules and decreasing training time. PyNET-CA has an invertible pyramidal structure for considering the local and global features of the image. The Basic modules of PyNET-CA are the channel attention module based on [138], the DoubleConv module, which has two operations of 2D convolution with a LeakyReLU activation, and the MultiConv channel attention module, which concatenates the features from the DoubleConv modules and a channel attention module.

The superpixel reconstruction module helps the network reconstruct the final image with quality and better computational efficiency. For this, PyNET-CA upsamples the image with the MultiConv channel attention module, followed by a 1×1 convolution layer and upsamples the features by subpixel shuffling at the final level of the model. The results were presented on the Zurich Dataset [49] where it has shown better PSNR and SSIM values when compared to PyNET [49].

Del-Net [35] is a single-stage end-to-end deep learning model that learns the entire ISP pipeline to convert RAW Bayer data to sRGB-image. This network uses a combination of Spatial and Channel Attention blocks (modified UNet) [131] and Enhancement Attention Modules blocks [4]. The Spatial and Channel Attention blocks allow the network to capture global details both spatial-wise and channel-wise, therefore helping with color enhancement. The Enhancement Attention Modules blocks help in denoising, which improves the PSNR value. The images generated by Del-Net are visually comparable to the state-of-the-art networks (PyNET [49], AWNet [21], and MW-ISPNet [48]) when considering the color enhancement, denoising, and detail retention capabilities while reducting in Mult-Adds (Number of composite multiply-accumulate operations for an image). Altogether, this makes the network ideal for smartphone deployment. It also has a competitive trade-off between accuracy metrics and complexity. The results were presented on Zurich Dataset [49] where it has shown better detail retention compared to PyNET [49], better denoising compared to MW-ISPNet ignatov2020aim, and better color enhancement compared to AWNet [21].

In this paper, the authors have proposed an ISP-Net that addresses JPEG image compression in network training [113], which we called here ICDC-Net. The fact that images can lose information in the compression process has not been addressed on previously ISP pipelines with convolutional neural networks.

To this end, the authors applied a fully convolutional **compression artifacts simulation network (CAS-Net)**. This network can add JPEG compression artifacts to an image, and it is trained by inverting the inputs and outputs needed for training compression artifact reduction networks.

In this work, the authors connected the CAS-Net to an ISP network, so the ISP network can be trained with consideration to image compression, taking compression artifacts into account. The ISP-Net used in this work was U-Net with channel attention module [114] and the architecture of CAS-Net was U-Net [93] without channel attention module.

Results are present on the Nikon D700 subset from the MIT-Adobe FiveK dataset [11]. To render the ground truth of sRGB images from RAW images, the LibRaw library was used. The sRGB images were compressed with two different QFs, 80 and 90, and each model was trained separately. The experimental results have shown that this proposed network can produce better quality images when compared to its compression agnostic version.

Achieving second place in the Mobile AI 2021 Learned Smartphone ISP Challenge [44] and first place in PSNR score, the **Channel Spatial Attention Network (CSANet)** [40] focuses on balancing computational performance and image quality, with an inference time of 90.8 ms per image. The network is composed of three main components: downscaling, cascaded processing blocks, and upscaling.

In the downscaling stage, the authors reduced computational time and the number of parameters by using a strided convolution block followed by a conventional convolution for feature extraction. Next, the core of the network consists of a **Double Attention Module (DAM)**, inspired by the **Convolutional Block Attention Module (CBAM)** [121]. The DAM includes a spatial attention module, which captures spatial dependencies within feature maps, and a channel attention module, which learns inter-channel relationships.

Finally, in the upscaling stage, a transposed convolution and depth-to-space operation are used to reconstruct the final RGB image. An important aspect of this work is its loss function, which combines Charbonnier loss [138], SSIMloss, and Perceptual loss tominimize perceptual differences between the generated image and the ground truth.

In some datasets, the RAW and RGB images are captured with different cameras. Consequently, the pairs of images have misalignment and color inconsistency, making the training process more difficult and producing blurry results. Taking it into account, Zhang et al. [139] proposed a method to train the networks with misaligned images and map RAW to RGB. The authors used the pre-trained optical flow estimation network, PWC-Net [107], to align the image pairs and designed a **global color mapping (GCM)** to match the color between the input and ‘target images to facilitate alignment. Besides, LiteISPNet is responsible for mapping the RAW-to-RGB. It simplifies MW-ISPNet [48], which proposed a U-Net based multi-level wavelet ISP network, reducing the number of RCAB in each residual group and changing the position of convolutional layer and residual group before each wavelet decomposition. These changes decreased the model size and running time by approximately 40% and 20%, respectively.

The authors tested the network in two datasets, the Zurich Dataset [49], and the SR-RAW [137], with two variants of ground truth: the original GT and align GT. In the Zurich Dataset, the LiteISP-Net was compared with three states of the art (PyNet [49], AWNet [21], and MW-ISPNet) and outperformed all metrics on the aligned GT but was a little worse than MW-ISPNet in the SSIM metric on the original GT. Moreover, the GAN version of this model obtains better perceptual results in the LPIPS metric [136]. Finally, in qualitative comparison, the network could retain more fine details than other models. With the SR-RAW dataset, the authors also compared with SR methods, as SRGAN [66], ESRGAN [118], SPSR [75], and RealSR [53]. It generates images with less noise, less blurriness, more details, and it had better scores in almost all metrics, losses only in the PSNR metric on original GT, which favors blurred images.

In conclusion, this work outperformed state-of-the-art models in ISP and SR tasks and introduced a novel method for training DNN models on misaligned datasets. This method also enabled the use of a lightweight network, as demonstrated in this study, while achieving results

comparable to or better than those of more robust models. However, the authors did not report any tests of the model on embedded systems.

Usually, the ISP pipeline is an operations sequence with three core components in a fixed order: demosaicing, denoising, and super-resolution. However, Qian et al. [88], in extensive experiments, showed that a simple reordering of the operation sequence can increase the image quality. Then, the authors created the **Trinity Enhancement Network (TENet)**, a network that reordered the operation sequence to denoising (DN), super-resolution (SR), and demosaicing (DM). The DN block is the first because the noise on a RAW image has a Gaussian-Poisson distribution [36], then is more straightforward to resolve; the RAW image noise can hinder subsequent tasks; also, this noise become complex over image processes operations. Furthermore, the DM in higher resolution images results in fewer artifacts, and super-resolution algorithms could amplify the artifacts generated by DM. Therefore, the SR was the second block in this architecture.

As the DN produces blur in the image, the authors joined the DN and SR in a unique block eliminating the accumulated error over image operations and resulting in this final pipeline: DN + SR -> DM. To effectively leverage in consideration these two stages, the loss function is composed of two losses: the $\mathcal{L}_{joint}$, which is the l_2-norm loss on the final output image, and LSR, the l_2-norm loss between the DN+SR result and the high-resolution noise-free mosaiced image of the input image. Thus, the final loss was the sum of these two losses. Besides, the authors used the **Residual in Residual Dense Block (RRDB)** [118] to construct the central part of all blocks.

They also notice that previous datasets that synthesized the DM are sub-optimal for three reasons: (1) The images used to synthesize RAW images are the result of interpolation by the camera ISP; (2) The model was trained to learn an average DM algorithm used in the camera ISP; (3) The synthesized RAW images had less information than real RAW images. For this reason the PixelShift200 Dataset [88] was created with 200 2k-resolution full color sampled images. Each pixel of images was all color information without demosaicing because of the pixel shift technique. Besides, from these high-resolution RAW images was created the low-resolution RAW images through the bicubic downsampling kernel [23], mosaic kernel [9], and addition of the Gaussian-Poisson noise model [36].

The model was compared with the ADMM [109], Condat [19], Flex-ISP [38], and DemosaicNet[1] [30] on the commonly used benchmark datasets to denoise and demosaicing tasks: Urban 100 [42], Kodak, McMaster [134], and BSD100 [77]. TENet outperforms these models in qualitative metrics, therefore generating much less moiré, color artifacts, and more fine-grained textures. The network generated clean images with accurate details validated in the datasets with the addition of Gaussian white noise, where the ADMM and DemosaicNet generate smooth results, FlexISP does not treat the noise correctly, and the ADMM generates color aliasing artifacts.

ReconfigISP [129] is a reconfigurable ISP where the architecture and parameters are adapted according to a specific task. To accomplish this, the authors have implemented several ISP modules and given a specific task, an optimal pipeline is configured by automatically adjusting hundreds of parameters. This method maintained the modularity of the steps in an image reconstruction process, where each module performs a clear role in the ISP pipeline and allows back-propagation for each module by training a differentiable proxy. The differentiable proxy aimed to imitate a non-differentiable module via a convolutional neural network, thus allowing the optimization of the module's parameters. Therefore, the ISP architecture was explored with neural architecture search, where modules receive an architecture weight and are removed if the weight is below a pre-set threshold.

To validate the effectiveness of this proposal, the authors performed experiments with image restoration and object detection with different sensors, light conditions, and efficiency constraints.

The results were validated over the SID Dataset [13] and S7 ISP Dataset [102] and showed that this network outperforms the traditional ISP pipelines achieving a higher PSNR value than Camera ISP.

In this paper [80], the authors proposed a method for enhancing an image inspired by photographers who perform image retouching based on global image adjustment curves. This method, called CURL, can be used in two different scenarios. The first is the RGB-to-RGB mapping where an input RGB image is mapped to another visually pleasing RGB image and the second scenario is the RAW-to-RGB mapping where the entire ISP pipeline is done.

This method is composed of two architectures called **Transformed Encoder-Decoder (TED)** backbone and CURL block. The TED is similar to U-Net [93] but without the skip connections, except for the level-1 skip connections which were replaced by a multi-scale neural processing block that provides enhanced images via local pixel processing to the CURL block. The CURL block is a Neural Curve Layers block that exploits the representation of the image in three color spaces (CIELab, HSV, RGB) intending to globally refine its properties through color, luminance, and saturation adjustments guided by a new multi-color space loss function. The CURL loss function aims to optimize the final image in its different properties such as chrominance, hue, luminance, and saturation.

Two experiments were done for the validation of the method, where the medium-to-medium exposure RAW to RGB mapping and the predicting the retouching of photographers for RGB to RGB mapping was evaluated. In the first, results were validated on the Samsung S7 dataset [102], and CURL scored the best PSNR and LPIPS metrics when compared to the U-Net [93] and DeepISP [102] methods, but tied with the DeepISP method on the SSIM metric. In the second, results were validated over the MIT-Adobe FiveK dataset [11] and compared with HDRNet [31], DPE [16], White-Box [41], Distort-and-Recover [84], and DeepUPE [117] methods where CURL scored better on PSNR and LPIPS metrics, but DeepUPE scored the best on SSIM.

InvISP [124] redesigns the ISP pipeline allowing the reconstruction of RAW images almost identical to camera RAW images without any memory overhead and also generates human-pleasing sRGB images like traditional ISPs. This is interesting since end users can only access processed sRGB images because RAW images are too large to store on devices. The reconstruction of RAW images in this method is done by the compression of RGB images with the inverse process. To achieve this goal, the authors designed a RAW-to-RGB and RGB-to-RAW mapping from an invertible neural network consisting of a stack of affine coupling layers and an invertible 1×1 convolution. In addition, a differentiable JPEG compression simulator was integrated into the model, allowing the reconstruction of near-perfect RAW images from JPEG images by Fourier series expansion. The network was trained bidirectionally to jointly optimize the RGB and RAW reconstruction process. Model evaluation was performed on the Canon EOS 5D subset and Nikon D700 subset from the MIT-Adobe FiveK dataset [11]. To render the ground truth of sRGB images from RAW images, the LibRAW library was used, which allows simulation of the steps of an ISP pipeline. The experiments demonstrated an improvement of PSNR over the RAW synthesizing methods UPI [9] and CycleISP [130], which implies a more accurate retrieval of RAW images. The method was compared to Invertible Grayscale [123] and U-net [13] baselines and the results showed better PSNR and SSIM values.

A comparative summary between the training schemes is available in Table 6.

2.7 Miscellaneous

Recently, other authors focused on other more specific problems but which are still worth mentioning. Hence, in this section we present recent advancements in DL-powered HDR photography and knowledge distillation from raw data.

Merging-ISP [12] is a deep neural network designed to reconstruct multiple **LDR (low dynamic range)** image layers into a single **HDR (high dynamic range)** image. The input consists of RAW images captured in either dynamic or static scenes, which are processed by the network to map and merge the LDR layers into a single HDR output.

Before merging, the network applies a DnCNN [133] structure with the following se quence of filters: one 5×5 filter with 64 layers, followed by two 5×5 filters with 64 layers each, and finally three 1×1 filters with sigmoid activation. This process reduces the data volume without requiring additional training. The merging process combines the LDR layers into an HDR image using four convolutional layers, with receptive fields decreasing from 7×7 with 100 filters in the first layer to 1×1 with three filters in the final layer. Notably, no optical flow was required for the input data.

To train the network, synthetic and real datasets based on Kalantari et al. [57] datasets were used. The data contained dynamic and statics scenes, as was stated before. Secondly, rotation techniques were used to increase the dataset, contributing to extracting 210,000 non-overlapping patches of size 50×50 pixels using a stride of 50. Besides, they perform training over 70 epochs with a constant learning rate of $10e^{-4}$ and batches of size 32. During each epoch, all batches are randomly shuffled. In comparison to other merging-ISP methods, this one approach obtained the best result in PSNR, SSIM, and HDR-VDP-2 parameters.

In ISP Distillation [100], the authors proposed a model for image classification with RAW images using an sRGB image classification model and Knowledge Distillation [39] of an ISP pipeline to reduce the compute cost of the traditional ISP. Traditional ISP pipelines focus on human vision, while this paper provided a solution for machine vision only. A dataset of RAW and RGB pairs was used to overcome the performance drop that occurs when data was trained directly on RAW images. This dataset is used to pre-train a model that was subsequently distilled to another model responsible for treating directly the RAW data.

To validate the proposal, two cases were tested. The first was by discarding denoising and demosaicing pre-processing on a model pre-trained on the ImageNet dataset [96]. The second was to discard the entire ISP pipeline in a model pre-trained on the HDR+ dataset [37]. ResNet18 [56] and MobileNetV2 [97] were used for validation. Both experiments demonstrated good performance when evaluated on top-1 and top-5 metrics. Therefore, ISP Distillation is a step towards achieving similar classification performance on RAW images when compared to RGB.

PyNet-V2 Mobile [46] presents a novel approach to on-device photo processing on mobile devices, leveraging the efficacy of neural networks to address the performance limitations of traditional ISPs. This work introduces a custom-designed network architecture based on PyNET [61], PyNET-V2 Mobile, specifically optimized for the resource constraints inherent in mobile hardware. Consequently, PyNet-V2 Mobile enables the real-time conversion of RAW sensor data into high-quality RGB images directly on mobile devices, significantly surpassing the speed and quality offered by conventional ISPs.

The core technical contributions of PyNet-V2 Mobile lie in its innovative architecture and efficient processing techniques. The network employs an encoder-decoder structure, effectively extracting salient features from the RAW image and subsequently reconstructing the final RGB photo. Channel attention mechanisms are strategically incorporated to dynamically focus the network's attention on critical image features, thereby enhancing detail preservation and noise reduction. Furthermore, PyNET-V2 Mobile utilizes lightweight convolutional layers and other mobile-friendly optimizations to minimize computational requirements, ensuring efficient execution on mobile GPUs.

This work demonstrates the substantial advantages over traditional ISPs. The network achieves remarkably fast processing times, converting 12MP RAW images into vibrant RGB outputs in under 1.5 seconds on modern mobile hardware. In comparison, conventional ISPs often require several

seconds, potentially leading to missed photo opportunities. Moreover, PyNet-V2 Mobile delivers superior image quality, significantly reducing noise, sharpening details, and ensuring accurate color reproduction, even in challenging lighting conditions. Additionally, the power efficiency of PyNET-V2 Mobile surpasses that of traditional ISPs, translating to longer battery life and uninterrupted photo capture experiences.

RGBW (Red-Green-Blue-White) is a recently developed CFA pattern with better capacities for low-light images. To deal with the RAW-to-RGB conversion, OTST [26], a two-phase framework dedicated to jointly denoising and remosaicing RGBW images, transforms them into RGB images.

Phase one works with the dynamic duo of **Omni-dimensional Dynamic Convolution (ODC)** and Half-Shuffle Transformers. ODC modules adapt their tools (convolution kernels) to match the unique features of each image element, erasing noise while preserving the intricate details. Meanwhile, Half-Shuffle Transformers seamlessly exchange information across color channels and spatial dimensions, ensuring the denoising is thorough. Phase two uses the **Spatial Compressive Transformer (SCT)**. This agile transformer deals across the image, capturing both the nearby pixels and the entire scene. With its spatially-aware attention mechanisms, SCT paints a complete picture, reconstructing missing color channels, and transforming the fragmented RGBW mosaic into a Bayer. As far as we searched, this is the only work using Transformers that can be found.

MicroISP [47] is an ISP network designed specifically for mobile and edge devices, addressing four key challenges associated with running deep learning models in such environments: (i) limited RAM; (ii) a small set of commonly available machine learning operations on the device; (iii) restricted computational power; and iv. limited storage for large deep learning models.

To tackle these challenges, the authors proposed amodelwith three independent branches that process the RGB channels, decomposed from the RAW input, separately. Each branch includes 3×3 convolutions, a novel attention block, skip connections, and PReLU and Tanh activation functions. The outputs of these branches are then concatenated to produce the final image.

By employing a simple architecture and lightweight operations, MicroISP can process images up to 32MP directly on edge hardware, requiring only 158KB of storage when exported in the TFLite FP32 format. The model was evaluated on the Fujifilm UltraISP dataset, where it achieved state-of-the-art performance.

3 Results

In this section, we analyzed the works on a quantitative comparison and indicated possible reasons for a method to present better than others. We divided this section into four subsections, where each one discussed results obtained in a dataset. In the first, we discussed the Zurich RAW to RGB dataset, which contains pairs of images from different cameras and misalignment between the pairs. In the second, we discussed the Urban 100 dataset, which is composed of 100 High-Resolution images with real-world urban scenarios and structures. Then, in sequence, we discussed the works in the McMaster dataset, with 18 images captured by Kodak film. Finally, in the fourth section, we discussed the Kodak dataset with 24 photographic quality images of size 768×512 or 512×768, generally used in compression tests and to validate methods for tasks like demosaicking and denoising, and so on.

3.1 Zurich RAW to RGB

HERNet and LiteISPNet achieve better results in Table 1, but this is due to the use of, respectively, a different data distribution and RAWimages aligned with the ground truth. Among the remaining methods, AWNet outperformed others in terms of PSNR and SSIM scores. This superior

performance can be attributed to its use of a self-ensemble strategy, as the scores for the RAWand demosaiced versions are significantly lower compared to the ensembled version.

Additionally, the RAWversion of AWNet achieved slightly higher results than LiteISPNet and PyNet-CA, likely due to the inclusion of wavelet transforms and global context blocks. Furthermore, LiteISPNet demonstrated impressive results by employing an additional method to align images, which enhanced the network.s training process.

3.2 Urban 100

The Urban100 dataset is widely used for various image restoration tasks. To adapt it to specific tasks, modifications are often applied by different works. As shown in Table 1, RLDD achieved the highest scores compared to other methods; however, this was done with the image borders removed during validation, which can artificially boost the scores.

Additionally, PIPNet demonstrated results comparable to DPN, despite being validated with noisy data. The promising performance of PIPNet on this dataset may be attributed to the introduction of attention mechanisms, which establish strong correlations across depth and spatial dimensions.

3.3 McMaster

The DRDN+ and DRDN methods showed better results in the PSNR metric in Table 1. These methods are CNN-based models and focus on demosaicking. The DPN method, which also focused on demosaicing, showed better results in the SSIM metric and better artifacts reduction in its qualitative evaluation. The TENet network had the lowest performance in the PSNR and SSIM metrics, however, it is worth noting that the goal of this network is to make an entire ISP pipeline enhancement, instead of only the demosaicing task.

3.4 Kodak

Table 1 shows the reviewed paper results in the Kodak dataset. RLDD [34] achieved the best PSNR metric performance, whereas DRDN+ [83] had the best SSIM metric performance. The RLDD framework combines Denoising and Demosaicing techniques, delivering proper quantitative and qualitative results. It is important to reinforce that RLDD authors removed 10 pixels from the Kodak image's borders to calculate the PSNR. TENet and PIPNet introduced artificial noise models into the dataset for deeper denoising study. DRDN stands out in terms of efficiency and accuracy, in large part because of its block-wise convolutional neural networks which consider local features and a sub-pixel interpolation layer.

4 Discussion

This section provides points of improvement in these works about their methodology and evaluation. It addresses the negative and positive points of the works described above.

4.1 State of the Art in End-to-End ISP using Deep Learning

In this section, we focus on the state-of-the-art methods for end-to-end ISP using Deep Learning techniques. End-to-end approaches have gained traction in recent years due to their ability to streamline the ISP pipeline, learning to optimize the entire process directly from raw data to the final image output.

4.1.1 Neural Architectures for End-to-End ISP. The majority of end-to-end ISP methods rely heavily on CNNs, which have proven effective for a variety of tasks such as denoising, demosaicing, and super-resolution. These methods benefit from CNNs ability to capture local spatial hierarchies,

Table 1. Dataset Performances

Datasets	Architecture	PSNR	SSIM
Zurich Raw2RGB [49]	Del-Net [35]	21.46	0.745
	PyNet [49]	21.19	0.746
	AWNet (Ensemble) [21]	21.86	0.781
	AWNet (Demosaiced) [21]	21.38	0.745
	AWNet (RAW) [21]	21.58	0.749
	PyNet-CA [61]	21.50	0.743
	LiteISPNet [139]	21.55	0.748
	LiteISPNet [139]	**23.76**	**0.873**
	HERNet [78]	22.59	0.81
MIT-Adobe FiveK Nikon D700 subset [11]	InvISP [124]	37.47	0.9473
MIT-Adobe FiveK Canon EOS 5D subset [11]	InvISP [124]	33.61	0.9007
Kodak Photo CD [28]	DRDN [83]	42.43	0.9889
	DRDN+ [83]	42.66	**0.9893**
	DPN [62]	40.1	0.9846
	TENet [88]	31.39	0.8965
	PIPNet [1]	39.37	0.9768
	RLDD [34]	**42.76**	0.9893
McMaster [134]	DRDN [83]	38.88	0.9689
	DRDN+ [83]	39.02	0.9697
	DPN [62]	37.6	**0.9842**
	TENet [88]	32.40	0.9163
	PIPNet [1]	38.13	0.9612
	RLDD [34]	36.61	0.9725
Urban 100 [42]	RLDD [34]	39.52	0.9864
	DPN [62]	37.70	0.9799
	PIPNet [1]	37.51	0.9731
	TENet [88]	29.37	0.9061
	SGNet [73]	34.54	0.9533
	RLDD [34]	**39.52**	**0.9864**
	RestoreNet w/ PatchNet [108]	34.66	-
SID Dataset [13]	ReconfigISP [129]	25.65	0.7527
	CameraNet [70]	22.47	0.744
S7 ISP [101]	Reconfig ISP [129]	23.31	0.7007
HDR+ [37]	CameraNet [70]	24.98	0.858
Ciuera and Funt [18]	Deep Camera [91]	30.71	-
Sony IMX586 Quad Bayer RGB mobile sensor [44]	CSANet [44]	23.73	0.8487

making them well-suited for image processing tasks that require detailed attention to pixel-level features.

Recently, a pioneering method utilizing **Vision Transformers (ViTs)** for end-to-end ISP has emerged [26]. This approach leverages the global attention mechanism inherent to ViTs, allowing the model to capture long-range dependencies and context across the entire image. Although this method is relatively new, it shows promise, particularly in tasks where understanding global image context is crucial, such as color correction and tone mapping.

4.1.2 Performance Analysis Based on Dataset Results. The effectiveness of these end-to-end methods has been evaluated across multiple datasets, with results presented in Table 1. This table provides a comparative analysis of the methods' performance based on metrics such as **Peak Signal-to-Noise Ratio (PSNR)** and **Structural Similarity Index Measure (SSIM)**. Based on the results, we provide a deeper discussion about the methods with the best results: LiteISPNet [139], RLDD [34], and PyNET-CA [61]. The comparison focuses on the architectural differences and their impact on the overall performance, particularly explaining why LiteISPNet and the RLDD might outperform PyNET-CA under certain conditions.

LiteISPNet is designed with efficiency in mind, featuring a lightweight architecture that minimizes computational complexity while maintaining competitive performance. This efficiency is achieved through a reduction in the number of parameters and a simplified network design, which allows LiteISPNet to process images more quickly and with lower resource requirements compared to more complex architectures like PyNET-CA.

The architectural simplicity of LiteISPNet makes it particularly effective in scenarios where computational resources are limited. Despite its reduced complexity, LiteISPNet still manages to deliver high-quality results in end-to-end ISP pipeline. The balance between efficiency and performance gives LiteISPNet an edge over PyNET-CA, especially when real-time processing is necessary.

RLDD leverages a residual learning framework to address the two tasks simultaneously. By integrating demosaicing and denoising into a single architecture, this approach minimizes the potential loss of information that might occur when these tasks are performed sequentially in separate stages, as seen in more complex pipelines.

The strength of this method lies in its ability to effectively preserve fine details and reduce noise without introducing artifacts. This is particularly advantageous in environments where image quality is paramount, such as professional photography or medical imaging. The residual learning framework's relative simplicity, combined with its targeted approach, often results in superior performance for these specific tasks compared to the more generalized PyNET-CA architecture.

4.1.3 PyNET-CA. PyNET-CA is a more complex architecture that incorporates attention mechanisms to enhance the processing of image data. While this complexity allows PyNET-CA to achieve high-quality results, particularly in capturing fine details and handling diverse lighting conditions, it also comes with significant drawbacks. The increased computational load and longer training times make PyNET-CA less suitable for real-time applications or deployment on devices with limited processing power.

Moreover, the complexity of PyNET-CA does not always translate to a clear advantage in performance, particularly in tasks like demosaicing and denoising where more specialized methods like LiteISPNet and the Residual Learning approach can achieve similar or better results with far less computational overhead. The generalist nature of PyNET-CA, while powerful, can sometimes be less effective than these more focused, efficient architectures.

4.1.4 Conclusion Based on Table Analysis. Based on the analysis of the results in the tables, CNN-based methods currently dominate the end-to-end ISP landscape, offering a reliable balance between performance and computational efficiency. The Vision Transformer-based method, while showing potential, particularly in tasks requiring global context, is still in its early stages and requires further exploration and validation. The table indicates that methods using ViTs, GANs, and Diffusion Models should be more explored in this specific domain

In conclusion, the state of the art in end-to-end ISP using deep learning is largely driven by classical CNNs. The comparative results highlight the strengths and limitations of these approaches, providing a clear direction for future research and development in this area.

4.2 Critical Insights into Deep Learning Approaches for ISP

While deep learning has revolutionized ISP by enabling automated feature extraction and optimization, it is crucial to critically examine the underlying trends, promises, and limitations associated with these approaches.

4.2.1 Trends. Deep learning approaches for ISP have evolved significantly, moving from simpler convolutional architectures to more complex and powerful models such as Vision Transformers (ViTs). A key trend is the increasing adoption of *end-to-end* pipelines, which simplify the ISP workflow by encapsulating all processing steps within a single model. Another important trend is the rise of ***Self-Supervised Learning (SSL)***, which reduces the reliance on large labeled datasets by leveraging unlabeled data, thus addressing one of the major bottlenecks in deep learning applied to image processing problems [17, 52, 135], however, we could not find any SSL method applied to end-to-end ISP processing, this could be a trend for research.

4.2.2 Promises. The promise of deep learning in ISP lies in its ability to learn intricate patterns and correlations that are often missed by traditional methods [98]. For example, deep learning models can achieve state-of-the-art performance in tasks like denoising [87], demosaicing [46], and super-resolution [133], often surpassing classical algorithms. Moreover, the flexibility of these models allows for continuous improvement as more data becomes available, making them well-suited for applications in rapidly evolving fields such as autonomous driving and medical imaging.

4.2.3 Limitations. Despite the significant advancements, there are inherent limitations in applying deep learning to ISP. One major challenge is the *lack of interpretability* in these models. Unlike traditional methods where the processing pipeline is well-understood and each step is transparent, Deep Learning models operate as "black boxes," making it difficult to diagnose errors or understand the decision-making process. Even though there are methods for interpretability in Deep Learning [103, 116], these methods might not help specifically in the end-to-end ISP pipeline based on Deep Learning methods because they would not highlight the areas that contribute with errors.

Another limitation is the *data dependency* of these models. High performance models often requires vast amounts of labeled data, which can be difficult to obtain, particularly in specialized domains like image signal processing. Additionally, these models are computationally intensive, requiring significant resources for training and inference, which may not be feasible in all deployment environments.

Finally, there is the issue of *overfitting* [98] to specific datasets. While deep learning models can achieve high accuracy on training data, they may not generalize well to unseen data, especially when the training data lacks diversity. This raises concerns about the robustness and reliability of these models in real-world applications.

In conclusion, while deep learning holds great promise for advancing ISP, it is essential to critically assess these approaches to understand their strengths and limitations. This balanced view will guide the development of more robust and effective ISP solutions in the future.

4.3 Challenges

Challenges and benchmarks were crucial to fast advance some research fields [96], and it was not different for this. The AIM 2019 Challenge on RAW to RGB Mapping [50] brought new perspectives to this area, with the creation of Zurich Raw to RGB Dataset and the use of comparison metrics which are used in image restoration benchmark datasets. Still, with the addition of **mean opinion score (MOS)**, possibility the comparison with the same measures method to many works besides the definition of new states of the art. This challenge encouraged the creation of two others: the AIM 2020 Challenge on Learned Image Signal Processing Pipeline [48] with a similar structure, and the Learned Smartphone ISP on Mobile NPUs with Deep Learning Mobile AI 2021 Challenge [44]. The Mobile AI 2021 Challenge has the addition of runtime as a comparison metric, evaluation in mobile NPUs, and creation of a dataset similar to the Zurich Dataset [50]. As can be observed, the creation of these challenges possibility a base of comparison between works and encourages the development of new works. Finally, the creation of fresh challenges will help the advancement of this area.

One of the major challenges in deep learning-based ISP is the scarcity of high-quality RAW datasets and the alignment issues between synthetic and real-world benchmarks. These challenges are particularly relevant when trying to train models that generalize well across different devices and conditions.

4.4 Dataset Created with Different Cameras

Some datasets [44, 49] contain paired images captured by different cameras. One of the primary goals of studies using these datasets is to learn the characteristics of high-quality cameras and transfer them to more constrained devices. However, due to differences between devices, these datasets often face alignment issues with their image patch pairs. Mapping from RAW to RGB can be particularly challenging, as the corresponding pixels in the patches may not align perfectly.

A straightforward solution is to align the image pairs, but determining the best alignment method remains a key challenge. Zhang et al. [139] addressed this by using Deep Learning techniques, achieving superior results compared to models trained on misaligned datasets. Another approach commonly used in the literature [21, 35, 44, 49] is adapting the loss function to avoid penalizing models when the generated image is misaligned with the ground truth. Loss functions that prioritize human perceptual similarity tend to be more robust to misalignments and produce images with greater detail.

Currently, the most effective metrics for addressing this issue leverage Deep Learning by extracting features from images using pre-trained models [82, 136]. This raises the question of how best to address misalignment issues without adding significant computational overhead to the training process. This challenge could become a research focus in its own right, encompassing the development of alignment methods for RAW and RGB images from different cameras. Such methods would facilitate the creation of new datasets and enhance the performance of future models.

4.5 Methodologies

The justification for each stage, from architecture construction to the choice of datasets and validation of the work, is crucial for sharing knowledge and advancing the research field. However, some works fail to provide such justifications. For example, Liu et al. [73] do not adequately explain the use of certain datasets, and Liu et al. [72] do not provide a clear rationale for using the U-Net model. Additionally, many methods do not make their source code available in the paper

[34, 70, 78, 83, 89, 91], making future validations and comparisons with other methods difficult and reducing the reliability of the validation process.

4.6 Evaluation Protocols

Train models using synthesized RAW image data when RAW and RGB images are scarce in the dataset is a good way to increase generalization. However, this cannot be a great option to validate methods proposed to map ISP because the synthesized RAW image data results from interpolation of RGB processed images, has pixels with less information, and the resulting noises of camera hardware are complex and hard to generate. Nevertheless, many works use datasets created for tasks like denoising, deblurring, or super-resolution with the generation of RAW images from these datasets that disbelieve the applicability of these methods in the real world. This problem raises the question of how to create RAW to RGB datasets with good representation and high quality for further research.

On the other hand, few works tested the models in embedded systems or mobile devices. This can be a problem because most applications of ISPs are in devices with low computational capacity, and the fault of validation in this way cannot prove the possible use in real applications. In view of this, the Mobile 2021 Challenge released the task: "Learned Smartphone ISP on Mobile NPUs with Deep Learning." [44] It considers the processing and time of execution in mobile NPUs and the image quality like the other two previous challenges. This proposal can encourage future works to care more about these points.

4.7 More Recent Deep Learning Methods for Software ISP

In the realm of Software ISP, recent advancements in deep learning methods have garnered attention for their potential to revolutionize image processing workflows. However, a critical examination reveals a discernible scarcity of research in exploring these cutting-edge techniques in the specific context of RAW to RGB conversion for mobile devices.

One area of interest lies in the application of more recent deep learning architectures, such as Transformers, Diffusion Models, and **Generative Adversarial Networks (GANs)**, to the challenges posed by RAW image data. Despite their success in various domains, including natural language processing and image generation, the literature review exposed a limited body of work focusing on their role in Software ISP.

Transformers: The transformative impact of Transformers in tasks like language understanding has been well-established, yet their application to RAW to RGB conversion within the framework of Software ISP remains an underexplored domain.

Diffusion Models: Diffusion models, with their ability to model complex distributions, present a compelling avenue for image processing. However, the literature reveals a scarcity of research investigating their efficacy in the specific context of RAW data transformation.

GANs: Generative Adversarial Networks, known for their ability to generate realistic images, offer a promising direction for Software ISP. Nevertheless, their role in the nuanced task of RAW to RGB conversion on mobile devices lacks comprehensive exploration.

In light of the identified gaps, the field beckons researchers to delve into the intersection of these more recent deep learning methods and Software ISP, particularly in the context of RAW image manipulation. This underexplored territory offers a fertile ground for future investigations, presenting an exciting opportunity to enhance the capabilities of image processing software for mobile devices and beyond.

4.8 Limitations in Performance Comparisons

While this survey provides a comprehensive comparison of image reconstruction performance using metrics such as PSNR and SSIM, it is important to acknowledge certain limitations. Specifically,

metrics related to training time, computational efficiency, or memory usage are not consistently reported across the surveyed studies. This inconsistency limits the ability to perform a holistic comparison of execution performance.

Future research could address this gap by adopting standardized benchmarks that evaluate not only image quality but also factors like computational complexity, energy efficiency, and inference time. Such benchmarks would provide a more complete evaluation of deep learning-based ISP methods, facilitating a better understanding of the trade-offs between accuracy and efficiency, particularly in resource-constrained environments.

4.9 Future Research Directions

While significant progress has been made in deep learning-based ISP, several areas remain open for further exploration. One promising direction is the combination of synthetic and real-world data to enhance model generalization. Techniques that generate synthetic RAW images, such as CycleISP [130], could be further refined to produce data that resemble more closely real-world conditions, particularly by incorporating more realistic noise models that capture variations in sensor behavior, lighting conditions, and compression artifacts.

Furthermore, future work could focus on improving alignment techniques between synthetic and real-world datasets. Adaptive alignment methods that take advantage of machine learning could ensure more precise matching between paired RAW and RGB images, reducing artifacts and improving training effectiveness.

These directions offer actionable pathways for enhancing the robustness, generalization, and perceptual quality of deep learning-based ISP models in future studies.

5 Conclusion

The ISP pipeline is an important combination of techniques, essential for the creation of quality digital images from camera sensors. This survey provided an in-depth research on the applications of deep learning techniques to ISP tasks, regarding the application of networks for solving partial steps or the complete pipeline. Additionally, it also provided an introduction to both ISP and deep learning areas, alongside with a detailed overview of software ISP, regarding its fundamentals and individual steps.

The works surveyed in this paper were selected based on their novelty, their target task, and the applied deep learning techniques. Among the 27 reviewed papers, 30% had applied DNNs for replacing the complete ISP pipeline. This reveals a new trend that explores the generalization ability of CNNs to learn all the ISP individual tasks.

Furthermore, this work also summarized the most commonly used datasets for evaluation. The availability of quality datasets is necessary for the research and development of new solutions to any deep learning application area. In this scenario, new ISP-related datasets can improve some limitations existent in the surveyed datasets.

Finally, during the development of this survey, some critical points were detected and are highlighted below:

- The use of different cameras for producing RAW-to-RGB datasets creates alignment issues that require the application of additional techniques during the training of DNNs. These mitigators can add computational costs and interference in the overall performance of the networks;
- As more approaches are proposed to the ISP pipeline replacement task, a more consistent evaluation procedure, alongside with the definition of common target datasets;
- Besides being one of the main target applications, mobile application for networks that replace the complete ISP pipeline are not often discussed. A more in-depth evaluation of

the proposed methods performance on edge devices is important to identify components that can be optimized.

As future work, we intend to pursue two main approaches to Deep Learning-based ISP applications: the development of a complete ISP pipeline network focused on execution in a Raspberry Pi board, aiming to explore the deployment on edge challenge; and the application of vision transformers to the complete ISP task, aiming to explore the good results achieved by transformers in other computer vision tasks.

References

[1] S. M. A. Sharif, Rizwan Ali Naqvi, and Mithun Biswas. 2021. Beyond joint demosaicking and denoising: An image processing pipeline for a pixel-bin image sensor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 233–242.

[2] Michael S. Brown, Abdelrahman Abdelhamed, and Stephen Lin. 2018. A high-quality denoising dataset for smartphone cameras. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (June 2018). https://doi.org/10.1109/CVPR.2018.00182

[3] Raquel Urtasun, Andreas Geiger, and Philip Lenz. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition* (June 2012). https://doi.org/10.1109/CVPR.2012.6248074

[4] Saeed Anwar and Nick Barnes. 2020. Real image denoising with feature attention. arXiv:cs.CV/1904.07396

[5] Ammar Askar, Abbas Askar, Mario Pasquato, and Mirek Giersz. 2019. Finding black holes with black boxes – using machine learning to identify globular clusters with black hole subsystems. *Mon. Not. R. Astron. Soc.* 485, 4 (June 2019), 5345–5362.

[6] Jiawen Chen, Dillon Sharlet, Ren Ng, Robert Carroll, Ben Mildenhall, and Jonathan T. Barron. 2018. Burst denoising with kernel prediction networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (June 2018). https://doi.org/10.1109/CVPR.2018.00265

[7] Daniel Berman, Anna Buczak, Jeffrey Chavis, and Cherita Corbett. 2019. A survey of deep learning methods for cyber security. *Information* 10, 4 (April 2019), 122. https://doi.org/10.3390/info10040122

[8] Forrest Iandola, Peter H. Jin, Kurt Keutzer, Bichen Wu, and Alvin Wan. 2016. SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. arXiv:cs.CV/1612.01051

[9] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T. Barron. 2018. Unprocessing images for learned raw denoising. arXiv:cs.CV/1811.11127

[10] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. 2005. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation* 4, 2 (2005), 490–530.

[11] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Fredo Durand. 2011. Learning photographic global tonal adjustment with a database of input / output image pairs. In *CVPR 2011*. 97–104. https://doi.org/10.1109/CVPR.2011.5995332

[12] Prashant Chaudhari, Franziska Schirrmacher, Andreas Maier, Christian Riess, and Thomas Köhler. 2021. Merging-ISP: Multi-exposure high dynamic range image signal processing. arXiv:eess.IV/1911.04762

[13] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. 2018. Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3291–3300.

[14] K. Chen, V. Kvasnicka, P. C. Kanen, and S. Haykin. 2001. Multi-valued and universal binary neurons: Theory, learning, and applications [book review]. *IEEE Trans. Neural Netw.* 12, 3 (May 2001), 647–647.

[15] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2017. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. arXiv:cs.CV/1606.00915

[16] Yu-Sheng Chen, Yu-Ching Wang, Man-Hsin Kao, and Yung-Yu Chuang. 2018. Deep photo enhancer: Unpaired learning for image enhancement from photographs with GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6306–6314.

[17] Zhaojie Chen, Qi Li, Huajun Feng, Zhihai Xu, Yueting Chen, and Tingting Jiang. 2024. Dehaze on small-scale datasets via self-supervised learning. *The Visual Computer* 40, 6 (2024), 4235–4249.

[18] Florian Ciurea and Brian V. Funt. 2003. A large image database for color constancy research. In *Color Imaging Conference*.

[19] Laurent Condat and Saleh Mosaddegh. 2012. Joint demosaicking and denoising by total variation minimization. In *2012 19th IEEE International Conference on Image Processing*. 2781–2784. https://doi.org/10.1109/ICIP.2012.6467476

[20] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 764–773.

[21] Linhui Dai, Xiaohong Liu, Chengqi Li, and Jun Chen. 2020. AWNet: Attentive wavelet network for image ISP. arXiv:eess.IV/2008.09228
[22] Padideh Danaee, Reza Ghaeini, and David A. Hendrix. 2017. A deep learning approach for cancer detection and relevant gene identification. *Pac. Symp. Biocomput.* 22 (2017), 219–229.
[23] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2015. Image super-resolution using deep convolutional networks. arXiv:cs.CV/1501.00092
[24] Weishong Dong, Ming Yuan, Xin Li, and Guangming Shi. 2018. Joint demosaicing and denoising with perceptual optimization on a generative adversarial network. *arXiv preprint arXiv:1802.04723* (2018).
[25] Linwei Fan, Fan Zhang, Hui Fan, and Caiming Zhang. 2019. Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art* 2, 1 (July 2019). https://doi.org/10.1186/s42492-019-0016-7
[26] Zhihao Fan, Xun Wu, Fanqing Meng, Yaqi Wu, and Feng Zhang. 2023. OTST: A two-phase framework for joint denoising and remosaicing in RGBW CFA. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2832–2841.
[27] Claudio Filipi Gonçalves dos Santos, Diego de Souza Oliveira, Leandro A. Passos, Rafael Gonçalves Pires, Daniel Felipe Silva Santos, Lucas Pascotti Valem, Thierry P. Moreira, Marcos Cleison S. Santana, Mateus Roder, Jo Paulo Papa, et al. 2022. Gait recognition based on deep learning: A survey. *ACM Comput. Surv.* 55, 2, Article 34 (Jan. 2022), 34 pages. https://doi.org/10.1145/3490235
[28] R. Franzen. 1999. Kodak Lossless True Color Image Suite. http://r0k.us/graphics/kodak/
[29] L. van der Maaten, G. Huang, Z. Liu, and K. Q. Weinberger. 2017. Densely connected convolutional networks. *IEEE Conf. Comput. Vis. Pattern Recognit.* (2017), 2261–2269.
[30] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. 2016. Deep joint demosaicking and denoising. *ACM Trans. Graph.* 35, 6, Article 191 (Nov. 2016), 12 pages.
[31] Michaël Gharbi, Jiawen Chen, Jonathan T. Barron, Samuel W. Hasinoff, and Frédo Durand. 2017. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–12.
[32] Faustino J. Gomez. 2005. Co-evolving recurrent neurons learn deep memory POMDPs. In *InGECCO-05: Proceedings of the Genetic and Evolutionary Computation Conference*. 491–498.
[33] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. 2020. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics* 37, 3 (April 2020), 362–386. https://doi.org/10.1002/rob.21918
[34] Yu Guo, Qiyu Jin, Gabriele Facciolo, Tieyong Zeng, and Jean-Michel Morel. 2020. Residual learning for effective joint demosaicing-denoising. arXiv:cs.CV/2009.06205
[35] Saumya Gupta, Diplav Srivastava, Umang Chaturvedi, Anurag Jain, and Gaurav Khandelwal. 2021. Del-Net: A single-stage network for mobile camera ISP. arXiv:eess.IV/2108.01623
[36] Samuel W. Hasinoff. 2014. *Photon, Poisson Noise*. Springer US, Boston, MA, USA, 608–610. https://doi.org/10.1007/978-0-387-31439-6_482
[37] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. 2016. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (TOG)* 35 (2016), 1–12.
[38] Felix Heide, Markus Steinberger, Yun-Ta Tsai, Mushfiqur Rouf, Dawid Pająk, Dikpal Reddy, Orazio Gallo, Jing Liu, Wolfgang Heidrich, Karen Egiazarian, Jan Kautz, and Kari Pulli. 2014. FlexISP: A flexible camera image processing framework. *ACM Trans. Graph.* 33, 6, Article 231 (Nov. 2014), 13 pages.
[39] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. arXiv:stat.ML/1503.02531
[40] Ming-Chun Hsyu, Chih-Wei Liu, Chao-Hung Chen, Chao-Wei Chen, and Wen-Chia Tsai. 2021. CSANet: High speed channel spatial attention network for mobile ISP. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'21)*. 2486–2493. https://doi.org/10.1109/CVPRW53098.2021.00282
[41] Yuanming Hu, Hao He, Chenxi Xu, Baoyuan Wang, and Stephen Lin. 2018. Exposure: A white-box photo post-processing framework. *ACM Transactions on Graphics (TOG)* 37, 2 (2018), 1–17.
[42] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. 2015. Single image super-resolution from transformed self-exemplars. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*. 5197–5206. https://doi.org/10.1109/CVPR.2015.7299156
[43] Lei Liao, Ronghe Chu, Jingyu Yang, Huanjing Yue, and Cong Cao. 2020. Supervised raw video denoising with a benchmark dataset on dynamic scenes. arXiv:cs.CV/2003.14013
[44] Andrey Ignatov, Cheng-Ming Chiang, Hsien-Kai Kuo, Anastasia Sycheva, Radu Timofte, Min-Hung Chen, Man-Yu Lee, Yu-Syuan Xu, Yu Tseng, Shusong Xu, et al. 2021. Learned smartphone ISP on mobile NPUs with deep learning, mobile AI 2021 challenge: Report. arXiv:eess.IV/2105.07809
[45] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. 2017. DSLR-quality photos on mobile devices with deep convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 3277–3285.

[46] Andrey Ignatov, Grigory Malivenko, Radu Timofte, Yu Tseng, Yu-Syuan Xu, Po-Hsiang Yu, Cheng-Ming Chiang, Hsien-Kai Kuo, Min-Hung Chen, Chia-Ming Cheng, et al. 2022. PyNet-V2 mobile: Efficient on-device photo processing with neural networks. In *2022 26th International Conference on Pattern Recognition (ICPR'22)*. IEEE, 677–684.

[47] Andrey Ignatov, Anastasia Sycheva, Radu Timofte, Yu Tseng, Yu-Syuan Xu, Po-Hsiang Yu, Cheng-Ming Chiang, Hsien-Kai Kuo, Min-Hung Chen, Chia-Ming Cheng, et al. 2022. MicroISP: Processing 32mp photos on mobile devices with deep learning. In *European Conference on Computer Vision*. Springer, 729–746.

[48] Andrey Ignatov, Radu Timofte, Zhilu Zhang, Ming Liu, Haolin Wang, Wangmeng Zuo, Jiawei Zhang, Ruimao Zhang, Zhanglin Peng, Sijie Ren, et al. 2020. AIM 2020 challenge on learned image signal processing pipeline. (2020). arXiv:cs.CV/2011.04994

[49] Andrey Ignatov, Luc Van Gool, and Radu Timofte. 2020. Replacing mobile camera ISP with a single deep learning model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 536–537.

[50] Andrey D. Ignatov, Juncheng Li, Jiajie Zhang, Haoyu Wu, Jie Li, Rui Huang, Muhammad Haris, Greg Shakhnarovich, Norimichi Ukita, Yuzhi Zhao, et al. 2019. AIM 2019 challenge on RAW to RGB mapping: Methods and results. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW'19)* (2019), 3584–3590.

[51] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1125–1134.

[52] Hyemi Jang, Junsung Park, Dahuin Jung, Jaihyun Lew, Ho Bae, and Sungroh Yoon. 2024. PUCA: Patch-unshuffle and channel attention for enhanced self-supervised image denoising. *Advances in Neural Information Processing Systems* 36 (2024).

[53] Xiaozhong Ji, Yun Cao, Ying Tai, Chengjie Wang, Jilin Li, and Feiyue Huang. 2020. Real-world super-resolution via kernel estimation and noise injection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2020), 1914–1923.

[54] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*. Springer, 694–711.

[55] Jun Nishimura, Timo Gerasimow, Sushma Rao, Aleksandar Sutic, Chyuan-Tyng Wu, and Gilad Michael. 2019. Automatic ISP image quality tuning using non-linear optimization. arXiv:cs.CV/1902.09023

[56] S. Ren, K. He, X. Zhang, and J. Sun. 2016. Deep residual learning for image recognition. arXiv:cs.CV/1512.03385

[57] Nima Khademi Kalantari, Ravi Ramamoorthi, et al. 2017. Deep high dynamic range imaging of dynamic scenes. *ACM Trans. Graph.* 36, 4 (2017), 144–1.

[58] Andreas Kamilaris and Francesc X. Prenafeta-Boldú. 2018. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture* 147 (April 2018), 70–90. https://doi.org/10.1016/j.compag.2018.02.016

[59] Hakki Can Karaimer and Michael S. Brown. 2016. A software platform for manipulating the camera imaging pipeline. In *European Conference on Computer Vision (ECCV'16)*.

[60] Daniel Khashabi, Sebastian Nowozin, Jeremy Jancsary, and Andrew W. Fitzgibbon. 2014. Joint demosaicing and denoising via learned nonparametric random fields. *IEEE Transactions on Image Processing* 23, 12 (2014), 4968–4981.

[61] Byung-Hoon Kim, Joonyoung Song, Jong Chul Ye, and JaeHyun Baek. 2020. PyNET-CA: Enhanced PyNET with channel attention for end-to-end mobile image signal processing. In *European Conference on Computer Vision*. Springer, 202–212.

[62] Irina Kim, Seongwook Song, Soonkeun Chang, Sukhwan Lim, and Kai Guo. 2020. Deep image demosaicing for submicron image sensors. *Electronic Imaging* 2020, 7 (2020), 60410–1.

[63] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. 2016. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1646–1654.

[64] Boon Tatt Koik and Haidi Ibrahim. 2013. A literature survey on blur detection algorithms for digital imaging. *2013 1st International Conference on Artificial Intelligence, Modelling and Simulation* (2013), 272–277.

[65] Filippos Kokkinos and Stamatios Lefkimmiatis. 2018. Deep image demosaicking using a cascade of convolutional residual denoising networks. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. 303–319.

[66] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. arXiv:cs.CV/1609.04802

[67] Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang. 2018. Multi-scale residual network for image super-resolution. In *ECCV*.

[68] Xin Li, Bahadir Gunturk, and Lei Zhang. 2008. Image demosaicing: A systematic survey. In *Visual Communications and Image Processing 2008*, William A. Pearlman, John W. Woods, and Ligang Lu (Eds.). SPIE. https://doi.org/10.1117/12.766768

[69] Zewen Li, Wenjie Yang, Shouheng Peng, and Fan Liu. 2020. A survey of convolutional neural networks: Analysis, applications, and prospects. arXiv:arXiv:2004.02806

[70] Zhetong Liang, Jianrui Cai, Zisheng Cao, and Lei Zhang. 2021. CameraNet: A two-stage framework for effective camera ISP learning. *IEEE Transactions on Image Processing* 30 (2021), 2248–2262.

[71] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A. W. M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. 2017. A survey on deep learning in medical image analysis. *Medical Image Analysis* 42 (Dec. 2017), 60–88. https://doi.org/10.1016/j.media.2017.07.005

[72] Jiaming Liu, Chi-Hao Wu, Yuzhi Wang, Qin Xu, Yuqian Zhou, Haibin Huang, Chuan Wang, Shaofan Cai, Yifan Ding, Haoqiang Fan, et al. 2019. Learning raw image denoising with Bayer pattern unification and Bayer preserving augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 0–0.

[73] Lin Liu, Xu Jia, Jianzhuang Liu, and Qi Tian. 2020. Joint demosaicing and denoising with self guidance. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*. 2237–2246. https://doi.org/10.1109/CVPR42600.2020.00231

[74] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[75] Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. 2020. Structure-preserving super resolution with gradient guidance. arXiv:eess.IV/2003.13081

[76] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. 2016. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing* 26, 2 (2016), 1004–1016.

[77] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proceedings of the IEEE International Conference on Computer Vision* 2, 416–423 vol.2. https://doi.org/10.1109/ICCV.2001.937655

[78] Kangfu Mei, Juncheng Li, Jiajie Zhang, Haoyu Wu, Jie Li, and Rui Huang. 2019. HighEr-Resolution Network for image demosaicing and enhancing. arXiv:eess.IV/1911.08098

[79] Daniele Menon and Giancarlo Calvagno. 2011. Color image demosaicking: An overview. *Signal Processing: Image Communication* 26 (Oct. 2011). https://doi.org/10.1016/j.image.2011.04.003

[80] Sean Moran, Steven McDonagh, and Gregory Slabaugh. 2021. CURL: Neural curve layers for global image enhancement. In *2020 25th International Conference on Pattern Recognition (ICPR'21)*. IEEE, 9796–9803.

[81] Mukesh Motwani, Mukesh Gadiya, Rakhi Motwani, and Frederick Harris. 2004. Survey of image denoising techniques. (01 2004).

[82] Aamir Mustafa, Aliaksei Mikhailiuk, Dan Andrei Iliescu, Varun Babbar, and Rafal K. Mantiuk. 2021. Training a task-specific image reconstruction loss. arXiv:eess.IV/2103.14616

[83] Bumjun Park and Jechang Jeong. 2019. Color filter array demosaicking using densely connected residual network. *IEEE Access* 7 (2019), 128076–128085.

[84] Jongchan Park, Joon-Young Lee, Donggeun Yoo, and In So Kweon. 2018. Distort-and-recover: Color enhancement using deep reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5928–5936.

[85] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2337–2346.

[86] Ibrahim Pekkucuksen and Yucel Altunbasak. 2010. Gradient based threshold free color filter array interpolation. In *2010 IEEE International Conference on Image Processing*. 137–140. https://doi.org/10.1109/ICIP.2010.5654327

[87] Rafael G. Pires, Daniel F. S. Santos, Cláudio F. G. Santos, Marcos C. S. Santana, and Joao P. Papa. 2020. Image denoising using attention-residual convolutional neural networks. In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI'20)*. IEEE, 101–107.

[88] Guocheng Qian, Yuanhao Wang, Chao Dong, Jimmy S. Ren, Wolfgang Heidrich, Bernard Ghanem, and Jinjin Gu. 2021. Rethinking the pipeline of demosaicing, denoising and super-resolution. arXiv:eess.IV/1905.02538

[89] Ramchalam Ramakrishnan, Shangling Jui, and Vahid Partovi Nia. 2019. Deep demosaicing for edge implementation. In *International Conference on Image Analysis and Recognition*. Springer, 275–286.

[90] Rajeev Ramanath, Wesley E. Snyder, Youngjun Yoo, and Mark S. Drew. 2005. Color image processing pipeline. *IEEE Signal Processing Magazine* 22, 1 (2005), 34–43.

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

[92] Qolamreza Razlighi and Nasser Kehtarnavaz. 2007. Imaeg blur reduction for cell-phone cameras via adaptive tonal correction. 1 (09 2007), I – 113. https://doi.org/10.1109/ICIP.2007.4378904

[93] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 234–241.

[94] F. Rosenblatt. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* (1958), 65–386.
[95] Dr. Sabeenian R.S. 2012. A survey on image denoising algorithms (IDA). *Science, Measurement and Technology, IEE Proceedings A* 1 (11 2012), 456–462.
[96] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet large scale visual recognition challenge. arXiv:cs.CV/1409.0575
[97] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2019. MobileNetV2: Inverted residuals and linear bottlenecks. arXiv:cs.CV/1801.04381
[98] Claudio Filipi Gonçalves dos Santos and João Paulo Papa. 2022. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. *ACM Comput. Surv.* 54, 10s, Article 213 (Sep. 2022), 25 pages. https://doi.org/10.1145/3510413
[99] Juergen Schmidhuber. 2015. Deep learning. *Scholarpedia J.* 10, 11 (2015), 32832.
[100] Eli Schwartz, Alex Bronstein, and Raja Giryes. 2021. ISP distillation. arXiv:cs.CV/2101.10203
[101] Eli Schwartz, Raja Giryes, and Alex M. Bronstein. 2019. DeepISP: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing* 28, 2 (Feb. 2019), 912–923. https://doi.org/10.1109/tip.2018.2872858
[102] Eli Schwartz. [n. d.]. DeepISP: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing* 28, 2 ([n. d.]). https://doi.org/10.1109/tip.2018.2872858
[103] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2020. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision* 128 (2020), 336–359.
[104] Andrew W. Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W. R. Nelson, Alex Bridgland, et al. 2020. Improved protein structure prediction using potentials from deep learning. *Nature* 577, 7792 (Jan. 2020), 706–710.
[105] Zhetong Liang Shi Guo and Lei Zhang. 2021. Joint denoising and demosaicking with green channel prior for real-world burst images. arXiv:cs.CV/2101.09870
[106] Hugo Storm, Kathy Baylis, and Thomas Heckelei. 2020. Machine learning in agricultural and applied economics. *Eur. Rev. Agric. Econ.* 47, 3 (June 2020), 849–892.
[107] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. 2018. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. arXiv:cs.CV/1709.02371
[108] Shuyang Sun, Liang Chen, Gregory Slabaugh, and Philip Torr. 2020. Learning to sample the most useful training patches from images. *arXiv preprint arXiv:2011.12097* (2020).
[109] Hanlin Tan, Xiangrong Zeng, Shiming Lai, Yu Liu, and Maojun Zhang. 2017. Joint demosaicing and denoising of noisy Bayer images with ADMM. In *2017 IEEE International Conference on Image Processing (ICIP'17)*. 2951–2955. https://doi.org/10.1109/ICIP.2017.8296823
[110] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. 2020. Deep learning on image denoising: An overview. arXiv:eess.IV/1912.13171
[111] Jiajun Wu, Donglai Wei, William T. Freeman, Tianfan Xue, and Baian Chen. 2019. Video enhancement with task-oriented flow. arXiv:cs.CV/1711.09078
[112] Stefan Roth and Tobias Plötz. 2017. Benchmarking denoising algorithms with real photographs. arXiv:cs.CV/1707.01313
[113] Kwang-Hyun Uhm, Kyuyeon Choi, Seung-Won Jung, and Sung-Jea Ko. 2021. Image compression-aware deep camera ISP network. *IEEE Access* 9 (2021), 137824–137832. https://doi.org/10.1109/ACCESS.2021.3116702
[114] Kwang-Hyun Uhm, Seung-Wook Kim, Seo-Won Ji, Sung-Jin Cho, Jun-Pyo Hong, and Sung-Jea Ko. 2019. W-Net: Two-stage U-Net with misaligned data for Raw-to-RGB mapping. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (Oct. 2019). https://doi.org/10.1109/iccvw.2019.00448
[115] Fagun Vankawala, Amit Ganatra, and Amit Patel. 2015. A survey on different image deblurring techniques. *International Journal of Computer Applications* 116 (2015), 15–18.
[116] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. 2020. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 24–25.
[117] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. 2019. Underexposed photo enhancement using deep illumination estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6849–6857.
[118] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. 2018. ESRGAN: Enhanced super-resolution generative adversarial networks. arXiv:cs.CV/1809.00219

[119] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Vol. 2. IEEE, 1398–1402.
[120] Peter Wilson. 2016. Chapter 7 - High speed video application. In *Design Recipes for FPGAs (Second Edition)* , Peter Wilson (Ed.). Newnes, Oxford, 67–77. https://doi.org/10.1016/B978-0-08-097129-2.00007-6
[121] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. CBAM: Convolutional block attention module. arXiv:cs.CV/1807.06521
[122] Chyuan-Tyng Wu, Leo F. Isikdogan, Sushma Rao, Bhavin Nayak, Timo Gerasimow, Aleksandar Sutic, Liron Ainkedem, and Gilad Michael. 2019. VisionISP: Repurposing the image signal processor for computer vision applications. In *2019 IEEE International Conference on Image Processing (ICIP'19)*. IEEE, 4624–4628.
[123] Menghan Xia, Xueting Liu, and Tien-Tsin Wong. 2018. Invertible grayscale. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–10.
[124] Yazhou Xing, Zian Qian, and Qifeng Chen. 2021. Invertible image signal processing. arXiv:eess.IV/2103.15061
[125] Ke Yu Chao Dong Chen Change Loy Xintao Wang, Kelvin C.K. Chan. 2019. EDVR: Video restoration with enhanced deformable convolutional networks. arXiv:cs.CV/1905.02716
[126] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. arXiv:cs.LG/1505.00853
[127] Run xu Tan, K. Zhang, Wangmeng Zuo, and Lei Zhang. 2017. Color image demosaicking via deep residual learning.
[128] Liu Yongji and Yuan Xiaojun. 2020. A design of dynamic defective pixel correction for image sensor. In *2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS'20)*. 713–716. https://doi.org/10.1109/ICAIIS49377.2020.9194921
[129] Ke Yu, Zexian Li, Yue Peng, Chen Change Loy, and Jinwei Gu. 2021. ReconfigISP: Reconfigurable camera image processing pipeline. arXiv:eess.IV/2109.04760
[130] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. 2020. CycleISP: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2696–2705.
[131] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. 2020. Learning enriched features for real image restoration and enhancement. arXiv:cs.CV/2003.06792
[132] Georgi Zapryanov, Ivanova, and Iva Nikolova. 2012. Automatic white balance algorithms for digital still cameras - a comparative study. *Information Technologies and Control* 1 (01 2012), 16–22.
[133] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. 2017. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing* 26, 7 (2017), 3142–3155.
[134] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. 2011. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic Imaging* 20, 2 (2011), 1 – 17. https://doi.org/10.1117/1.3600632
[135] Molin Zhang, Junshen Xu, Yamin Arefeen, and Elfar Adalsteinsson. 2024. Zero-shot self-supervised joint temporal image and sensitivity map reconstruction via linear latent space. In *Medical Imaging with Deep Learning*. PMLR, 1713–1725.
[136] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. arXiv:cs.CV/1801.03924
[137] Xuaner Cecilia Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun. 2019. Zoom To learn, learn To Zoom. arXiv:cs.CV/1905.05169
[138] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. 2018. Image super-resolution using very deep residual channel attention networks. arXiv:cs.CV/1807.02758
[139] Zhilu Zhang, Haolin Wang, Ming Liu, Ruohao Wang, Jiawei Zhang, and Wangmeng Zuo. 2021. Learning RAW-to-sRGB mappings with inaccurately aligned supervision. arXiv:cs.CV/2108.08119
[140] Yu Zhu, Zhenyu Guo, Tian Liang, Xiangyu He, Chenghua Li, Cong Leng, Bo Jiang, Yifan Zhang, and Jian Cheng. 2020. EEDNet: Enhanced encoder-decoder network for AutoISP. In *Computer Vision – ECCV 2020 Workshops*, Adrien Bartoli and Andrea Fusiello (Eds.). Springer International Publishing, Cham, 171–184.

Appendices

A Tables

In this section, we provide all tables used for comparison (Tables 4, list of methods (Table 2), and other relevant information for the work.

Table 2. Summarization of the Approaches Considered in the Survey

Short Name	Description	Type
HerNet [78]	CNN that processes the reverse order of the CFA processing pipeline.	FIE
CameraNet [70]	Two stages network that divides ISP subtasks with poor correlation.	JD, FIE
Deep Camera [91]	An inverse ISP CNN to synthesize RAW images.	FIE
DRDN [83]	Residual learning and densely connected CNN for CFA Demosaicing.	Dm
Deep Demosaicing for Edge Implementation [89]	Deep learning-based demosaicing algorithms on low-end edge devices.	Dm
BayerUnify and BayerAug [72]	Different Bayer patterns unify and RAW image augmentation.	Dn, RIA
VisionISP [122]	ISP method to increase computer vision applications performance.	FIE
RLDD [34]	CNN that processes the reverse order of demosaicking and denoising.	FIE
DPN [62]	Quad Bayer CFA demosaicing network for submicron sensors.	Dm
CycleISP [130]	ISP network to produce realistic image pairs for denoising training.	Dn, RR
PyNET [49]	A novel pyramidal CNN to replace the mobile camera ISP.	FIE
PyNET-CA [61]	Improves PyNET performance by adding new modules.	FIE
SGNet [73]	An adaptive method for high and low image frequencies regions.	JD, RE
PatchNet and RestoreNet [108]	Active learning for the selection of the most useful image patches.	JD, RIA
AWNet [21]	ISP pipeline w/ wavelet transform and non-local attention.	FIE
Del-Net [35]	A multi-scale ISP network for smartphone deployment.	FIE
InvISP [124]	Neural network for the RAW data reconstruction and sRGB images rendering.	FIE, RR
ICDC-Net [113]	An approach with ISP-Net that addresses JPEG image compression.	FIE
CSANet [44]	Use of cascaded channel attention modules.	FIE
LiteISPNet [139]	Pairs of images captured by different cameras alignment.	RE, FIE
TENet [88]	Reordered the traditional ISP sequence of denoising, super-resolution, and demosaicing.	FIE
ReconfigISP [129]	General ISP networks parameters optimization for different tasks.	FIE
ISP Distillation [100]	Uses an sRGB image classification model and distills the knowledge of an ISP.	ID
Merging-ISP [12]	Reconstruct of multiple LDR image layers in just one HDR image.	LH
GCP-Net [105]	A joint denoising-demosaicking neural network for burst images.	JD
PIPNet [1]	Joint demosaicing and denoising network in CFA patterns.	JD
CURL [80]	Image enhancement for RAW-to-RGB and RGB-to-RAW mapping.	FIE
MicroISP [47]	An edge hardware focused CNN capable of processing up to 32MP RAW images.	FIE
PyNET-V2 Mobile [46]	A PyNET improvement to run on mobile devices.	FIE
OTST [26]	Transformer-based model for RAW-to-RGB using RGBW CFA pattern.	FIE

Legend for types: JD - Joint Denoising-Demosaicing, Dn - Denoising, Dm - Demosaicing, RE - Resolution Enhancement, RIA - Raw Image Augmentation, FIE - Entire ISP Enhancement, RR - RGB-to-Raw, ID - ISP Distillation, and LH - LDR to HDR.

B Software ISP

Over the last two decades, since the rise in popularity of embedded devices that use digital cameras as a secondary or main feature, the demand for reliable digital image capture and processing systems have grown significantly. Nowadays, processing speed and image quality are great selling points for most of those devices.

Table 3. Comparison of Training Schemes between DL Denoisers

Method	Dataset	Training scheme	Loss function	Augmentation procedure	Input format	Output format
BayerMethods [72]	SIDD [2]	Supervised	L1	Padding + Flip + Crop	Bayer	Bayer
CycleISP [130]	DND [112] + SIDD [2]	Custom	L1	Synthetic noisy image	Bayer sRGB	Bayer sRGB

Table 4. Comparison of Training Schemes between Deep Demosaicing Methods

Method	Dataset	Training scheme	Loss function	Augmentation procedure	Input format
DRDN [83]	DIV2K	Adversarial	MSE	Rotation + Flip	Bayer
DeepEdge [89]	Flickr500 adapted	Pareto frontier + Model search	CPSNR	n.a.	Bayer
PIPNet [1]	DIV2K + Flickr2K	Adversarial	L1 + RFL + PCL + Adversarial	n.a.	Quad-CFA Bayer
DPN [62]	Kodak, McMaster, HDR-VDP, Moiré, Urban100	Supervised	L2	Rotation + Flip	Quad-CFA Bayer
PatchNet and RestoreNet [108]	DIV2K	Supervised	L2	Learned patches selection	Bayer
GCP-Net [105]	Synthesized from Vimeo-90K	Supervised	Charbonnier penalty in lin-RGB + sRGB	n.a.	Bayer multi frame
SGNet [73]	Dense Texture Sparse Texture MIT moire Urban100 + DIV2K + Flickr2K	Supervised	Adaptive threshold edge + edge-aware smoothness (custom losses)	Crop	Bayer

Table 5. Comparison of Training Schemes between DL ISP Inverters

Method	Dataset	Training scheme	Loss function	Augmentation procedure	Target application
CycleISP [130]	DND + SIDD	Custom	L1	Synthetic noisy image	Denoising
InvISP [124]	MIT-FiveK	Supervised	L1 raw + L1 RGB	Rotation + Flip + Crop	Full ISP

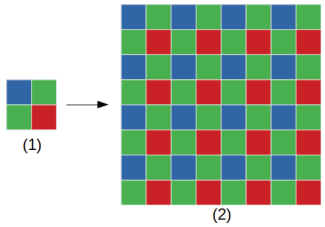

Fig. 3. (1) Bayer pattern and (2) extended pattern. Based on [120].

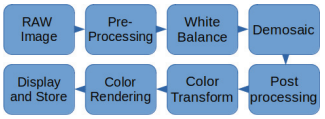

Fig. 4. Traditional ISP pipeline. Based on [59].

Traditionally, digital cameras are composed by the bond of two subsystems, the first one being dedicated to the acquisition of signals measured by a grid of photosensitive analog sensors, usually referred to as sensor element [90]. Modern sensor elements have high sensitivity to light variation but are unable to identify color variation on their own. A possible solution to this problem would be to use three distinct sensor elements, each with a specific filter to capture a certain frequency range of visible light.

This strategy would bring many technical issues, related primarily to sensor alignment, difference on light incidence, among others, in addition to an increase in hardware cost [90]. Modern sensor elements have a known pattern light frequency filter embedded into them, which makes it possible to reconstruct a color image with a single sensor element. These filters are known as **Color Filter Arrays (CFA)**. The **Bayer Color Filter (BCF)** is a special type of **Red-Green-Blue (RGB)** CFA pattern that is widely used in modern image sensors. Based on the human visual system, BCF consists of a 2 by 2 grid pattern containing two green, one red, and one blue sensor [90] as shown in Figure 3.

The BCF filter is placed right in front of the sensor element. The signal resulting from the capture process, filtered by the BCF is called **Bayer Array (BA)** and is composed of the monochromatic intensity of each pixel, following BCF pattern. RAW image files are composed by BA data in addition to metadata acquired at the time of capture, in this section, information such as capture time, preprocessing strategy, black level, aperture, exposure, ISO, among others, are usually encapsulated in standard **Exchangeable Image File (EXIF)** data. Modern image sensors, such as OmniVision's OV5647, uses a sensor element composed by a grid of 2624×1956 photosensitive sensors, covered by a layer of BCF. In addition to image data acquisition, this device also provides various processing options such as **Automatic Exposure Control (AEC)**, **Automatic White balance (AWB)**, **Automatic Band Filter (ABF)**, and Automatic Black Level Calibration.

An ISP pipeline is usually referred to as the second subsystem of a digital camera. Traditionally, ISP pipelines are constructed as a sequential series of operations. The input of an ISP is usually a RAW image and the output is a RGB encoded digital image. Commercially used ISPs may vary in order and type of operations, depending on the manufacturer needs. This information is not available, however, there are some basic operations that are necessary for most ISPs and are used

as a basis for the study of this type of subsystem. That said, there are known stages common to almost all traditional ISPs, these stages are shown in the Figure 4

Three stages are commonly mentioned in the context of ISP pre-processing: Signal conditioning, defective pixel correction, and black level offset. Signal conditioning refers to normalization, linearization, and other operations necessary to adapt the data obtained by the sensor to be processed by the ISP [90].

B.1 Black Level Offset Correction

Black level offset correction is a necessity created by the imprecision of image sensors, its goal is to correct the ISP input value to reduce black current effects, which tend to increase the light intensity measured by the sensor element and can cause an blur effect in a processed image [90]. The black level offset aims to ensure that the black tones contained in the image are correctly registered. Modern image sensors usually provides an array that contains a mask for black level correction [4].

B.2 Defective Pixels Correction

Defective Pixels are also common and expected acquisition errors up to a certain amount [128], they occur due to measurement issues caused by production errors, storage methods, and temperature problems. The identification of defective pixels is made from the analysis of the light intensity variation of a central pixel in relation to its neighbors [128].

B.3 White Balance

After the fixing acquisition issues, a usual first step of a conventional ISP is to perform a white balance on the imputed data. Although the HVS is able to identify the white color of objects illuminated by different types of light sources, digital systems do not have this capability, different frequency range of light results in different measured values [132]. White balance is a step that aims to ensure that the measured colors have a natural tone for the human eye after reconstruction [90]. The ratio between the average light intensity measured in the green channel and the others channels used as a basis to make the correction in all pixels [132].

B.4 Demosaicing

The heaviest computationally step of a digital image reconstruction refers to the conversion of CFA data to visible image, this process is called demosaicing. A recurrent strategy to perform this operation is by some variation of weight interpolation of the absolute values of each pixel in the CFA. Some open source applications like RawTherapee are transparent with the demosaicing technique used. In this application, visible image is reconstructed using algorithms.

Figure 5 shows the comparison between the ISP pipeline output (last image) of the smartphone Samsung G9600 and the RAW image sent by the image sensor (first image). Analyzing the metadata of the sensor output it was possible to identify that the CFA color pattern used in the capture process was the Bayer Green-Red-Blue-Green pattern. This information was used to perform a demosaicing (second image) on the original RAW. The comparison between the depicted images highlights the importance of all steps in an ISP pipeline in order to reconstruct high quality photos.

B.5 Denoising

Image denoising is a complex step of the digital image reconstruction task, where the goal is to remove the noise from an input to estimate the original image. This step is usually used in the traditional ISP pipeline because of defects or heterogeneity of the image sensor hardware components, and due to image compression. Image denoising is very important for several applications

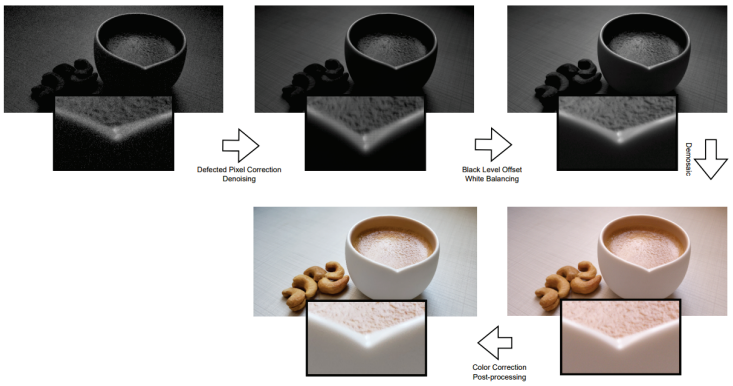

Fig. 5. Steps on a simple ISP pipeline.

in the vision computing field and has received a lot of attention over the years [10, 25, 81, 95, 110]. Several algorithms have been proposed for image denoising. These methods can be classified into classical approaches and deep learning approaches. Classical approaches encompass the spatial domain method, which applies filters in the image to remove the noise, and the transform domain method, which changes the domain from the input image and then uses a denoising procedure to improve the image. The deep learning approach, in most cases, is a CNN-based method. In this survey, some works of deep learning for image denoising will be cited [4, 9, 30, 72, 73, 109].

B.6 Deblurring

Blur is a general artifact that is hard to avoid in digital image processing and can be caused by various sources like motion blur, out-of-focus, camera shake, extreme light intensity, and others. Given this, many handcraft deblurring algorithms exist in the literature to mitigate this problem [64, 115] and some are included in ISPs. [92] proposes a deblurring method for cell phones that takes into consideration the brightness and the contrast to correct the blurred image. However, many of these classical methods englobe only some cases, realize handcraft feature extraction and are necessary two previous steps [64]: blur detection and blur classification.

B.7 Post-processing Step

Each camera manufacturer can use different, often proprietary, processing methods to improve image quality. The post-processing step aims to make some adjustments to the images that went through the previous processes. Some of the most common post-processing steps used are edge enhancement, removal of colored artifacts, and coring [90]. These techniques use heuristics and require considerable fine-tuning.

For example, the demosaicing step can introduce undesirable artifacts, such as zippered edges and confetti. In the post-processing stage, it is essential to keep these artifacts to a minimum without losing image sharpness [90]. Some camera manufacturers use edge enhancement techniques to make the image more attractive by reducing low-frequency objects contained in the image. The solution to these problems involves many variables, from the size of the capture sensor to the demosaicing technique used.

B.8 Rendered Color Spaces

Rendered color spaces are generally used as output and have a limited scale, unlike unrendered which are based on scenes. The rendered color space is typically 8 buts and it is computed from

raw sensor data data, which typically is between 12 and 16b [90]. It means that transforming unrendered to rendered space implies in dynamic range loss.

The most common rendered space is the sRGB [90] color space, which has become common for multimedia. Another common rendering space is the ITU-R BT.709-3, which was created with high-definition televisions in mind. The sRGB standard adopts the primaries defined by ITU-R BT.709-3. It is these patterns that define the methods of transforming unrendered spaces to values of 8b imposed by most output media.

Regarding storage, two groups of solutions exist. Professional cameras with large sets of sensors and much storage generally use a proprietary format or **Tag Image File Format/Electronic Photography (TIFF/ EP)**. On the other hand, JPEG2000 and JPEG offer much higher compression rates. JPEG2000 offers more efficient compression than the standard JPEG, in addition to offering several features such as control over data compression and image resolution. Despite its benefits, its computational complexity, high memory cost, and lack of support in common devices limit its usage.

Given the high complexity and need for tuning of modern ISP pipelines, many studies are being made aiming to use machine learning to convert RAW image data into high quality outputs. Hence, the present work presents the latest advancements on the field.

C Methodology

This section describes how we made a qualitative comparison among the works covered in this survey and the analysis of these papers to highlight points of improvement, highlights, and ways to evolve this field of study.

C.1 Datasets

For the quantitative evaluation, we provide a comparison among the works covered in this survey with the more explored datasets. To do this, we use the results provided by these works and the most commonly used metrics in the image restoration task, PSRN and SSIM, as the base for comparison. We provide a brief discussion on each most commonly used dataset on the studies we analyzed. Table 7 gives a list of all datasets discussed in this section with details about the number of images, and size.

C.1.1 Zurich RAW to RGB (ZRR). ZRR dataset was proposed by Ignatov et al. [50] in order to get a large-scale real-world dataset, that deals with the task of converting original RAW photos captured by smartphone cameras to superior quality images achieved by a professional DSLR camera. The proposed database is publicly available and amounts to 22 GB, containing 20K real images captured synchronously by a Canon 5D Mark IV DSLR camera and Huawei P20 phone, in a variety of places and in various illumination and weather conditions. ZRR was a recurrent dataset choice for some RAW to RGB mapping problem works considered in this survey [21, 35, 49, 61, 78, 139].

C.1.2 Urban 100. The Urban 100 dataset consists of 100 high-resolution images with an assortment of real-world urban scenarios and structures. Urban 100 was constructed with synthetic images, under Creative Commons license, resulting in a 1.14 GB dataset. It is a well-known public database for super-resolution tasks [1, 34, 62, 62, 73, 108].

C.1.3 McMaster Dataset. [134] consists of 18 sub-images of size 500x500 captured by Kodak film and then digitized. The sub-images were cropped from eight high-resolution natural images with size 2310x1814. The McMaster dataset is used for color demosaicing in some of the articles in this survey [1, 34, 62, 83, 88].

C.1.4 Kodak. It is a little dataset composed of 24 photographic quality images of size 768x512 or 512x768 with a large variety of locations and lighting conditions. This dataset contains raw images in **photo-cd (PCD)** format and PNG format with 24 bits per pixel. Besides, many works use the Kodak dataset for compression tests and to validate methods that do tasks like demosaicking, denoising, and full ISP pipeline [1, 34, 62, 83, 88].

C.2 Papers Analysis

The analysis of the works in some fields of study is fundamental to discover new ways to its evolution and improvement in future works. In this survey, the papers are analyzed about the following points:

— **Details of method:** The analysis of details of many methods can bring new ideas and identification of problems that future works can propose to solve.
— **Used datasets:** The camera hardware has many nuances and generates your own noise that is hard to simulate. Then the use of an appropriate dataset to train and validate the work is an important part to consider in creating a new ISP method.
— **Preoccupation with computational cost:** The computational cost is an important point to consider in almost all applications of ISP, mainly used in embedded systems and mobile devices.
— **Method evaluation:** How the method was evaluated may be well planned to indicate the contribution of this work in a determined study area.

D Neural Networks Architectures

It is relevant to understand how each architecture works to elucidate how to improve results. This appendix shows the architecture of the most relevant works that disclose this information.

D.1 U-Net

Figure 6 shows the U-Net architecture.

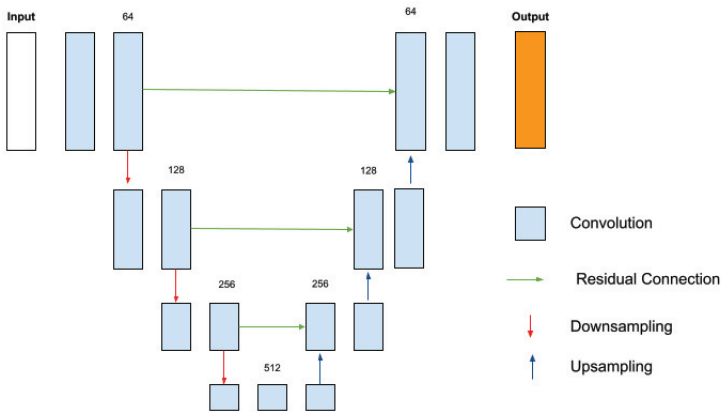

Fig. 6. The U-Net architecture.

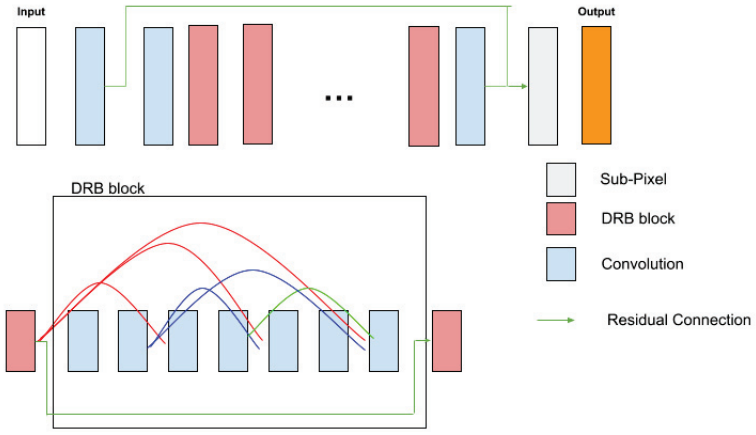

Fig. 7. The U-Net architecture.

D.2 DRDN

Figure 7 shows the DRDN architecture.

D.3 DPN

Figure 8 shows the DPN architecture.

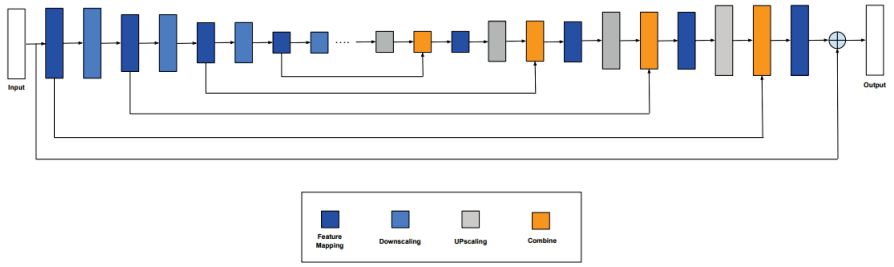

Fig. 8. The DPN architecture.

D.4 CSANet

Figure 9 shows the CSANet architecture.

D.5 Deep Camera

Figure 10 shows the DeepCamera architecture.

D.6 HERN

Figure 11 shows the HERN architecture.

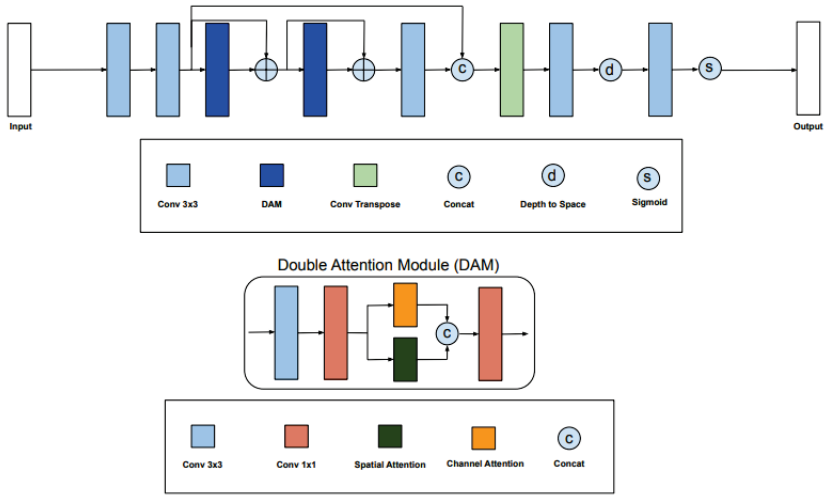

Fig. 9. The CSANet architecture.

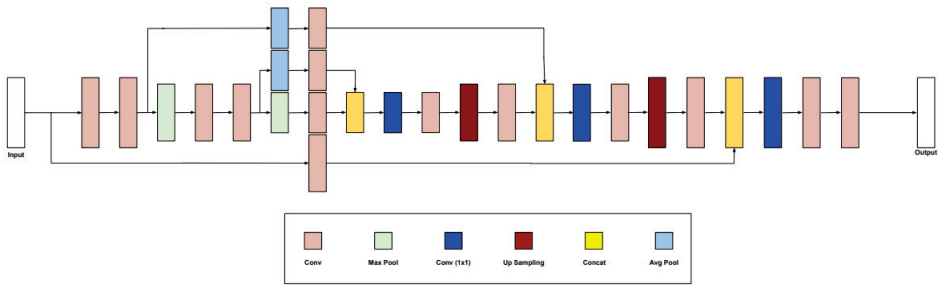

Fig. 10. The Deep Camera architecture.

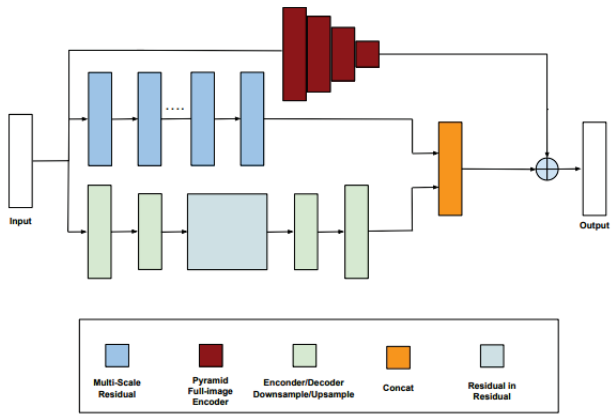

Fig. 11. The HERN architecture.

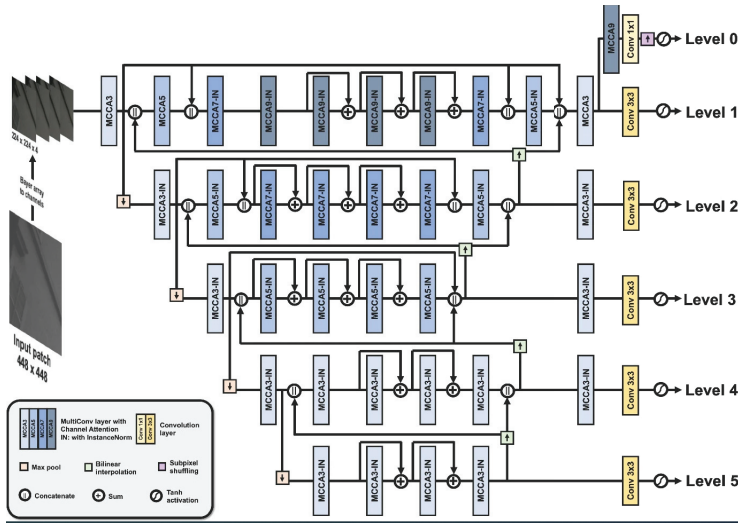

Fig. 12. The PyNet-CA architecture.

D.7 PyNet-CA

Figure 12 shows the PyNet-CA architecture.

Table 6. Comparison of Training Schemes between Entire ISP Enhancement Models

Method	Dataset	Training scheme	Loss function	Augmentation procedure	Image format
HERNet [78]	ZRR dataset [49]	Supervised	L1	Horizontal and vertical flips	Bayer
CameraNet [70]	HDR+ [37] FiveK [11] SID [13]	Supervised	L1	Random rotations, horizontal and vertical flips	sRGB
Deep Camera [91]	Ciurea and Funt [18]	Supervised	L1	None	Bayer
VisionISP [122]	KITTI 2D [3]	Supervised	-	None	sRGB
RLDD [34]	Kodak [28] McMaster [134] Urban 100 [42]	Supervised	MSE	180° rotation, horizontal and vertical flips	Bayer
PyNET [49]	ZRR dataset [49]	Supervised	MSE VGG SSIM	None	Bayer
AWNet [21]	ZRR dataset [49]	Supervised	Pixel Perceptual SSIM Multiscale	horizontal and vertical flips	Bayer
PyNET-CA [61]	ZRR dataset [49]	Supervised	MSE Perceptual Multiscale	90° rotation, horizontal and vertical flips	Bayer
Del-Net [35]	ZRR dataset [49]	Supervised	SSIM Perceptual	horizontal and vertical flips	Bayer
InvISP [124]	FiveK [11]	Supervised	L1	Random rotation, random crop and random flip	sRGB
CSANet [40]	Mobile AI 2021	Supervised	Charbonnier SSIM Perceptual	Random flip	Bayer
LiteISPNet [139]	ZRR dataset [49] SR-RAW [137]	Supervised	L1 Perceptual	90° rotation, horizontal and vertical flips	sRGB
TENet [88]	Kodak [28] McMaster [134] Urban 100 [42] BSD100 [77]	Supervised	L2-Norm	Random rotation Random flip	Bayer
ReconfigISP [129]	S7 ISP Dataset [102] SID [13]	Supervised	L1 L2	Random crop Random flip	Bayer
CURL [80]	S7 ISP Dataset [102]	Supervised	CURL	None	Bayer
PyNet-V2 Mobile [46]	S7 ISP Dataset [102]	Supervised	CURL	None	Bayer
OTST [26]	S7 ISP Dataset [102]	Supervised	CURL	None	Bayer
MicroISP [47]	S7 ISP Dataset [102]	Supervised	CURL	None	Bayer

Table 7. Summarization of Datasets Considered in the Survey and their Respective Number of Images, and Size

Dataset name	**N° images**	**Size**
Zurich RAW to RGB [50]	20.000	≈ 22 GB
Urban 100 [42]	100	1.14 GB
McMaster dataset [134]	18	13.6 MB
Kodak	25	119.7 MB

Received 10 December 2021; revised 4 December 2024; accepted 7 December 2024