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

**Matheus Henrique Marques da Silva, Jhessica Victoria Santos da Silva, Rodrigo Reis Arrais**
Eldorado Research Institute
Campinas, São Paulo - Brazil
{matheus.marques, jhessica.silva, rodrigo.arrais}@eldorado.org.br

**Wladimir Barroso Guedes de Araújo Neto, Leonardo Tadeu Lopes, Guilherme Augusto Bileki**
Eldorado Research Institute
Campinas, São Paulo - Brazil
{wladimir.neto, leonardo.lopes, bilekig,}@eldorado.org.br

**Iago Oliveira Lima, Lucas Borges Rondon, Bruno Melo de Souza**
Eldorado Research Institute
Campinas, São Paulo - Brazil
{iagolima, lucas.rondon, brunobms}@eldorado.org.br

**Mayara Costa Regazio, Rodolfo Coelho Dalapicola, Claudio Filipi Gonçalves dos Santos**
Eldorado Research Institute
Campinas, São Paulo - Brazil
{mayara.regazio, rodolfo.dalapicola, claudio.santos}@eldorado.org.br

## Abstract

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 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.

***K*eywords** image signal processing, deep learning, convolutional neural networks

# 1 Introduction

The Image Signal Processor (ISP) is a component of digital cameras capable of performing various tasks to improve image quality, as demosaicing, denoising, and white balance. The set of tasks performed by the ISP is called ISP pipeline, divided in preproccessing and postprocessing steps, and may differ from manufacturer to manufacturer [1]. Nowadays, Machine Learning is used to replace partially or the entire ISP pipeline. Particulary, Deep Learning is employed to replace ISP tasks, working on noise removal or some image feaure 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, 27 research papers, that implemented Deep Learning based ISP pipeline.

## 1.1 Image Signal Processing

Traditionally, ISPs are digital signal processors that reconstruct RGB images from RAW images. In traditional camera pipelines, complex and proprietary hardware processes are used to perform image signal processing [2]. 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 [3]. 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 [2], so numerous parameters need to be adjusted. Moreover, multiple ISP processes executed sequentially with module-based algorithms lead to cumulative errors at each execution step. To minimize those errors, new techniques were 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[4], it was only in the last decade 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 [5], Medicine [6], Economics [7] and Physics [8], have been able to 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 that use several layers of neurons to extract higher level features from the raw data being provided to it [9, 10, 11]. This class of algorithms requires a huge amount of computational power that 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 being 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.

One of the fields that found great results with the use of Deep Learning is the Image Processing (a sub set of Computer Vision), more specifically with the use of Convolutional Neural Networks (CNN). The CNNs are a class of Neural Networks more prone to work with visual imagery for being inspired by biological processes, created in a manner that the connective pattern of the neurons imitates the pattern of the visual cortex of animals [12, 13]. Another important aspect of the CNNs is the fact that, following a different path from other networks, they use very minimal pre-processing, being able to learn by themselves to optimize kernels. These features make the use of CNNs more common for Image Processing tasks.

## 1.3 The ISP and Deep Learning Relation

The intention of using a CNN to replace the hardware-based ISP is justified by the fact that a CNN can compensate 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 [1]. [31] was one of the first to propose a CNN 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.

Network CNNs not only show significant advantages in low-level vision tasks [2], they also show good results in high-level tasks such as object detection and segmentation [14]. 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. [3] showed the difficulties in performing the necessary adjustments in traditional ISP pipelines and developed a CNN that performs the ISP pipeline.

### 1.4 Comparison with Other Works

One of the great ways to consolidate a growing field in science is to perform surveys about the most current techniques in that field. This way, the access for this kind of information is facilitated, making it easier to understand and choose which technique to use for one's individual case. In the field of Deep Learning, for example, a great number of surveys is already available in many different areas, such as agriculture [15], cyber security [16], autonomous driving [17], medical imaging [18], and also on more technical areas, such as on CNN [19]. 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.

While there are already some surveys about individual steps of an ISP, such as demosaicing [20] and denoising [21], there is still no easy way to find and compare end-to-end ISP deep learning approaches. In this paper we summarize many of these approaches, bringing some of the state-of-the-art Artificial Neural Networks (ANNs) in this field.

### 1.5 Scope of this work

For this study, the articles were studied according to three 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 2021, making this study very up-to-date.

Table 1: Summarization of the approaches considered in the survey.

| **Short Name** | **Description** |
|---|---|
| HerNet [22] | Combined CNN's with traditional algorithms to reverse the order of the CFA processing pipeline. |
| CameraNet [23] | Divides the subtasks with poor correlation, creating a network with two stages. |
| Deep Camera [3] | Create a network with four parallel paths with convolutional layers and an inverse ISP to synthesize RAW images. |
| DRDN [24] | Color filter array demosaicking based on residual learning and densely connected convolutional neural network. |
| Deep Demosaicing for Edge Implementation [25] | Discussed the edge implementation of deep learning-based demosaicing algorithms on low-end edge devices |
| BayerUnify and BayerAug [26] | Create a method to unify different Bayer patterns and an effective approach for raw image augmentation. |
| VisionISP[27] | ISP method to increase computer vision applications performance. |
| RLDD [28] | Combined CNN's with traditional algorithms to reverse the order of demosaicking and denoising. |
| DPN [29] | An efficient deep neural network architecture for Quad Bayer CFA demosaicing adopted in submicron sensors. |
| CycleISP [30] | Models camera imaging pipeline in forward and reverse directions, producing realistic image pairs for denoising. |
| PyNET [31] | Uses a novel pyramidal CNN architecture to replace the mobile camera ISP. |
| PyNET-CA [32] | Improves PyNET performance by adding channel attention and subpixel reconstruction modules. |
| SGNet [33] | Created an adaptative method to treat regions with high and low frequencies. |

| | |
|---|---|
| PatchNet and RestoreNet [34] | Select the most useful patches from an image for the training step using active learning. |
| AWNet [35] | Use of wavelet transform and non-local attention mechanism in ISP Pipeline. |
| Del-Net [36] | A multi-scale architecture that learns the entire ISP pipeline. Ideal for smartphone deployment. |
| InvISP [37] | Reconstruct the RAW data, in addition, to rendering sRGB images using an invertible neural network. |
| ICDC-Net [38] | An approach with ISP-Net that addresses JPEG image compression in network training. |
| CSANet [39] | Uses cascaded blocks composed of channel attention modules. |
| LiteISPNet [40] | Method to align pairs of images captured by different cameras during the train. |
| TENet [41] | Reordered the traditional operation sequence to denoising + super-resolution -> demosaicing. |
| ReconfigISP [42] | Adapted the architecture and parameters according to a specific task. |
| ISP Distillation [43] | Uses an sRGB image classification model and distills the knowledge of an ISP pipeline. |
| Merging-ISP [44] | Approach to reconstructing multiple image layers LDR in just one image HDR. |
| GCP-Net [45] | A joint denoising and demosaicking method for real-world burst images, based on a green channel prior neural network. |
| PIPNet [46] | Uses a deep network to work with joint demosaicing and denoising in CFA patterns. |
| CURL [47] | Image enhancement method that can be used in the RAW-to-RGB and RGB-to-RAW mapping tasks. |

Those considered papers did not seek the same ISP tasks and improvements. Figure 1 shows the ISP studied functions distribution among the reviewed articles and Table 1 gives a list of all methods discussed in this work.

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.

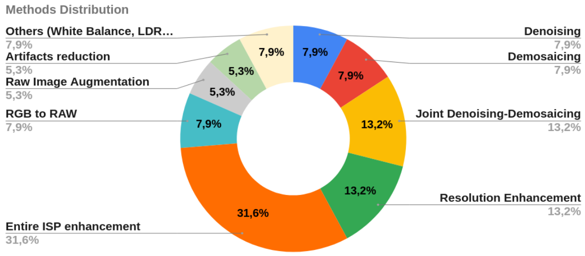

Figure 1: Reviewed ISP tasks distribution

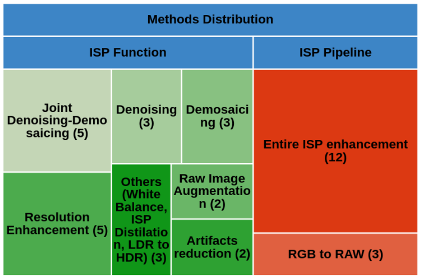

Figure 2: Mapped ISP tasks

## 1.6 Work structure

The rest of this work is structure as follows: Section 2 describes the overall concept, traditional pipelines and algorithms for ISP. Section 3 contains the resumes for the works collected through this research. Section 4 covers the results of the collected works. Section 5 describes in detail the methodology collected during this research. Section 6 concludes this work.

## 2 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. That being said, studies of image reconstruction systems have never been so important.

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 [1]. 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 3 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 [1]. 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. The BCF is build on the assumption that the human visual system (HVS) has more sensitivity to colors on the green spectrum. Based on that BCF consists of a 2 by 2 grid pattern containing two green, one red and one blue sensor [1] as shown in Figure 3.

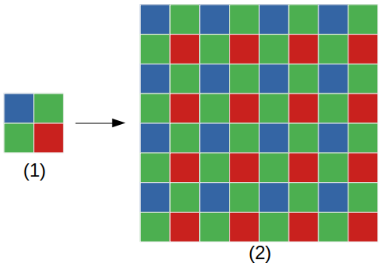

Figure 3: (1) Bayer Pattern and (2) extended pattern. Based on [48]

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, pre-processing 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, to provide a RAW output with better overall image quality[4].

An ISP pipeline usually referred as the second subsystem of a digital camera. It is build on a series of processes that aim to convert a RAW image file into a visible digital object, in order to display and store the image that was captured. Some of the steps of a traditional pipeline will be explored here in general lines.

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 below.

Although modern image sensors are capable to fix many data acquisition issues[4], a pre-processing of the data contained in the RAW image is important to verify signal integrity, identify and fix acquisition failures, preventing errors from being propagated through the following operations. Three stages are commonly mentioned in the context of ISP pre-processing: Signal conditioning, defective pixel correction, and black

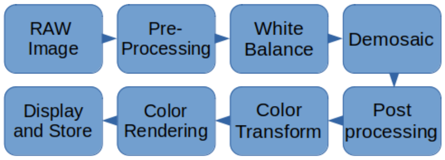

Figure 4: Traditional ISP pipeline. Based on [49]

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 [1].

### 2.1 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 [1]. 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].

### 2.2 Defective Pixels Correction

Defective Pixels are also common and expected acquisition errors up to a certain amount [50], they occur due to measurement issues caused by production errors, storage methods and temperature problems. The identification of defective pixels is made, in general, from the analysis of the light intensity variation of a central pixel in relation to its neighbors [50]. There are several methods of correction of defective pixels, one of them is to apply the average value of the neighboring points to the pixel identified as defective [50].

### 2.3 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 [51]. White balance is a step that aims to ensure that the measured colors have a natural tone for the human eye after reconstruction [1]. A strategy often used by ISPs is the AWB strategy, while many image sensors already have this feature built in as in OmniVision's OV5647, many ISPs implement it separately. A usual way of doing this is using the gray world assumption, which dictates that the average color of each sampled channel tends to be equal in most cases. Based on this principle, 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 [51].

### 2.4 Demosaicing

The most computationally heavy 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. Although most of the demosaicing techniques used commercially are protected by patent. Some open source applications like RawTherapee are transparent with the demosaicing technique used. In this application, visible image is reconstructed using algorithms such as Adaptive Homogeneity-Directed [52].

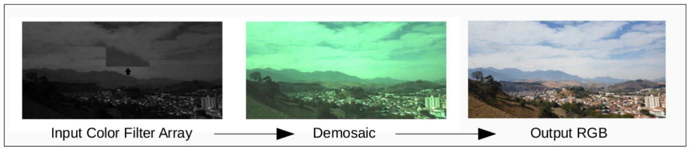

Figure 5: Demosaicing a CFA.

Figure 5 shows the comparison between the ISP pipeline output (third 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 second and third images highlights the importance of all steps in an ISP pipeline in order to reconstruct high quality photos.

### 2.5 Denoising

Image denoising is a complex step of the digital image reconstruction task, where the goal is to remove the noise from an input image 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 in the vision computing field and has received a lot of attention over the years [53, 54, 55, 21, 56]. Several methods have been proposed for image denoising. They can be classified into classical approaches and deep learning approaches. The 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 [57, 26, 58, 33, 59, 60].

### 2.6 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, accumulated error in ISP pipeline, etc. Given this, many handcraft deblurring algorithms exist in the literature to mitigate this problem[61, 62] and some are included in ISPs. The Razligh and Kehtarnavaz [63] proposed a deblurring method for cell phones that takes into consideration the brightness and the contrast to correct the blurred image. On the other hand, Hu et al. [64] consider the use of smartphone's inertial sensors, gyroscope and accelerometer, for kernel estimation and utilize an online calibration to synchronize cameras and sensors. However, many of these classical methods englobe only some cases, realize handcraft feature extraction and are necessary two previous steps[61]: blur detection and blur classification. Deep learning was an alternative to solving these problems in the last years, possibly jointly all these steps on a unique step[65].

### 2.7 Post-processing step

Each camera manufacturers 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 [1]. These techniques use heuristics and require considerable fine-tuning.

For example, the demosaicing step can introduce artifacts that can be problematic, such as zippered edges and confetti in the rest of the image. In the post-processing stage, it is essential to keep these artifacts to a minimum without losing image sharpness [1], 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.

### 2.8 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 created based on data extracted from an image in unrendered space and contains a maximum of 8b while the unrendered space has a variable range between 12 and 16b [1], for this reason, the process of transforming unrendered space to rendered space contains a loss in dynamic range.

The most common rendered space is the sRGB [1] 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.

Depending on the preview mode, images need to be converted to proper color space. An example is in the case of viewing by a CRT monitor with an additive color system, images need to be converted to an 8b output given the display model used, offset values, color temperature, and values of gamma.

When the goal is storage, we have two solutions, professional cameras that have a very large set of sensors and much larger storage space, and generally store the images in a proprietary format or as Tag Image File Format/Electronic Photography (TIFF/ EP). Images stored as TIFF/EP have additional information such as camera settings details and the color transformation matrix [1]. JPEG2000 is an international standard that offers a more efficient compression than the common JPEG standard, in addition to offering several features such as control over data compression size and image resolution. However, despite the benefits that JPEG2000 presents, its computational complexity, and high memory cost are limiting factors.

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. This work shows state of the art of these studies.

## 3 Know Approaches

In this section, we review the applications of deep learning-based methods to substitute the traditional handcraft ISP pipelines, where include operations as demosaicing, denoising, white balance, tone adjustment, and exposure balance. Also,we briefly summarize some works that use this approach to increase the performance in other computer vision tasks.

### 3.1 HERNet

HighEr-Resolution Network (HERNet) [22] 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 [22] that realizes 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)[66] that have two convolutional layers with 3x3 and 5x5 kernels in parallel. Furthermore, in the global information path, was applied an Autoencoder mechanism with modified Residual in Residual (RIR)[67] modules for feature extraction. The modified RIRs modules aim to decrease the GPU memory usage, especially to high-resolution images, then the Channel Attention Units were removed, and stacked the remaining. Finally, the authors trained and validated the model with the ZRR dataset[31] and used only the L1 loss in training. Unfortunately, the L1 loss favors blurry images in datasets with misalignment.

The local information path consists of Multi-Scale Residual Blocks (MSRBs)[66] that have two convolutional layers with 3x3 and 5x5 kernels in parallel. Furthermore, in the global information path, is apply an Autoencoder mechanism with modified Residual in Residual (RIR)[67] modules for feature extraction. The modified RIRs modules aim to decrease the GPU memory usage, especially to high-resolution images, then the Channel Attention Units are removed and stacked the remaining. The authors trained and validated the model with the ZRR dataset[31] and used only the L1 loss in training. Unfortunately, the L1 loss favors blurry images in datasets with misalignment.

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. 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.

### 3.2 CameraNet

The CameraNet[23] 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+[68] and FiveK[69] using Adobe Camera Raw[1] and Adobe Lightroom [2]. Each ground truth was used to train a different stage. In the CameraNet pipeline, before these two main stages, the input image is pre-processed with the bad pixels removing, initial demosaicing with interpolation, and converting RGB to CIE XYZ space because it is related 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[70] Network in the HDR+ and, mainly, SID[71] dataset. The explanation for this difference can be the high level of noise in the SID dataset and the separation of weakly related subtasks in two stages in the CameraNet. In the FiveK dataset, both methods achieve comparables results in SSIM, but the CameraNet obtain superior results in PSNR and Color Error indices because this dataset was captured with high-end cameras, reducing the noise level. DCRAW and CameraRAW, which generate images with default settings, have the lowest results in comparison with Deep Learning methods, explained by the limitation of traditional methods.

### 3.3 Deep Camera

As the first CNN Network proposed to substitute the entire ISP pipeline, the Deep Camera[3] 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, the authors created an inverse ISP to recreate RAW images from a large dataset [72] 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, the 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.

### 3.4 DRDN

The DRDN [24] proposed 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 mosaicked input images. It also solved the vanishing-gradient problem experienced by many deep neural networks, due to the residual learning [73] and densely connected convolutional neural network [74]. Moreover, the proposed model applied block-wise convolutional neural networks to consider local features and a sub-pixel interpolation layer, generating demosaiced output images more efficiently and accurately.

This paper has an exceptional explanation and contextualization about the demosaicing challenging task, citing similar previous methods and highlighting what could be improved in each one. The study detailed the training parameters and aspects related to the considered Datasets. Finally, there is a vast performance comparison, in which the proposed method stood out in the vast majority. On the other hand, the paper did not reveal the inference time obtained during the validation phase, considering that the studied datasets have medium size images. Likewise, the authors could discuss deeper the reason that the DRDN did not reach the best PSNR in some evaluated cases [24].

### 3.5 Deep Demosaicing for Edge Implementation

In this paper [25], the authors discussed the edge implementation of deep learning-based demosaicing algorithms on low-end edge devices major challenge. 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 contributed with a valuable reference frame about demosaicing methods, divided into six categories: Edge-sensitive methods, Directional

[1] https://helpx.adobe.com/br/camera-raw/using/supported-cameras.html

[2] https://www.adobe.com/lightroom

interpolation and decision methods, Frequency domain approaches, Wavelet-based methods, Statistical reconstruction techniques, and Deep learning-based methods [75]. Likewise, the 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 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. As a drawback, the authors should have discussed more the use of the given architecture search in other image processing tasks, not only in the edge implementation challenge.

### 3.6 BayerUnify and BayerAug

Liu Et al. [26] presented two new techniques for DNN-based RAW image denoising. The first is a Bayer pattern unification (BayerUnify) method, that effectively deals with varied Bayer patterns from different data sources. The second method is the Bayer preserving augmentation (BayerAug), allowing proper RAW images augmentation. Associating these two techniques with a modified U-Net, the proposed method achieved a satisfactory SOTA PSNR of 52.11 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.

Traditional data augmentation methods are frequently based on image flipping or cropping. However, for RAW images, the flipping procedure may affect the Bayer pattern. BayerAug sorts out this question, combining both flipping and cropping. The paper presented three different flipping methods, which enabled data augmentation on Bayer RAW images without any issues.

The study evaluated the proposed method on the Smartphone Image Denoising Dataset (SIDD) [76] - 320 pairs of noisy and noise-free images that covered three different Bayer patterns. The networks were trained with L1 loss, AdamW [77] optimizer, and 200,000 iterations.

The paper presented a satisfactory introduction about the Bayer pattern pre-processing and augmentation, besides discussing related works. The proposed method is properly explained, as well as the network architecture and an exceptional training detailing. At that frame reference, the study shown a promising direction on RAW image processing with deep learning techniques. As a limitation, the authors could give a comparison to other works in the same NTIRE 2019 challenge, and a deeper discussion about the application of the proposed method in real devices.

### 3.7 VisionISP

Wu Et al. [27] proposed a particular ISP method for computer vision applications. VisionISP reduced 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. The study adopted the technique presented by Nishimura [78] to optimize the denoising parameters, constructing the denoiser for the computer vision task, not for image quality. The paper also highlighted that the demosaicing step can be skipped and the color filter array image usage, instead of a demosaiced image, 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 [79], 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 [80] framework and its original code.

The paper provided a proper explanation and evaluation of each VisionISP component. As a positive aspect, the experiments show that once TVS is trained, it can be used with other computer vision systems. VisionISP as a whole can be trained jointly or separately with the computer vision backbone. Besides that, each component of the proposed framework enhances a computer vision engine performance and can be deployed independently.

Nevertheless, the paper could provide more training details (e.g., epochs number, time inference, hardware, etc.) and a deeper comparison of the VisionISP effectiveness with other computer vision optimization systems at that time.

### 3.8 RLDD

In this paper [28], the authors combined convolutional neural networks with traditional algorithms to reverse the order of the traditional CFA pipeline (demosaicking and denoising). The method, which we called here RLDD, uses two stages for demosaicking-denoising. The first stage performed the demosaicking by composing the gradient-based threshold-free (GBTF) method [81] and a convolutional neural network to overcome the reduction of image resolution in the subsampling operation. The second stage performed the denoising using another convolutional neural network whose goal was 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 [82], McMaster [83], and Urban 100 [84] datasets and have shown that this model outperforms state-of-the-art demosaicking and joint demosaicking 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, demonstrating a better visual quality of the images. However, the average running time of demosaicking and denoising of this method did not surpass all the evaluated methods.

### 3.9 DPN

This study proposed a duplex pyramid network (DPN [29]), an efficient deep neural network architecture for Quad Bayer CFA demosaicing adopted in submicron sensors.

The article delivers an accurate background in respect to the ISP state-of-the-art challenge and a helpful Quad Bayer CFA Analysis, referencing related Deep Learning based solutions.

The proposed architecture consisted of two connected feature map pyramids. One of them is composed of downscaling blocks and the other is composed of upscaling blocks, combined with dense skip connections.

As well as the skip connections, inspired by U-Net [85], the given network also applied residual learning, inspired by ResNet [73]. DPN also implemented a Linear Feature Map Growth. Compared to the traditional exponential method, this linear method led to a fewer number of parameters, which is more precise for mobile applications with limited memory constraints.

In the results topic, the study brought a comparison against conventional ISP algorithm implemented in 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. Otherwise, the input must be cropped before the downscaling blocks. Furthermore, the study could have done tests with larger resolution images which are also captured in mobile phones of that 2019 frame reference, like FullHD and 4K images.

### 3.10 CycleISP

Zamir [30] Et al. proposed 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, an auxiliary color correction network branch, and a noise injection module. 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 mosaicked 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 2x2 block packaging into four channels (RGGB) and an image resolution reduction block. To ensure that the input RAW data may come from any camera and have different Bayer patterns, the RAW2RGB branch applies the Bayer pattern unification technique [26]. Next, a convolutional layer and a proposed recursive residual group encode the packed RAW image into a deep feature tensor. 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 that provides optimal-quality images. For this, the RGB2RAW's output becomes the RAW2RGB's input, and the 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. For the sRGB space, the CycleISP model is fine-tuned with the SIDD [76] dataset, which contains clean and noisy image pairs in both RAW and sRGB spaces.

For the training stage, the authors used the MIT-Adobe FiveK dataset [69], followed by the Fine-Tuning. They evaluated CycleISP performance with state-of-the-art RAW and sRGB denoising methods, using the DND [86] and SIDD [76] benchmarks. CycleISP achieved better results in both scenarios, with almost a 5 times smaller network parameters number than the previous best RAW denoising method [58] and performance gain against the previous best sRGB denoising algorithm [57]. Besides that, in contrast to the other evaluated models, the proposed method provided clean and artifact-free results, also preserving image details. The proposed framework modules were appropriately described and reinforced with an ablation study. The authors provide a decent implementation details section, a suitable comparison with related methods, and a solid generalization capability study. As a negative aspect, the paper could show more details at the ablation study, giving more information about individuals contributions of the other CycleISP modules aside from the RAW2RGB branch.

### 3.11 PyNET

PyNET [31] is a novel pyramidal CNN architecture designed to replace the entire ISP pipeline present 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 with the trained output being used in the above level training stage. The convolutional filters size in this method varies from 3x3 at level five up to 9x9 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, four and five, are trained with mean squared error (MSE) to learn global color and brightness correction. Levels two and three are trained by a combination of MSE and VGG-based [87] to refine the color and shape of objects. Finally, level one is trained with MSE, VGG, and SSIM loss [88] and performs corrections in the local color processing, noise removal, texture enhancement, etc.

Additionally, the authors present the Zurich RAW to RGB dataset, composed of 20 thousand 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 are conducted. The first, compared PyNET with the methods SPADE [89], DPED [90], U-Net [85], Pix2Pix [91], SRGAN [92], VDSR [93], and SRCNN [94]. 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[3] 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. Finally, the last experiment tested the PyNET using RAW images captured by the BlackBerry KeyOne smartphone without re-training the network. The image produced by PyNET was compared with the image produced by BlackBerry's ISP, and PyNET generated good results, however, the results were not ideal in terms of exposure and sharpness.

### 3.12 SGNet

Many methods have joined highly correlated tasks have success, decreasing the accumulated error in the several process units in the ISP pipeline. Thus, the SGNet[33] 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 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[59], CDM[95], Kokkinos[96], Deepjoint[4] [60], and FlexISP[97], the SGNet can remove moiré artifacts more effectively and give images with more definition in high-frequency areas than ADMN and Deepjoint. However, as a negative point, this work did not show a preoccupation with computational efficiency, which is essential to applications that use demosaicing and denoising.

### 3.13 PatchNet and RestoreNet

In this paper [34], the authors proposed a method based on active learning (data-driven) that learns to select the most suitable patches from an image for the training step without adding additional cost to the inference step. 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 of 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, the authors also proposed RestoreNet, an architecture that applies the structural knowledge extracted from PatchNet and is responsible for restoring the original image.

The results were validated on the Vimeo-90k [98], MIT Moire[60], and Urban 100 [84] datasets and compared with the Kokkinos [96], SGNet [33], CDM [95], and DeepJDD[4] [60] methods. The methods were compared at three different noise levels (5, 15, 25). In all comparisons, the proposed method achieved better PSRN values in the JDD task.

The ablation studies analyzed the effects of different patch sizes when PatchNet is evaluated on Demosaicing and shown that performance improves as patch size increases. It would be interesting to study the computational costs involved in increasing patch sizes and how much this would change the network's complexity. 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, but it would have been interesting to evaluate other metrics.

### 3.14 AWNet

In this work [35], the researchers propose a method capable of enhancing smartphone generated images by replacing the base ISP by a U-Net [85] resembled CNN (Convolutional Neural Network), called AWNet.

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

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

The structure of this network, following the 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 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 the test phase, a self-ensemble mechanism was applied, made up of 8 ensemble variants. Those variants where, 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 model. By applying this tunned model to the two tracks of the Zurich dataset from AIM 2020 Learned Smartphone ISP Challenge [99], the study reached the 5th and 2nd positions, respectivelly.

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 [67] and PyNet [31], with the use of the ZRR 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.

### 3.15 PyNET-CA

PyNET-CA is an end-to-end mobile ISP deep learning algorithm for RAW to RGB reconstruction [32]. This network improves PyNET [31] 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 [67], 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 Zurich Dataset [31] where it has shown better PSNR and SSIM values when compared to PyNET [31]. This paper did not present the number of network parameters and although the authors cite the decrease in training time, a table comparing these results to PyNET [31] was not presented.

### 3.16 Del-Net

Del-Net [36] is a single-stage end-to-end deep learning model that learns the entire ISP pipeline to convert RAW Bayer data to high-quality sRGB-image. This network uses a combination of Spatial and Channel Attention blocks (modified UNet) [100] and Enhancement Attention Modules blocks [57]. 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 [31], AWNet [35], and MW-ISPNet [99]) when considering the color enhancement, denoising, and detail retention capabilities while presenting a reduction 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 [31] where it has shown better detail retention compared to PyNET [31], better denoising compared to MW-ISPNet ignatov2020aim, and better colour enhancement compared to AWNet [35]. Although, the detail recovery capability of Del-Net is inferior to that of MW-ISPNet [99].

### 3.17 InvISP

InvISP [37] 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 1x1 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 [69]. 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 [58] and CycleISP [30], which implies a more accurate retrieval of RAW images. The method was compared to Invertible Grayscale [101] and U-net [71] baselines and the results showed better PSNR and SSIM values, indicating a more robust model for RAW image retrieval and RGB image generation.

### 3.18 ICDC-Net

In this paper, the authors have proposed an ISP-Net that addresses JPEG image compression in network training [38], 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 [102] and the architecture of CAS-Net was U-Net [85] without channel attention module.

Results are present on the Nikon D700 subset from the MIT-Adobe FiveK dataset [69]. 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.

### 3.19 CSANet

As the second place in the Mobile AI 2021 Learned Smartphone ISP Challenge[39] and first place in PSNR score, the Channel Spatial Attention Network (CSANet)[103] is a network that aims at computational performance and the quality of final image results, inferred at most 90.8 ms per image. The network has three main parts: downscale, processing blocks cascaded, and upscale. In the first part, to reduce the computational time and number of parameters, the authors did a downscale with a strided convolution block following a conventional convolution to feature extraction. Sequentially, the network has a Double Attention Module (DAM) inspired by the Convolutional Block Attention Module (CBAM)[104]. The DAM comprises a spatial attention module to learn spatial dependencies in the feature maps and a channel attention module to learn the inter-channel relationship of features maps. And in the last part has the convolution transpose and depth to space to upscale to a final RGB image. Another essential part of this work is the loss function composed by the Charbonnier loss[67], the SSIM loss, and the Perceptual loss to decrease the perceptual difference between the generated image and the ground truth.

In quantitative metrics on the validation dataset of Mobile AI 2021 Learned Smartphone ISP Challenge, the CSANet outperforms the PUNet, the baseline model for this Challenge, and has a shorter runtime. Also, it was comparable results and better inference time than the AWNet [35], a network with a high score in the AIM 2020. Compared with other candidates in the Challenge, this work was ranked second with satisfactory runtime e highest quality score. The CSANet can be used in embedded systems as the tests in smartphones showed. Besides, as expected, the use of the modules of attention helped in better color mapping.

### 3.20 LiteISPNet

In some datasets, the RAW and RGB images are captured with different cameras. Consequently, the pairs of images have misalignment and color inconsistency, difficulting the training process and producing blurry results. Thinking about this problem, Zhang et al. [40] 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 [105], 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, the LiteISPNet is responsible for mapping the RAW-to-RGB. It simplifies MW-ISPNet [99], 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 ZRR dataset[31], and the SR-RAW[106], with two variants of ground truth: the original GT and align GT. In the ZRR dataset, the LiteISPNet was compared with three states of the art (PyNet[31], AWNet[35], 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[107]. 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[92], ESRGAN [108], SPSR [109], and RealSR[110]. It generates images with less noise, less blurry, 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 problems, in addition to provide a new method to train DNN models with misalignment datasets. Besides, this new method enabled a lightweight network, such as used at work, and generated results near or higher than more robust models. However, the authors did not test the model in embedded systems.

### 3.21 TENet

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. [41], in extensive experiments, shown 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 [111], then 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) [108] 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 citeqian2021rethinking 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 [94], mosaic kernel [58], and addition of the Gaussian-Poisson noise model [111].

The model was compared with the ADMM [59], Condat [112], Flex-ISP [97], and DemosaicNet[4] [60] on the commonly used benchmark datasets to denoise and demosaicing tasks: Urban 100 [84], Kodak, McMaster [83] and BSD100 [113]. 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.

### 3.22 ReconfigISP

ReconfigISP [42] 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. This also reduces computational complexity and speeds up the training process. The loss function in this network was chosen according to the specific task desired.

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 [71] and S7 ISP Dataset [70] and showed that this network outperforms the traditional ISP pipelines achieving a higher PSNR value than Camera ISP. When compared to U-Net [71], the method obtained a better PSNR value for data with smaller numbers of patches in training but a lower value for larger-scale data.

### 3.23 ISP Distillation

In ISP Distillation [43], the authors proposed a model for image classification with RAW images using an sRGB image classification model and Knowledge Distillation [114] 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. Because of this, the authors applied the vision models directly to the RAW data.

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 [115]. The second was to discard the entire ISP pipeline in a model pre-trained on the HDR+ dataset [68].

ResNet18 [73] and MobileNetV2 [116] 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. Although the paper cited that ISP Distillation saves the computation cost of the ISP, the computation cost was not presented in the article.

### 3.24 Merging-ISP

Merging-ISP [44] consists of a deep neural network responsible for reconstructing multiple image layers LDR (low dynamic range) in just one image HDR (high dynamic range). Thus, the input data contains RAW images in dynamic or static scenes, where the network maps them and convolute all the layers in one exit HDR. Before the convolution, a DnCNN [117] conception applies a filter with 5x5 size and 64 layers, then applies two filters, both with 5x5 size and 64 layers, and, finally, applies three filters with 1x1 size and activation sigmoid. The output obtained, the data volume is reduced without applying another trainee and the LDR merge in just one HDR, comprising four convolutional layers Merging-ISP: Multi-Exposure High Dynamic Range Image Signal Processing 7 with decreasing receptive fields of 7 × 7 for 100 filters in the first layer to 1 × 1 for three filters in the last layer. Note that it was not necessary to apply an optical flow on input data.

To train the network, synthetic and real datasets based on Kalantari [118] datasets were used. The data contained dynamic and statics scenes, how was stated before. Secondly, rotation techniques were used to increase the dataset, contributing to extracting 210000 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 $10\text{e}^{-4}$ 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.

### 3.25 GCP-Net

Guo Et al. [45] 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 evaluate the frames offset, relieving the noise impact. The given work also sought out realistic noise models [119], [58], 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 LReLu blocks [120], 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 [121] 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 [122] and RViDeNet [123]. 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.

The study synthesized trained data using the Vimeo-90k open high-quality video dataset [98]. The training lasted two days, using the PyTorch framework and two Nvidia GeForce RTX 2080 Ti GPU. The authors used the Vid64 [124] and the REDS4 [122] datasets for the ablation experiments.

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 performance to other state-of-the-art Join Denoising-Demosaicing algorithms, such as FlexISP [97] and ADMM [59]. The Paper provided a complete introduction and related work explanation. There was also a proper and detailed experiment section, including fine points about training parameters. The ablation study validated the effectiveness of major GCP-Net components. The chosen comparison datasets enhanced the proposed model value, especially with real-world data verification.

### 3.26 PIPNet

In this paper [46], 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, including news perceptual losses to produce pleasure images 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 to leverage the architecture used. 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 [125], BSD100 [113], McMaster [83], Urban 100 [84], Kodak [82], and WED [126] datasets and compared with Deepjoint[4] [60], Kokkinos [96], Dong [127], DeepISP [70], and DPN [29] methods at three different noise levels (5, 15, 25). In all comparisons, PIPNet performed better over the PSNR, SSIM, and DeltaE2000 metrics. Qualitatively, the method also outperformed the other approaches. However, the network was tested only with data collected by traditional Bayer sensors, which may hinder network performance in other scenarios.

### 3.27 CURL

In this paper [47], 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

[4] Method from paper Deep Joint Demosaicking and Denoising [60], called by [33, 46] as Deepjoint, by [41] as DemosaicNet, and by [34] as DeepJDD.

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 [85] 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 [70], and CURL scored the best PSNR and LPIPS metrics when compared to the U-Net [85] and DeepISP [70] methods, but tied with the DeepISP method on the SSIM metric. In the second, results were validated over the MIT-Adobe FiveK dataset [69] and compared with HDRNet [128], DPE [129], White-Box [130], Distort-and-Recover [131], and DeepUPE [132] methods where CURL scored better on PSNR and LPIPS metrics, but DeepUPE scored the best on SSIM. Qualitatively CURL generates images very pleasing to the human eye.

# 4 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.

## 4.1 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 2 gives a list of all datasets discussed in this section with details about the number of images, size, and link to download.

Table 2: Summarization of datasets considered in the survey and their respective amount of images, size, and link to download.

| Dataset name | N° images | Size | Link to download |
|---|---|---|---|
| Zurich RAW to RGB [133] | 20.000 | ≈ 22 GB | `http://people.ee.ethz.ch/~ihnatova/pynet.html` |
| Urban 100 [84] | 100 | 1.14 GB | `https://github.com/jbhuang0604/SelfExSR` |
| McMaster dataset [83] | 18 | 13.6 MB | `https://web.comp.polyu.edu.hk/cslzhang/CDM_Dataset.htm` |
| Kodak | 25 | 119.7 MB | `http://r0k.us/graphics/kodak/` |

### 4.1.1 Zurich RAW to RGB (ZRR)

ZRR dataset was proposed by Ignatov Et. al. [133] 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. The photos were captured in automatic mode, however, some RAW–RGB image pairs are not perfectly aligned, which requires a preprocessing and matching performance. ZRR was a recurrent dataset choice for some RAW to RGB mapping problem works considered in this survey [22, 36, 31, 40, 35, 32].

#### 4.1.2 Urban 100

The Urban 100 dataset consists of 100 High-Resolution images with an assortment of real-world urban scenarios and structures. It was proposed by Huang Et. al. [84], in order to address the lack of High-Resolution Datasets with indoor, urban, and architectural scenes. Urban 100 was constructed with synthetic images from Flickr[5], under Creative Commons license, resulting in a 1.14 GB dataset. It is a well-known public database for super-resolution tasks [29, 28, 34, 46, 33, 29].

#### 4.1.3 McMaster dataset

The McMaster dataset was proposed by Zhang et al. [83] and consists of eighteen 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. When compared with the Kodak color image dataset, McMaster shows images with more saturated colors and more color transitions between the image's objects. However, this dataset is still limited in scene variation and colors gradations. The McMaster dataset is used for color demosaicing in some of the articles in this survey [46, 28, 29, 24, 41].

#### 4.1.4 Kodak

Kodak [6] is a little dataset composed of 24 photographic quality images of size 768x512 or 512x768 with a large variety of locations and lighting conditions. Initially, it was created to be a sample Kodak Photo CD, with some images captured by Kodak's professional photographers and others selected from the winners of the Kodak International Newspaper Snapshot Awards(KINSA). 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 [46, 28, 24, 29, 41].

### 4.2 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.
- **Preocupation 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.

## 5 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 differents câmeras 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

[5] https://www.flickr.com/

[6] http://r0k.us/graphics/kodak/

by Kodak film. Finally, in the fourth section, we discussed the Kodak dataset with 24 photographic quality images of size 768x512 or 512x768, generally used in compression tests and to validate methods for tasks like demosaicking and denoising, etc.

Table 3: ZURICH RAW2RGB dataset [31]

| Networks | PSNR | SSIM |
|---|---|---|
| Del-Net[36] | 21.46 | 0.745 |
| PyNet[31] | 21.19 | 0.746 |
| AWNet (Ensemble)[35] | 21.86 | 0.781 |
| AWNet (Demosaiced)[35] | 21.38 | 0.745 |
| AWNet (RAW)[35] | 21.58 | 0.749 |
| PyNet-CA[32] | 21.50 | 0.743 |
| LiteISPNet[40] | 21.55 | 0.748 |
| LiteISPNet[40] | **23.76**[a] | **0.873**[a] |
| HERNet[22] | 22.59[b] | 0.81[b] |

[a] Align ground truth with RAW image.

[b] Data with diferent distribution.

Table 4: Urban 100 dataset [84]

| Networks | PSNR | SSIM |
|---|---|---|
| DPN [29] | 37.70 | 0.9799 |
| PIPNet [46] | 37.51[a] | 0.9731[a] |
| TENet [41] | 29.37[c] | 0.9061[c] |
| SGNet[33] | 34.54[a] | 0.9533[a] |
| RLDD [28] | **39.52**[b] | **0.9864**[b] |
| RestoreNet w/ PatchNet [34] | 34.66[a] | - |
| DPN [29] | 37.70 | 0.9799 |

[a] Data with noise.

[b] 10 pixels were removed from image's border to calculate the PSNR value.

[c] Downsampled data to do demosaicing + SR task.

### 5.1 Zurich RAW to RGB

The HERNet and LiteISPNet present versions with better results in the Table 3, but it uses, respectively, a different distribution of data and the RAW image aligned with the ground-truth. With the remain of results, the AWNet outperformed other methods in PSNR and SSIM scores, and the use of a self-ensemble strategy can explain this, as RAW and Demosaiced versions have scores relatively much lower than the ensembled version. Besides, the AWNet RAW version has slightly higher results than LiteISPNet and PyNet-CA, which can result from the wavelet transform and context global blocks. Furthermore, the LiteISPNet has impressive results with introducing an additional method to align the images, improving network training.

### 5.2 Urban 100

Urban 100 dataset is used for many tasks of image restoration. Then the works do modifications to adequate this dataset to a specific task. As shown in Table 4, the RLDD had the highest scores compared to other methods but was validated with the image's border removed, and this factor can help increase scores. Furthermore, the PIPNet, even being validated with noisy data, has comparable results with DPN. The introduction of attention mechanisms that bring good correlations in relation to depth and spatial dimensions may be the cause of these promising results in this dataset.

### 5.3 McMaster

The DRDN+ and DRDN methods showed better results in the PSNR metric in Table 5. 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.

Table 5: McMaster dataset [83]

| Networks | PSNR | SSIM |
|---|---|---|
| DRDN [24] | 38.88 | 0.9689 |
| DRDN+ [24] | **39.02** | 0.9697 |
| DPN [29] | 37.6 | **0.9842** |
| TENet[41] | 32.40[c] | 0.9163[c] |
| PIPNet [46] | 38.13[a] | 0.9612[a] |
| RLDD [28] | 36.61[b] | 0.9725[b] |

[a] Data with noise.
[b] 10 pixels were removed from image's border to calculate the PSNR value.
[c] Downsampled data to do demosaicing + SR task.

Table 6: Kodak Dataset [82]

| Networks | PSNR | SSIM |
|---|---|---|
| DRDN [24] | 42.43 | 0.9889 |
| DRDN+ [24] | 42.66 | **0.9893** |
| DPN [29] | 40.1 | 0.9846 |
| TENet[41] | 31.39 [a] | 0.8965[a] |
| PIPNet [46] | 39.37[a] | 0.9768[a] |
| RLDD [28] | **42.76**[b] | 0.9893 |

[a] Data with noise.
[b] 10 pixels were removed from image's border to calculate the PSNR value.

### 5.4 Kodak

Table 6 shows the reviewed paper results in the Kodak dataset [7]. RLDD [28] achieved the best PSNR metric performance, whereas DRDN+ [24] 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.

### 5.5 Source Code Links

The Table 7 presents the links to the source codes of some works mentioned in this survey.

Table 7: Summarization of some approaches considered in the survey and their respective source codes.

| Short Name | Source Code |
|---|---|
| CycleISP [30] | PyTorch: `https://github.com/swz30/CycleISP` |
| CURL [47] | PyTorch: `https://github.com/sjmoran/CURL` |
| LiteISPNet [40] | PyTorch: `https://github.com/cszhilu1998/RAW-to-sRGB` |
| PyNet [31] | PyTorch and Tensorflow: http://people.ee.ethz.ch/ ihnatova/pynet.html |
| PyNet-CA [32] | PyTorch: `https://github.com/egyptdj/skyb-aim2020-public` |
| BJDD [46] | PyTorch: `https://github.com/sharif-apu/BJDD_CVPR21` |
| TENet [41] | PyTorch: `https://github.com/guochengqian/TENet` |
| AWNet [35] | PyTorch: https://github.com/Charlie0215/AWNet-Attentive<br>-Wavelet-Network-for-Image-ISP |
| Bayer Methods [26] | PyTorch: `https://github.com/Jiaming-Liu/BayerUnifyAug` |
| HERNet [22] | PyTorch: `https://github.com/MKFMIKU/RAW2RGBNet` |

## 6 Discussion

We provide, in this section, points of improvement in these works about its methodology and evaluation. However, much of it has highlights which can contribute a lot about the learned cameras ISPs. This section will address the negative and positive points of the works described above and ask questions that may evolve into new paths to rods in this research field.

### 6.1 Dataset Created with Different Cameras

Some datasets [31, 39] have the pair images captured by different cameras, where one of the main objectives of the works that use these datasets is to learn the features of high-quality cameras and apply them in cameras with more limitations. Because of the differences between the devices, these datasets have alignment

[7] http://r0k.us/graphics/kodak/

problems about the image patch pairs. Doing the RAW to RGB mapping can be challenging because the patch pixels do not correspond to theirs pixels. The trivial solution is to align the pairs, but how to do this is the main question. Zhang et al.[40] propose the use of Deep Learning where outperforms the models trained with the misalignment dataset. Another way that many works [39, 35, 31, 36] use to mitigate this problem is to adapt the loss function to not punish the models where the generated image is misaligned with the ground truth. Models trained with losses that try to get closer to human perception are more invariant to misalignment and generate images with more details. Currently, the metrics that are closest to this make use of Deep Learning techniques, using extracted features of images with the use of pre-trained Deep Learning models [107, 134]. Thus comes the doubt of how the best misaligment solution should be used to mitigate the problem adding no overhead on the computational cost of the training process. This problem can become a search point by itself, such as alignment methods to RAW and RGB images of differents cameras, facilitating the creation of new datasets and improving the performance of the new works.

### 6.2 Methodologies

Seen as a point that helps share knowledge and the research field evolution is the justification of each stage, from architecture construction to validation of the work or the choice of datasets. Nevertheless, some works do not do it. For example, Liu et al. [33] do not explain well why the use of some datasets and Liu et al. [26] do not indicate the motivation to use the U-Net model. Besides, many methods do not make the source code available in its paper [22, 24, 23, 3, 25, 28], making it difficult for future validations and comparisons with other methods and making the validation less reliable.

### 6.3 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 can not 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."[39] 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.

## 7 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, aiming to apply all of them in a single forward operation.

Furthermore, this work also summarized five datasets often used for ISP tasks: Urban 100 [84], ZRR dataset [133], Kodak[8], and McMaster dataset [83]. 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:

[8] http://r0k.us/graphics/kodak/

- 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.
- Better ablation studies, regarding the effect of each architectural component present in proposed solutions, alongside with the distribution of the source code, can facilitate the future development of new techniques and the overall improvement of the research area.
- As more approaches are proposed to the ISP pipeline replacement task, a more consistent evaluation procedure, alongside with the definition of common target datasets, will facilitate the comparison between methods, helping to define state-of-the-art performance.
- 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 or new techniques focused on these target environments.

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

[2] Yu Zhu, Zhenyu Guo, Tian Liang, Xiangyu He, Chenghua Li, Cong Leng, Bo Jiang, Yifan Zhang, and Jian Cheng. Eednet: Enhanced encoder-decoder network for autoisp. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision – ECCV 2020 Workshops*, pages 171–184, Cham, 2020. Springer International Publishing.

[3] Sivalogeswaran Ratnasingam. Deep camera: A fully convolutional neural network for image signal processing, 2019.

[4] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

[5] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W R Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, January 2020.

[6] Padideh Danaee, Reza Ghaeini, and David A Hendrix. A deep learning approach for cancer detection and relevant gene identification. *Pac. Symp. Biocomput.*, 22:219–229, 2017.

[7] Hugo Storm, Kathy Baylis, and Thomas Heckelei. Machine learning in agricultural and applied economics. *Eur. Rev. Agric. Econ.*, 47(3):849–892, June 2020.

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

[9] Faustino J. Gomez. Co-evolving recurrent neurons learn deep memory pomdps. In *InGECCO-05: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 491–498.

[10] Juergen Schmidhuber. Deep learning. *Scholarpedia J.*, 10(11):32832, 2015.

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

[12] Y. Bengio and Yann Lecun. Convolutional networks for images, speech, and time-series. 11 1997.

[13] K Fukushima. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.*, 36(4):193–202, 1980.

[14] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017.

[15] Andreas Kamilaris and Francesc X. Prenafeta-Boldú. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147:70–90, April 2018.

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

[17] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, April 2020.

[18] 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. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, December 2017.

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

[20] Xin Li, Bahadir Gunturk, and Lei Zhang. Image demosaicing: a systematic survey. In William A. Pearlman, John W. Woods, and Ligang Lu, editors, *Visual Communications and Image Processing 2008*. SPIE, January 2008.

[21] Linwei Fan, Fan Zhang, Hui Fan, and Caiming Zhang. Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, 2(1), July 2019.

[22] Kangfu Mei, Juncheng Li, Jiajie Zhang, Haoyu Wu, Jie Li, and Rui Huang. Higher-resolution network for image demosaicing and enhancing, 2019.

[23] Zhetong Liang, Jianrui Cai, Zisheng Cao, and Lei Zhang. Cameranet: A two-stage framework for effective camera isp learning, 2019.

[24] Bumjun Park and Jechang Jeong. Color filter array demosaicking using densely connected residual network. In *Color Filter Array Demosaicking Using Densely Connected Residual Network*, 2019.

[25] Vahid Patrovi Nia Ramchalam Kinattinkara Ramakrishnan, Shangling Jui. Deep demosaicing for edge implementation, 2019.

[26] Yuzhi Wang Qin Xu Yuqian Zhou Haibin Huang Chuan Wang Shaofan Cai Yifan Ding Haoqiang Fan Jue Wang Jiaming Liu, Chi-Hao Wu. Learning raw image denoising with bayer pattern unification and bayer preserving augmentation, 2019.

[27] Sushma Rao Bhavin Nayak Timo Gerasimow Aleksandar Sutic Liron Ain-kedem Gilad Michael O Chyuan-Tyng Wu, Leo F. Isikdogan. Visionisp: Repurposing the image signal processor for computer vision applications, 2019.

[28] Yu Guo, Qiyu Jin, Gabriele Facciolo, Tieyong Zeng, and Jean-Michel Morel. Residual learning for effective joint demosaicing-denoising, 2020.

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

[30] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis, 2020.

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

[32] Byung-Hoon Kim, Joonyoung Song, Jong Chul Ye, and JaeHyun Baek. Pynet-ca: Enhanced pynet with channel attention for end-to-end mobile image signal processing. *Lecture Notes in Computer Science*, page 202–212, 2020.

[33] Lin Liu, Xu Jia, Jianzhuang Liu, and Qi Tian. Joint demosaicing and denoising with self guidance. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2237–2246, 2020.

[34] Shuyang Sun, Liang Chen, Gregory Slabaugh, and Philip Torr. Learning to sample the most useful training patches from images. *arXiv preprint arXiv:2011.12097*, 2020.

[35] Linhui Dai, Xiaohong Liu, Chengqi Li, and Jun Chen. Awnet: Attentive wavelet network for image isp, 2020.

[36] Saumya Gupta, Diplav Srivastava, Umang Chaturvedi, Anurag Jain, and Gaurav Khandelwal. Del-net: A single-stage network for mobile camera isp, 2021.

[37] Yazhou Xing, Zian Qian, and Qifeng Chen. Invertible image signal processing, 2021.

[38] Kwang-Hyun Uhm, Kyuyeon Choi, Seung-Won Jung, and Sung-Jea Ko. Image compression-aware deep camera isp network. *IEEE Access*, 9:137824–137832, 2021.

[39] 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, Jin Guo, Chao-Hung Chen, Ming-Chun Hsyu, Wen-Chia Tsai, Chao-Wei Chen, Grigory Malivenko, Minsu Kwon, Myungje Lee, Jaeyoon Yoo, Changbeom Kang, Shinjo Wang, Zheng Shaolong, Hao Dejun, Xie Fen, Feng Zhuang, Yipeng Ma, Jingyang Peng, Tao Wang, Fenglong Song, Chih-Chung Hsu, Kwan-Lin Chen, Mei-Hsuang Wu, Vishal Chudasama, Kalpesh Prajapati, Heena Patel, Anjali Sarvaiya, Kishor Upla, Kiran Raja, Raghavendra Ramachandra, Christoph Busch, and Etienne de Stoutz. Learned smartphone isp on mobile npus with deep learning, mobile ai 2021 challenge: Report, 2021.

[40] Zhilu Zhang, Haolin Wang, Ming Liu, Ruohao Wang, Jiawei Zhang, and Wangmeng Zuo. Learning raw-to-srgb mappings with inaccurately aligned supervision, 2021.

[41] Guocheng Qian, Yuanhao Wang, Chao Dong, Jimmy S. Ren, Wolfgang Heidrich, Bernard Ghanem, and Jinjin Gu. Rethinking the pipeline of demosaicing, denoising and super-resolution, 2021.

[42] Ke Yu, Zexian Li, Yue Peng, Chen Change Loy, and Jinwei Gu. Reconfigisp: Reconfigurable camera image processing pipeline, 2021.

[43] Eli Schwartz, Alex Bronstein, and Raja Giryes. Isp distillation, 2021.

[44] Prashant Chaudhari, Franziska Schirrmacher, Andreas Maier, Christian Riess, and Thomas Köhler. Merging-isp: Multi-exposure high dynamic range image signal processing, 2021.

[45] Zhetong Liang Shi Guo and Lei Zhang. Joint denoising and demosaicking with green channel prior for real-world burst images, 2021.

[46] SM A Sharif, Rizwan Ali Naqvi, and Mithun Biswas. 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*, pages 233–242, 2021.

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

[48] Peter Wilson. Chapter 7 - high speed video application. In Peter Wilson, editor, *Design Recipes for FPGAs (Second Edition)*, pages 67–77. Newnes, Oxford, second edition edition, 2016.

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

[50] Liu Yongji and Yuan Xiaojun. A design of dynamic defective pixel correction for image sensor. In *2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS)*, pages 713–716, 2020.

[51] Georgi Zapryanov, Ivanova, and Iva Nikolova. Automatic white balance algorithms for digital still cameras - a comparative study. *Information Technologies and Control*, 1:16–22, 01 2012.

[52] K. Hirakawa and T.W. Parks. Adaptive homogeneity-directed demosaicing algorithm. *IEEE Transactions on Image Processing*, 14(3):360–369, 2005.

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

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

[55] Dr.Sabeenian R.S. A survey on image denoising algorithms' (ida),. *Science, Measurement and Technology, IEE Proceedings A*, 1:456–462, 11 2012.

[56] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview, 2020.

[57] Saeed Anwar and Nick Barnes. Real image denoising with feature attention, 2020.

[58] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T. Barron. Unprocessing images for learned raw denoising, 2018.

[59] Hanlin Tan, Xiangrong Zeng, Shiming Lai, Yu Liu, and Maojun Zhang. Joint demosaicing and denoising of noisy bayer images with admm. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2951–2955, 2017.

[60] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Trans. Graph.*, 35(6), November 2016.

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

[62] Fagun Vankawala, Amit Ganatra, and Amit Patel. A survey on different image deblurring techniques. *International Journal of Computer Applications*, 116:15–18, 2015.

[63] Qolamreza Razlighi and Nasser Kehtarnavaz. Imaeg blur reduction for cell-phone cameras via adaptive tonal correction. 1:I – 113, 09 2007.

[64] Zhe Hu, Lu Yuan, Stephen Ching-Feng Lin, and Ming-Hsuan Yang. Image deblurring using smartphone inertial sensors. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1855–1864, 2016.

[65] Siddhant Sahu, Manoj Kumar Lenka, and Pankaj Kumar Sa. Blind deblurring using deep learning: A survey. *ArXiv*, abs/1907.10128, 2019.

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

[67] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks, 2018.

[68] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (TOG)*, 35:1–12, 2016.

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

[70] Eli a Schwartz. Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing*, 28(2).

[71] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark, 2018.

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

[73] S. Ren K. He, X. Zhang and J. Sun. Deep residual learning for image recognition, 2016.

[74] L. van der Maaten G. Huang, Z. Liu and K. Q. Weinberger. Densely connected convolutional networks. *EEE Conf. Comput. Vis. Pattern Recognit.*, pages 2261–2269, 2017.

[75] Daniele Menon and Giancarlo Calvagno. Color image demosaicking: An overview. *Signal Processing: Image Communication*, 26, oct 2011.

[76] Michael S. Brown Abdelrahman Abdelhamed, Stephen Lin. A high-quality denoising dataset for smartphone cameras. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, jun 2018.

[77] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[78] Sushma Rao Aleksandar Sutic Chyuan-Tyng Wu Gilad Michael Jun Nishimura, Timo Gerasimow. Automatic isp image quality tuning using non-linear optimization, 2019.

[79] Raquel Urtasun Andreas Geiger, Philip Lenz. Are we ready for autonomous driving? the kitti vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, jun 2012.

[80] Forrest Iandola Peter H. Jin-Kurt Keutzer Bichen Wu, Alvin Wan. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving, 2016.

[81] Ibrahim Pekkucuksen and Yucel Altunbasak. Gradient based threshold free color filter array interpolation. In *2010 IEEE International Conference on Image Processing*, pages 137–140, 2010.

[82] R. Franzen. Kodak lossless true color image suite. `http://r0k.us/graphics/kodak/`.

[83] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic Imaging*, 20(2):1 – 17, 2011.

[84] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, 2015.

[85] P. Fischer O. Ronneberger and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[86] Stefan Roth Tobias Plötz. Benchmarking denoising algorithms with real photographs, 2017.

[87] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[88] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

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

[90] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3277–3285, 2017.

[91] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[92] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2017.

[93] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[94] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks, 2015.

[95] Run xu Tan, K. Zhang, Wangmeng Zuo, and Lei Zhang. Color image demosaicking via deep residual learning. 2017.

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

[97] 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. Flexisp: A flexible camera image processing framework. *ACM Trans. Graph.*, 33(6), November 2014.

[98] Jiajun Wu Donglai Wei William T. Freeman Tianfan Xue, Baian Chen. Video enhancement with task-oriented flow, 2019.

[99] Andrey Ignatov, Radu Timofte, Zhilu Zhang, Ming Liu, Haolin Wang, Wangmeng Zuo, Jiawei Zhang, Ruimao Zhang, Zhanglin Peng, Sijie Ren, Linhui Dai, Xiaohong Liu, Chengqi Li, Jun Chen, Yuichi Ito, Bhavya Vasudeva, Puneesh Deora, Umapada Pal, Zhenyu Guo, Yu Zhu, Tian Liang, Chenghua Li, Cong Leng, Zhihong Pan, Baopu Li, Byung-Hoon Kim, Joonyoung Song, Jong Chul Ye, JaeHyun Baek, Magauiya Zhussip, Yeskendir Koishekenov, Hwechul Cho Ye, Xin Liu, Xueying Hu, Jun Jiang, Jinwei Gu, Kai Li, Pengliang Tan, and Bingxin Hou. Aim 2020 challenge on learned image signal processing pipeline. 2020.

[100] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement, 2020.

[101] Menghan Xia, Xueting Liu, and Tien-Tsin Wong. Invertible grayscale. *ACM Transactions on Graphics (TOG)*, 37(6):1–10, 2018.

[102] Kwang-Hyun Uhm, Seung-Wook Kim, Seo-Won Ji, Sung-Jin Cho, Jun-Pyo Hong, and Sung-Jea Ko. 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.

[103] Ming-Chun Hsyu, Chih-Wei Liu, Chao-Hung Chen, Chao-Wei Chen, and Wen-Chia Tsai. Csanet: High speed channel spatial attention network for mobile isp. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2486–2493, 2021.

[104] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module, 2018.

[105] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume, 2018.

[106] Xuaner Cecilia Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun. Zoom to learn, learn to zoom, 2019.

[107] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018.

[108] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. Esrgan: Enhanced super-resolution generative adversarial networks, 2018.

[109] Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance, 2020.

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

[111] Samuel W. Hasinoff. *Photon, Poisson Noise*, pages 608–610. Springer US, Boston, MA, 2014.

[112] Laurent Condat and Saleh Mosaddegh. Joint demosaicking and denoising by total variation minimization. In *2012 19th IEEE International Conference on Image Processing*, pages 2781–2784, 2012.

[113] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. volume 2, pages 416–423 vol.2, 02 2001.

[114] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[115] 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. Imagenet large scale visual recognition challenge, 2015.

[116] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019.

[117] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

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

[119] Jiawen Chen Dillon Sharlet Ren Ng Robert Carroll Ben Mildenhall, Jonathan T. Barron. Burst denoising with kernel prediction networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, jun 2018.

[120] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network, 2015.

[121] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[122] Ke Yu Chao Dong Chen Change Loy Xintao Wang, Kelvin C.K. Chan. Edvr: Video restoration with enhanced deformable convolutional networks, 2019.

[123] Lei Liao Ronghe Chu Jingyu Yang Huanjing Yue, Cong Cao. Supervised raw video denoising with a benchmark dataset on dynamic scenes, 2020.

[124] Deqing Sun Ce Liu. On bayesian adaptive video super resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2), feb 2014.

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

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

[127] Weishong Dong, Ming Yuan, Xin Li, and Guangming Shi. Joint demosaicing and denoising with perceptual optimization on a generative adversarial network. *arXiv preprint arXiv:1802.04723*, 2018.

[128] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017.

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

[130] Yuanming Hu, Hao He, Chenxi Xu, Baoyuan Wang, and Stephen Lin. Exposure: A white-box photo post-processing framework. *ACM Transactions on Graphics (TOG)*, 37(2):1–17, 2018.

[131] Jongchan Park, Joon-Young Lee, Donggeun Yoo, and In So Kweon. Distort-and-recover: Color enhancement using deep reinforcement learning. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 5928–5936, 2018.

[132] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6849–6857, 2019.

[133] Andrey D. Ignatov, Juncheng Li, Jiajie Zhang, Haoyu Wu, Jie Li, Rui Huang, Muhammad Haris, Greg Shakhnarovich, Norimichi Ukita, Yuzhi Zhao, Lai-Man Po, Radu Timofte, Tiantian Zhang, Zongbang Liao, Xiang Shi, Yujia Zhang, Weifeng Ou, Pengfei Xian, Jingjing Xiong, Changxian Zhou, Wing Yin Yu, Yubin Yubin, Sung-Jea Ko, Bingxin Hou, Bumjun Park, Songhyun Yu, Sangmin Kim, Jechang Jeong, Seung wook Kim, Kwang-Hyun Uhm, Seo-Won Ji, Sung-Jin Cho, Jun-Pyo Hong, and Kangfu Mei. Aim 2019 challenge on raw to rgb mapping: Methods and results. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3584–3590, 2019.

[134] Aamir Mustafa, Aliaksei Mikhailiuk, Dan Andrei Iliescu, Varun Babbar, and Rafal K. Mantiuk. Training a task-specific image reconstruction loss, 2021.