Erasmus+

COSI Computational Colour and Spectral Imaging

**Master in Computational Colour and Spectral Imaging (COSI)**

UNIVERSITY OF EASTERN FINLAND · UNIVERSIDAD DE GRANADA · UNIVERSITÉ DE LYON · UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE · NTNU Norwegian University of Science and Technology

# Learned Image Signal Processing Pipeline for Mobile Cameras

Master Thesis Report

Presented by

OMAR ELEZABI

and defended at the

Norwegian University of Science and Technology

September 2023

Academic Supervisor: Dr. Damien Muselet, Jean Monnet University
Host Supervisor: Prof. Radu Timofte, Julius-Maximilians-Universität Würzburg
Jury Committee:

1. Prof. Marius Pedersen, Norwegian University of Science and Technology, Gjøvik
2. Dr. Mozhdeh Seifi, European Patent Office

Submission of the thesis: 10$^{\text{th}}$ August 2023
Day of the oral defense: 4$^{\text{th}}$ September 2023

# Abstract

The image signal processing (ISP) pipeline is a crucial part of the image creation process. This pipeline consists of a handcrafted and complex sequence of image-processing tasks that are used to process the raw image from the camera sensor and produce the final RGB image. Because of the hardware limitation in mobile cameras from their compact size, the ISP of mobile phones became more advanced and complex to overcome these limitations. In previous years a new research direction proposed to replace this complex hand-crafted pipeline with an end-to-end learned-based ISP using deep learning. They achieved that by training a deep learning network to process the raw image of a phone camera by imitating the output of a DSLR camera. This approach showed promising results without the need for the long and complex process of handcrafted conventional ISP. But this approach is still a research direction that has a lot of limitations and problems compare to the conventional ISP used in mobile cameras nowadays. In order to reach production-level accuracy and robustness with this approach a lot of work needs to be done to address its issues.

In this work, we tried to improve the current state of learned-based ISP by addressing some of its main problems. We worked on night image rendering by using a learned-based ISP Network. We proposed an efficient network that was trained without the need for annotated data. Our proposed approach was one of the top 10 solutions on the NTIRE 2023 Challenge on Night Photography Rendering.

We also worked on the problems of the ISP datasets like alignment and availability. We proposed a novel idea to create a fully aligned high-quality synthetic ISP dataset with a weakly aligned ISP dataset. Our experiments show that We get better performance by training on our synthetic dataset than directly training on the weakly aligned dataset which shows the effectiveness of our pipeline. We also showed the ability of our pipeline to generate a new synthetic dataset from just DSLR RGB images.

Lastly, we addressed the problem of missed global information in the learned ISP networks. We proposed a novel color module that utilizes the global information from the full raw image in addition to local information from the input raw patch. Our module is a general module that can be integrated with any ISP Network to improve its color reproduction accuracy. We achieved state-of-the-art performance by utilizing our simple and efficient color module with a simple ISP network. We showed that by just utilizing the global information from the full image we can immensely improve the performance of ISP Networks.

# Acknowledgment

Praise be to Allah, Lord of the Worlds. Prayer and peace be upon the Prophet Muhammad. I express my gratitude to Allah for all the blessings and the strength to continue this long journey.

I want to thank Dr. Damien Muselet for his continuous support throughout the project. I would like to express my gratitude to Prof. Radu Timofte for giving me the opportunity to work in his lab and for his continuous support throughout the whole process.

I want to thank all my COSI classmates. The journey was easier because of these nice people. We managed to get through the hard parts together and enjoy the fun parts together.

Lastly, I want to thank my family, my father, my mother, and my three sisters for their nonending love and help. Without them, I will not be here today.

To a lot of memories worth remembering and more adventures to come.

# Contents

CONTENTS

# 1 Introduction

Photos have been and will always be a big part of our lives from capturing moments that we don't want to forget to share information and moments with others or even for identifications in official documents they have become an attached part of our everyday. All of this is because of the advances and the accessibility of the camera especially after moving from a camera that was dependent on chemical processes to digital cameras that exist nowadays. Digital cameras are basically a sensor that consists of an array of light meters that measure the amount of light that is incident on the camera and gathered using the optical system of the camera. The result of this measurement is called the Raw Image. Before we obtain the images we usually see we need to process these measurements first which is the role of the image signal processing (ISP) pipeline. This can be seen as a simulation of the human visual system as the eyes convert the light into neural signals that then get transferred to the brain to get processed to what we finally perceive.

## 1.1 What is Image Signal Processing (ISP) Pipeline?

The ISP consists of a sequence of low-level and global image processing tasks, like demosaicing, white balancing, and more, that process the light measurements to obtain a final RGB image that is suitable for the displaying device and in a representation similar to what HVS would perceive it. The ISP is very different for different camera devices and also depends on the application the camera is used for, For example, the ISP used in the cameras used in the manufacturing process is very different than the ISP in photography cameras and also the camera task heavily influence the final output as some ISPs are optimized for accuracy of reproduction like medical applications and other are optimized for pleasness. Additionally, ISPs are very limited with the computational resources available and this differs based on the application as we can notice in Capsule endoscopy with the limited hardware and size the ISP is much simpler. For our work, we focus on

the ISPs developed for photography cameras, especially for mobile phone cameras which are mostly optimized for pleasness and accuracy. In the Next chapter 2, we will describe the base conventional ISP Pipeline, the different tasks, and the details about each task and what is solving.

Mobile cameras and photography is currently the most popular way of photography, especially with the current advances in mobile cameras. What makes it so popular is the convinces and small size of mobile devices which you can take it anywhere with you without hustle which increases the chances of taking photos. This advantage is also the biggest problem that faces the quality of mobile cameras as this small size forces a lot of hardware limitations like small sensors and compact optical systems compare to the professional less compact camera systems like DSLR Cameras. Even though there has been a big improvement in the past several years to develop hardware that reduces the gap between smaller size compact systems and bigger systems, still currently the most crustal part to reduce this difference is the ISP of the mobile phone. Because of these hardware limitations, the ISPs developed for mobile phones are much more complex and advanced than other cameras to eliminate the difference between the mobile cameras with their limited hardware and what can be captured with a professional camera. Additionally, ISP tries to retrieve some of the information that is lost because of the limited hardware like sharpness, and dynamic range by estimation using computational process or through additional hardware. ISP can also be used to simulate some of the features that are not possible with the available hardware like Zoom and Bokeh.

## 1.2 Main Types Of ISP, Pros, and Cons.

Conventional ISP used nowadays in all mobile cameras consist of a number of steps and tasks that are cultivated together to process the sensor data to produce images compatible with the display device. This pipeline and its tasks are created through a long process of research, calibration, tuning, and evaluation and was improved throughout the years with teams that consist of tens and hundreds of people. This was the case with other tasks that requires a lot of feature engineering like object detection or automation but with the huge emergence of learning-based methods, these tasks improved and became less complex and less dependent on high human tuning. This started to be the case in some of the modules and tasks in the ISP pipeline like using Neural Network models for image enhancement modules Qi et al. (2021). This addition added a lot of improvement and robustness to some of the ISP modules but the whole ISP is still dependent on the same concept of a sequence of tasks that still require a lot of tuning and calibration which is a long and complex costly process. But this type of ISP is still used to this day because it was created through years and years of work and cumulative experience.

It is also reliable, provide a lot of flexibility, more controllable, which provides the expected output with much fewer outliers and the ability to debug and tune each component in the desired way. This also has a hardware aspect as this separation between tasks and modules allowed the manufacturers to integrate the ISP tasks and modules directly into chips which makes the computation much more efficient. But this accumulation of tasks in addition to the complexity provides accumulative error that is passed though out the modules which limits the ISP performance.

Currently With the shown power of neural networks in different tasks in addition to some of the ISP tasks neural networks can be used to decrease the complexity of ISP and improve its performance. In addition, some of the recent research showed the power of neural networks in solving multiple tasks in the ISP using with end-to-end network and can even exceed traditional methods like the work on joint demosaicing and denoising Gharbi et al. (2016). This shows the possibility to reconfigure the conventional way of creating ISP by utilizing learning-based methods and neural networks to a more robust and less complex design.

To investigate the limit of learned-based ISP a new research direction started to utilize the fully learned-based pipeline by replacing the whole ISP process with neural networks. Some even investigated replacing the whole ISP process with an end-to-end neural network Ignatov et al. (2020b). They achieved this by creating a Ground Truth (GT) using more capable professional cameras like DSLR which has much better hardware and much less size limitation and using this GT to train a neural network to process the input raw images that are produced from a mobile phone to imitate the output of the DSLR camera to reduce the quality difference between the 2 camera systems because of the hardware limitations.

This method has the benefit of being much simpler and requires much less tuning and human expertise. Also because it's an end-to-end process it has less error accumulation which can result in better performance. But this direction is an early-stage research direction and not mature enough for production and has many problems compared to the current more mature Conventional ISP design. This direction is very dataset dependent and limited with the dataset used in training and the scenes used which require the need of using a lot of data to produce a robust ISP. This increases the risk of outliers, as we can't predict the model performance on the cases that were not included in the training data (we can't use the model trained on daylight images with night images). Additionally, the dataset is expensive to collect and requires tuning and alignment, and even after that process still suffers from a miss-alignment because of the use of 2 different camera systems to create the dataset. Also, the use of an end-to-end network limits the flexibility of the ISP to adjust the settings of the system (different white-balance settings or different noise processing levels) to produce different outputs.

The research studies showed this process can produce very good results and

can even exceed the performance of the phone camera's internal ISP but only for the cases aligned with the training datasets. This direction still needs more investigation and development. Neural Networks are still considered a black box and hard to interpret and hard to predict output which makes it hard to debug and more prone to outliers, especially with cases that were not included in its development process. This makes it hard to replace the whole ISP with an end-to-end network. But for our research, we will focus on this direction to process the raw images from the phone camera system which requires the development of a robust and efficient Network. We will address some of the problems and drawbacks of this approach and try to introduce some solutions for these problems. We will also investigate the current state and the future of this direction and how it can be beneficial for phone cameras.

## 1.3 Work Overview

We started by reviewing the current state of Learned-based ISP with the currently proposed solutions, datasets, and benchmarks and identified the problems and drawbacks of each approach and the current general problems with the Learned-based ISP models.

### 1.3.1 Learned Based ISP Problems

The biggest problems we found are mostly related to the datasets. most of the Raw to RGB datasets are created by the raw images from a camera's sensor and the output of the camera's internal ISP which is more concerned with simulating the internal ISP of the camera and not creating a better more elevated ISP. These datasets mostly consist of DSLR cameras as it is hard to obtain raw images from mobile cameras. This limits us with the dataset available that we can use to develop ISP for mobile cameras. Additionally, because we are dealing with sensor data the datasets are also sensor specific so in order to work with a specific sensor you need to create a new dataset for this sensor which makes the process more expensive and depends on the available dataset. For the ISP-creating process, we focused on the dataset created by mobile phones and DSLR cameras to process the phone raw by simulating the output of the DSLR. This kind of dataset is very limited and according to our research, only 3 datasets are available. These datasets also have their own problem with miss-alignments because they need two different camera systems to create. The dependence on different camera systems makes the dataset creation process more complex and time-consuming. Because we are dealing with raw images augmentation is very limited and there is also no concrete

way of creating synthetic data to extend these datasets which makes the process limited with the available dataset. Additionally the datasets available only capture daylight images which don't allow the model to process different conditions other than daylight.

In addition to the dataset problems the developed methods are much less flexible than the conventional ISP because of the models' design and the dataset forces the model to train to process specific settings and simulate a specific output. So you can't change the ISP setting like white balance, brightness settings, or color mapping and reproduce different outputs which is an important part of the photography creative process. With the current process, the model can only simulate the ISP process in a fully automatic mode.

The current models also struggle with accurate color reproduction and global tasks processing like white balance and color correction. They are only trained on image patches from the full images because of the challenging possibility of training on full images. So the trained models don't utilize the global information from the full image during inference as they don't have access to them during training.

These are the most effective general problems we found with the current state of Learned-Based ISP. Additionally, some solution-specific problems exist with some of the learned ISP models which we described in our comprehensive literature review in the next chapter 2.

### 1.3.2 Our Work Focus

For our work, we focused on 3 main problems that improved different aspects of the learned-based ISP process. For the first problem, we focus on the problem of rendering night images. As we mentioned before the datasets available don't include any night images so there is no available dataset to train a learned-based ISP that can render night images. The available solution depends on complex and heavy models utilizing datasets they created and not publicly available. We proposed our own solution utilizing the knowledge distillation of a pre-trained complex model to create an efficient learned ISP that can process night images without the need to create our own dataset. For the second problem, we worked on the problem of miss-alignment and the synthetic dataset creation. We proposed a novel pipeline that can be used to create a synthetic fully aligned dataset with the same phone raw image characteristics using a weakly aligned ISP dataset for the development of the pipeline. This pipeline can also be used to create a dataset from only RGB images of the DSLR camera which increase the size of the dataset and heavily decrease the complexity and time required to create ISP datasets. For the third problem, we worked on the problem of color reproduction and the lack of full image global information during training. We developed a novel process to train

with both image patches for better local task processing like texture reconstruction and with access to the global information from the full raw image for better global task processing like white balance and color correction. Our new process drastically improved the model performance and produced much better colors with a very small computation cost addition.

### 1.3.3 Contribution

- A comprehensive overview of the current state of Learned-based ISP. Challenges, Methods, and Datasets.
- An efficient ISP Network to render night images without data annotation.
- A novel pipeline to generate a fully aligned synthetic ISP dataset from a weakly aligned ISP dataset.
- The first approach to generate a high-quality synthetic training dataset for Learned based ISP.
- A novel color module that can integrate with any ISP network to combine the full raw image with the raw patch image during training for better local and global processing.
- State-of-the-art performance in Learned-based ISP with just the addition of our simple and efficient color module.

## 1.4 Thesis Outline

Because we worked on different problems and the limitations of the available dataset and its components every problem we worked on includes its own datasets, setups, experiments, models, and results. Even though all these problems are related to learned-based ISP and improve the ISP in different aspects but we are limited with the available dataset and its setup so we couldn't merge all the work as a single concrete result (For example the night rendering model works with different sensor data than the ISP dataset used with other problems). For better Clarification and a more comprehensive explanation, we described each problem with its experiments, setup, models, and results in its own chapter.

For the outline of our report, in the second chapter 2 we described the conventional ISP pipeline and its main components and tasks. Then we did a comprehensive review of the current state of learned-based ISP with the available solution and datasets. For the third chapter, Night Image Rendering 3, we talked about the

problem we worked on and we described our developed model and its comparison with the other night image rendering methods. For the Forth chapter, ISP Data Limitations 4, we described our work on the problems of the mobile ISP datasets and our proposed Pipeline to create a high quality fully aligned synthetic data with the same characteristics as the phone raw images. For the Fifth chapter, Color Reproduction 5, We addressed the problem of missing global information during training and we proposed our novel model to integrate both raw image patches and full images in the model for better global and local tasks processing. For the Sixth chapter, Discussion 6, we discussed the current state of the learned-based ISP and what we need to move forward, the future work, and the future of this approach as a replacement to the conventional ISP pipeline. For the Seventh and final chapter, Conclusion 7, we concluded our work and findings.

## 1.5 AI Tools

No AI tools were used in this project in either writing of the thesis report or the development of the methods. Grammarly online tool gra (2023) was used to check the grammatical errors in the thesis.

# 2 Background and Literature Review

In this section, we will first describe the conventional image processing pipeline (ISP) and describe the main stages in the pipeline. Then we will go through the previous work that was introduced in the learning-based ISP and the effort that was made to make it less dependent on the long process tuning and feature engineering and more data-driven.

## 2.1 Conventional Image Signal Processing Pipeline

Fig. 2.1 shows the basic conventional image processing pipeline for mobile devices. Because of the hardware limitations of mobile cameras from their compact size, they require much more computation and processing. So the image signal processing pipelines in mobile cameras are much more complex than the ones that can be found in professional DSLR and mirrorless cameras to overcome these limitations. The figure shows one of the possible combinations and tasks that can be found in mobile ISP. The tasks and the order of the tasks are different for different manufacturers and also some tasks can be combined together like joint demonizing and denoising Gharbi et al. (2016). Additionally, different manufacturers add more modules like image enhancement or super-resolution modules and other modules that can deal with different conditions like night imaging modules. So the ISP in each mobile camera is very dependent on each manufacturer, model, and also mobile camera hardware. For this section we choose the pipeline that was proposed by Delbracio et al. (2021) that includes the main tasks that will be found in some way or another in the majority of the image processing pipelines and with the basic order of these tasks.

The majority of ISPs in current smartphones are similar to this conventional pipeline as it is more practical, more mature for production, and well-researched.

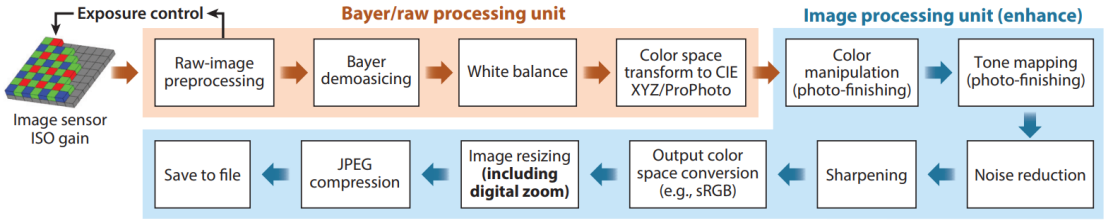


**Figure 2.1:** *Basic Mobile Image Signal Processing Pipeline. Image from Delbracio et al. (2021)*

There is a dedicated chip for The ISP with specific hardware operations that executes the full pipeline in milliseconds which can be hard to achieve the data-driven approaches that require more computation in most cases. Even though nowadays the algorithms for each one of these tasks became much better and more robust, this design still requires a lot of tuning and calibration. Additionally, it is executed in a sequence that results in error accumulation. In this part, we will describe the purpose of each task and the sequence the image goes through to produce the final image. After we will mention some of the effort that was made to improve this design.

### 2.1.1 Light Acquisition

Camera sensors consist of a 2D grid of photodiodes that convert photons (the light that hit the sensor in this area) into electric charge. To produce a colored image, color filters are placed in front of these photodiodes to correspond to low, medium, and high wavelengths similar to the cones that can be found in human retinas. This design is called a color filter array (CFA) which enables us to capture colored images using a single sensor design. Color filters are arranged as a mosaiced of color filters as shown in Fig. 2.1. There are different patterns that describe how the color filters are laid out on top of the photodiodes but the most common one are Bayer pattern. The process of converting these light measurements to the final image that we see is the purpose of the camera ISP. The first part of the camera ISP is to adjust the ISO gain factor that determines the sensitivity of the response of the photodiodes. This is done based on scene brightness, aperture, and shutter speed of the camera to obtain a well-light image. This produces the sensor Bayer frame which is called raw image. It is worth noticing that the raw RGB values are not in perceptual color space and it is specific to the CFA spectral sensitivities which is usually called sensor color space. This is why each camera sensor produces a unique raw image when they capture the same scene.

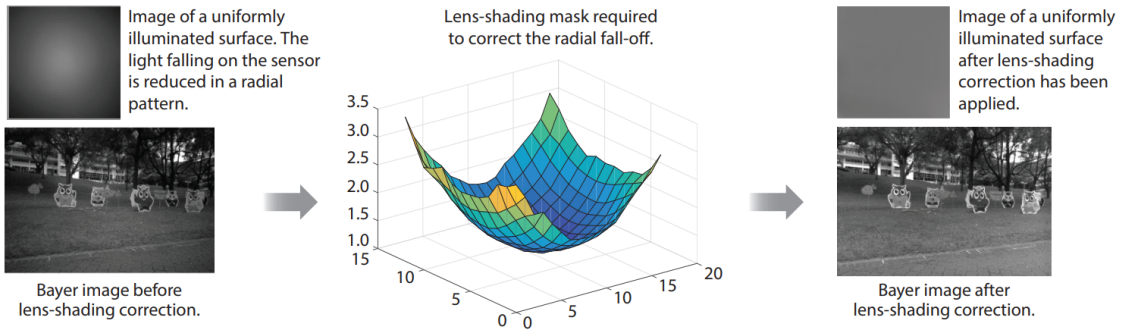


**Figure 2.2:** *Lens Shading process that is applied to remove the vignetting effect from the raw image. Image from Delbracio et al. (2021)*

## 2.1.2 Raw-Image Preprocessing

The next stage is the preprocessing of the raw image to prepare it for the next stages of the ISP. It starts by normalizing the raw image between 0 and 1 by applying black-level normalization. The black level represents the reading of the camera sensor when no light is hitting the sensor. This value is dependent on camera settings like ISO and obtained by a calibration process. The white level is also included in this process and represents the maximum sensor reading value. black level normalization is applied by subtracting the black level value from the raw image and dividing the raw image by the difference between the black level and the white level. Next is the correction of the defective pixels in the sensor. The manufacturer usually pre-calibrates the sensor and provides a defective pixels mask that is then interpolated from the neighboring pixels.

Additionally, at this stage, ISP applies lens-shading to correct the effect of vignetting (uneven lighting hitting the sensor because of the camera's optical system). The amount of light hitting the sensor becomes less and less as you go toward the edges which results in the center of the image being brighter than the edges as seen in Fig. 2.2. Camera manufacturers pre-calibrate a lens shading mask that is applied to the image to correct this effect. The process is shown in Fig. 2.2.

## 2.1.3 Demosaicing

Because of the design of CFA, each pixel in the raw image has only the information of one color of the RGB image. The demosaicing algorithm is concerned with interpolating the missing two colors at each pixel from the neighboring values. There is a big body of literature that tried to solve the problem of demosaicing using different methods Li et al. (2008). In previous years, researchers utilized deep learning to outperform the traditional demosaicing methods Liu et al. (2020);

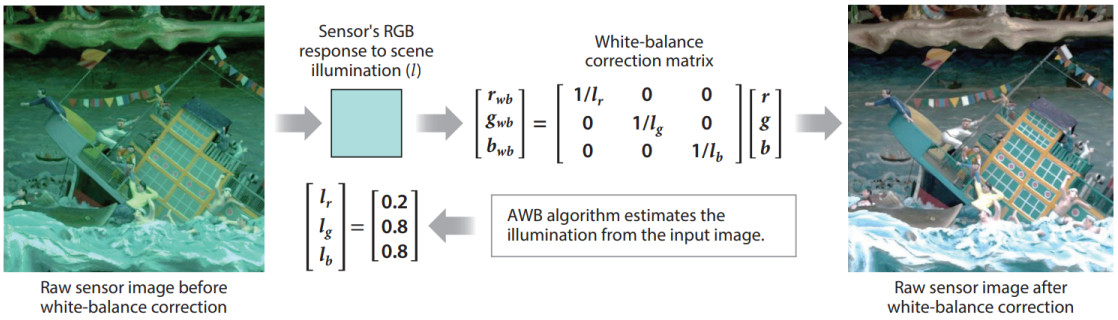


**Figure 2.3:** *The steps of auto-white-balance process. Image from Delbracio et al. (2021)*

Ehret et al. (2019).

## 2.1.4 White Balancing

The human visual system (HVS) performs chromatic adaptation to the scene illumination so that we can see the same color of the objects under different illumination conditions (color consistency). This is not the case for camera sensors as they are just light measurement sensors and they are dependent on the light that hit the objects in the scene. The role of White balancing is to mimic this process. White balance (WB) is applied by estimating the sensor's RGB response to the scene illumination and removing this response by dividing these values from the raw image. The sensor's RGB response to illumination can be precalibrated by measuring the sensor response to common illuminations like sunlight and fluorescent lighting and the user can choose the scene illumination which also can be used to give the image a different look and feel. Alternatively, the camera can rely on auto white balance (AWB) algorithms that will estimate the sensor response to the scene illuminant directly from the captured image which gives a more accurate reproduction of the captured scene. The process of AWB is shown in Fig.2.3. This is also called computational color constancy. This area of research are well investigated and there is a big number of methods that were developed utilizing the different attributes of the captured image Gijsenij et al. (2011). Additionally, deep learning is also utilized to solve this problem Hu et al. (2017) including the development of sensor-independent methods Afifi et al. (2021a) that can decrease the need for fine-tuning and calibration.

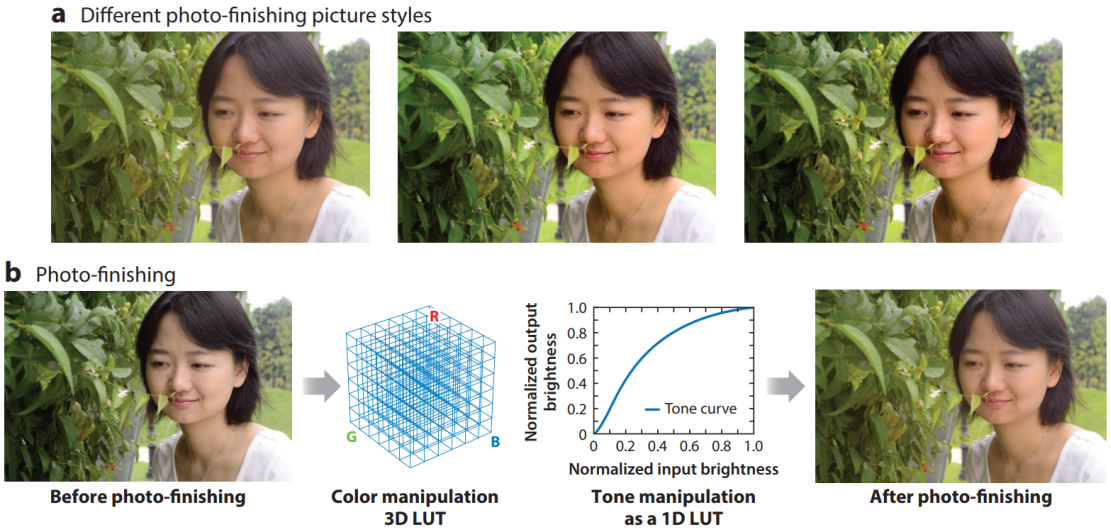


**Figure 2.4:** *Figure showing different image ascetics obtained after applying different color manipulations to the input image. Image from Delbracio et al. (2021)*

## 2.1.5 Color Space Transformation

As mentioned before the raw images are in sensor color space and up till this stage the image is still in the same color space. The purpose of this stage is to convert the image from the sensor color space to a device-independent perceptual color space. This color space is called the working color space that will be used to do the majority of the processing on the image to obtain the desired output look. Majority of the camera manufacturers use ProPhoto RGB color space Süsstrunk et al. (1999) as it is a wide-gamut color space and covers 90% of the visible colors Delbracio et al. (2021). This stage is done by computing a conversion matrix by utilizing a color calibration object, like a color chart.

## 2.1.6 Color Manipulation

At this stage, the ISP applies different color manipulations to enhance the visual ascetics and give the image a different look and feel which is usually called photo-finishing. These color manipulations are usually implemented as 3D look-up tables that are used to map the RGB values of the input image to different RGB values. Usually, cameras have different color manipulations that the user can choose from based on the preference and the content of the image, like vivid, landscape, or portrait photo-finishing. Some of the new cameras use scene understanding the decide the content of the image and choose the color manipulation based on

the content. Additionally, some color manipulations are done on a specific part of the image like enhancing the skin tone of the human in the image. These color manipulations are created by experts that fine-tune them to have a particular ascetics and look. The difference can be noticed between the different manufacturers as each manufacturer has their own look and ascetics based on their tuning. Also, some camera manufacturers tune the color manipulation parameters of the same camera differently based on the preferences of the users in the geographical location (The colors of the final image will have different aesthetics in Asia than in Europe).

### 2.1.7 Tone Mapping

This process is applied to adjust the tonal value of the image by applying a 1D LUT to the image. Tone mapping can be applied for different purposes like using tone mapping to change the image ascetics like increasing the image contrast as seen in Fig. 2.4. But the main purpose of tone mapping is to compress the tonal value of the image as the final output image is usually 8 to 10 bits because of the mobile displays but the raw image digital value can be from 10 to 14 bits. The design of the tone mapping is inspired by the adaptation of human eyes to scene brightness.

### 2.1.8 Noise Reduction

Noise reduction is applied to obtain a better visual quality image. While applying this stage you have to target a good balance for the intensity of the noise reduction as too much reduction will result in an artificial blurred look while too little reduction will result in visible noise in the image. The position of this stage is very dependent on the manufacturer design of the ISP as it can happen in the first stages or even jointly with other tasks like joint demosaicing and denoising Hirakawa and Parks (2006). It can also be applied multiple times throughout the ISP. This is a very complex problem to solve as the noise profile is dependent on everything in the capturing process including the camera setting like ISO and exposure and also the illumination conditions of the scene. There is a lot of work trying to create noise models that describe the noise distribution of the sensor under different settings and under different conditions. Also developing a calibration process that can be utilized to tune the parameters of the different models for the specific sensor. It is very important to develop a robust noise model that can accurately describe the noise to be able to remove it. This is also important for learning-based methods like deep learning-based methods that use noise models to create training datasets as it is very hard to create a real dataset that can account for all different conditions Zhang et al. (2022); Wei et al. (2020). There is a different method that tried to

create a general noise model Wang et al. (2022) that can work with different sensors without needing calibration but studies showed that sensor-specific noise models achieve better results and allow for more efficient noise reduction models Wang et al. (2020); Zhang et al. (2021a).

### 2.1.9 Output Color Space Conversion

The ISP working color space like wide-gamut ProPhoto color space has a bigger range of colors than the mobile displays can support. For that, the image is converted to a display referred color space that is supported by the mobile display like sRGB or AdobeRGB color spaces. That is usually applied using color profiles that contain a computed (though calibration) conversion matrix or LUTs that is used to convert between color spaces.

### 2.1.10 Image Resizing

The image is resized based on the output device or the user's preferences. Image resizing is not only limited to downsampling, it also includes upsampling like upsampling a crop of the image to provide digital zoom.

### 2.1.11 JPEG Compression

In the last stage, the image is compressed using JPEG compression standards to reduce the size of the image and maintain the perceptual quality of the image.

## 2.2 Learned Based ISP

The increased use of deep learning inspired the use of deep learning to solve different tasks in the ISP pipeline like demosaicing, denoising, and white balancing. But instead of only utilizing deep learning networks for specific tasks in the conventional ISP pipeline another direction arose to reconfigure the conventional ISP pipeline and target a learned-based end-to-end solution that requires no calibration or tuning. This can be achieved by creating a raw to RGB dataset that represents the desired output and creating a deep learning model that will learn the mapping from the input raw to the desired RGB output. This task is very challenging as the model needs to solve different problems and apply different local and global modifications to the input raw image to achieve the desired output with the desired look.

Schwartz et al. (2019) created a DeepISP model that tried to achieve this mapping. They created a mapping dataset using Samsung s7 phone by capturing images under normal lighting conditions (GT) and simulating low-light raw images by decreasing the exposure time (Input). They utilized the images processed by the mobile ISP under with good exposure as ground truth and raw images with low exposure as the input. They created the dataset that way to prove that learned-based ISP can provide better results than the ones created by the internal conventional camera ISP. The model they created consists of 2 parts the first part work on the full image patch and applies local modification to the input, and the second part encodes the features from the features learned from the first part to create a feature vector that is used to apply global modification. For the training process, they used a mix of l1 loss applied to the images in the LAB color space and SSIM loss. For evaluation, they used mean opinion score (MOS) based on the user's evaluation. They compared the Samsung ISP processed poor light raws, with the Samsung ISP processed well light raws (GT) and DeepISP processing of the poor light raws. Their results showed that DeepISP can generate better results than Samsung ISP with poor light raws and produce results close to the raw captured in a well-light environment. These results showed the ability of the deep learning model to process raw images without the need for much tuning or calibration.

Differently Liang et al. (2019) reformulated the ISP as 2 stage network. They proposed CameraNet, which consist of 2 stage framework that split the different task of the ISP into 2 networks as according to them, there are different tasks in the ISP that are uncorrelated and it will be better to separate them to be processed with separate networks. The first network is the restoration network which takes the raw image and restores it to a full linear unprocessed image in CIEXYZ color space. It works on different restoration tasks like demosaicing, denoising, and white balance. The second network is EnhanceNet which takes the CIEXYZ image after converting it to an sRGB image and applying image enhancement like tone mapping, and color manipulation. Both networks are UNet-based networks Ronneberger et al. (2015) with attention modules. Each network is trained separately and then trained jointly in an end-to-end fashion. they created the dataset by collecting different raw images from different datasets and using photo editing software to manually process the raw image to obtain the CIEXYZ image then applying different enhancement filters to obtain the final enhanced ground truth. Their results showed the effectiveness of separating the uncorrelated ISP tasks into 2 different Networks and showed fewer artifacts and better results compared to 1 stage architectures like DeepISP Schwartz et al. (2019). But the dataset creation process is based on the style of the photofinishing that was done and doesn't optimize the image for accuracy or reproduction.

Ignatov et al. (2017) proposed the idea of image enhancement for mobile cameras as a mapping from the image captured by a mobile phone to the image captured by a professional DSLR camera from the same scene. They created the dataset by capturing the same scene using a mobile camera and DSLR Camera and used an alignment algorithm to get patches that match in both images to learn the mapping. Then they designed a deep learning model, utilizing this dataset, to learn the mapping from a hardware-limited mobile camera to a DSLR camera to overcome the hardware limitations of the mobile camera's compact size. This dataset was created to work with the RGB output of the mobile camera that was already processed by the camera's internal ISP. These images lack a lot of raw information that will be very useful for this mapping. To extend this idea for learned-based ISP a new dataset was created but instead of working with the mobile RGB output, you work with the raw image of the mobile camera. So by this approach, the problem of learned-based ISP was reformulated to process the mobile raw image to imitate the output of the DSLR camera with much better hardware and much better scene reproduction accuracy. This is done by learning raw to RGB mapping but to the output of the DSLR camera. This approach allows us to overcome the hardware limitation of the mobile cameras with a software computation process and not limit the ISP processing to the hardware of the mobile camera.

This work was first introduced in Ignatov et al. (2020b) with the first dataset using this approach. They proposed Zurich Raw to RGB (ZRR) Dataset that consists of raw images captured by Huawei P20 phone as the input raw and RGB images captured by Canon D5 Mark IV as the ground truth. Additionally, They proposed PyNet Network (Fig. 2.5) to learn the raw to RGB mapping. The proposed network is a multi-stage network that works on different receptive fields. It is an inverted pyramid design where the lower stage has the biggest receptive field and decreases as you go to the higher stages. This design allows the model to apply both global and local modifications with the same architecture in an end-to-end manner. the lowest stage work on down-scaled images by a factor of 16 and targeted for global features like gamma correction, brightness, and global colors. The highest stage work on the original image scale and target local details like texture, noise removal, and local colors. For training, they utilized progressive training where they start training the lowest stage and add more stages progressively until training all stages together. Additionally, the loss function depends on the stage being trained. For the stages with the big receptive field, they choose loss functions that focus on global features like MSE and perceptual loss (VGG Loss). For higher stages, losses focused on local features like SSIM loss were used. Their multi-stage architecture and progressive training allowed the network to apply local and global modifications and achieve state-of-the-art (SOTA) results compared to the other architectures they tested.

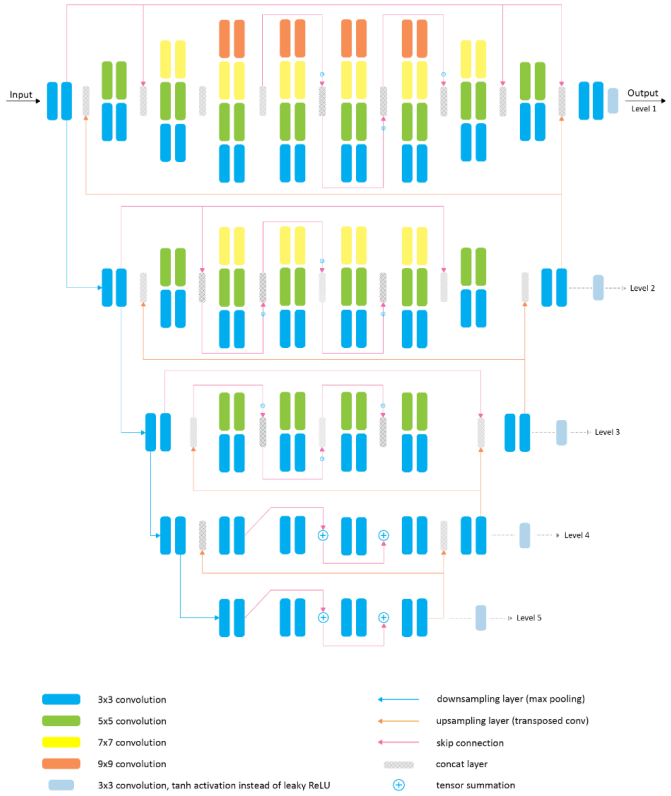


**Figure 2.5:** *Architecture Details of PyNet Model. Image from Ignatov et al. (2020b)*

To push the research in this direction the authors of Ignatov et al. (2020b) proposed the AIM 2019 Challenge on RAW to RGB Mapping Ignatov et al. (2019a). They used the same Zurich Raw to RGB (ZRR) Dataset. For the evaluation process, they had 2 tracks first fidelity track using SSIM and PSNR, and the second track the perceptual track which uses MOS by comparing the solution-processed image in comparison with the DSLR image by a scale of similarity. The winning solution for this challenge is W-Net Uhm et al. (2019) which produced the best score in the fidelity track. Their solution consists of 2 cascaded UNet architectures. The first one is used to reconstruct the input raw image and the second one is utilized to refine the output of the first Network. For their arch, they used a channel attention module to increase the receptive field and improve the global adjustments. For training, they used a mix of l1 loss, VGG perceptual loss, and color loss. Mei et al. (2019) proposed HighEr-Resolution Network (HERN) which achieved the best result in the perceptual track. Their network consists of a dual-path network to process global and local information separately. The first

path consists of encoder-decoder architecture that increases the receptive field which gets better global features and reduces the processing power. For the global path, they processed the input image patch in full size without downsampling to focus on texture and edges as downsampling destroys these details. Finally, they employ a Pyramid Full-Image Encoder to generate a high-level feature vector that represents a global receptive field that can be used to apply regulation to the output image for better construction. For training, they used l1 loss with progressive growing training by training with a small resolution and increasing the resolution throughout the training. This results in a better conversion time. One of the other top solutions was Zhao et al. (2019) which trained a GAN-based network to apply the reconstruction. They also utilized a saliency map as a guide for the network. They used a pre-trained network to generate the corresponding saliency map and trained a separate generator to learn to generate them. Their solution struggled with the local details which are the case with generative-based solutions in reconstruction applications.

Learned-based ISP challenge continued with AIM 2020 Challenge on Learned Image Signal Processing Pipeline Ignatov et al. (2020a). For this year the winning solution in the perceptual proposed multi-level wavelet ISP network (MW-ISPNet). They used a UNet base architecture with residual channel attention blocks (RCAB) Zhang et al. (2018b). They also replace the standard downsampling and upsampling operations with a discrete wavelet transform-based (DWT) decomposition. The second place in the fidelity track is the AWNet model Dai et al. (2020). They proposed a UNet-based architecture with channel attention. They used discrete Wavelet transformation on the features to get the high and low-frequency details as additional guidance to the network and used it for downsampling and upsampling as well. Their final architecture consists of 2 networks, the first one is trained on the raw image which makes the model focus on the reconstruction of high details, the second network is trained on demosaiced images which makes the model focus on the color mapping between the input and the output. They trained the 2 networks separately and averaged their output. For the fidelity track, the winning solution used an ensemble of 6 different ISP architectures. PyNET-CA Kim et al. (2020) was one of the top solutions in this competition which is an incremental improvement over PyNET Ignatov et al. (2020b). They added Channel attention to the architecture for better global feature extraction and cleaned the dataset from the miss-aligned and moving objects samples by screening out the images with a large area of reflection and moving objects.

One of the latest research in the learned-based ISP with this approach and the current SOTA is Shekhar Tripathi et al. (2022). They focused on the problem of color misalignment between the image pairs which results in bad color reproduction in the ISP output. The architecture diagram in Fig.2.6. Their architecture consists

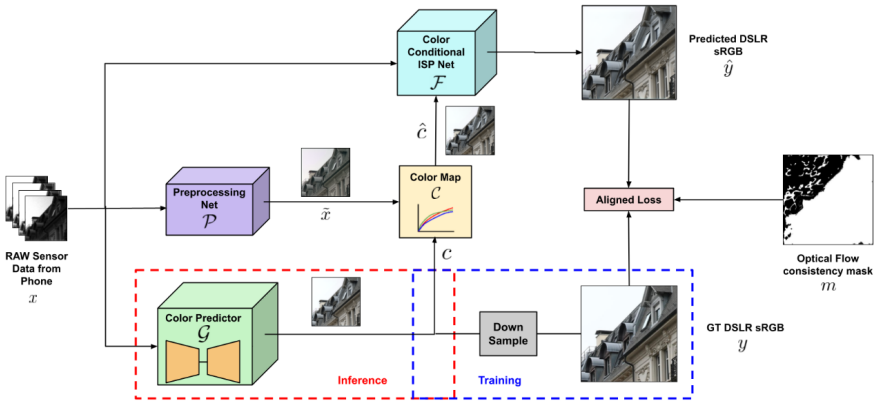


**Figure 2.6:** *Architecture Details of Learning the ISP in the Wild Model. Image from Shekhar Tripathi et al. (2022)*

of 4 different modules. The first one is the Color prediction network. This network aims to produce a low-resolution image of the input raw with the same color characteristics and dynamic range as the DSLR image ($\hat{x}$). The second module is the Pre-processing network. it is used to provide a cleaned simply processed version of the input raw ($\widetilde{x}$). This is done by simple demosaicing, gamma correction, and denoising using a CNN model. The third module is the Color Mapping module which is used to calculate an affine transformation mapping that maps from the processed raw image ($\widetilde{x}$) to the raw image with the DSLR color characteristics ($\hat{x}$). The last module is the ISP Network. The processed raw image ($\widetilde{x}$) with the color mapping applied is used as input to the ISP Network as guidance. They also utilized miss-alignment loss Zhang et al. (2021b) to reduce the effect of miss-alignment in the dataset. These extra modules that focus on learning the color difference between the input raw and the DSLR GT improved the color reproduction accuracy and the performance of the ISP Network. But all these modules added a large computational cost to the final ISP pipeline.

The biggest problems with the approaches described before they are very computationally complex and require a lot of resources for inference. Even though they produce better results than the conventional ISP in some cases, they are not feasible to use in production because of the computational requirements that will not allow them to run on mobile hardware. This shows the importance of more efficient and less complex architecture to produce solutions that will be possible to use in practice.

## 2.3 Efficient Learned Based ISP

With the big increase in using deep learning models in different tasks including tasks for mobile phones good adaptation became required. Mobile processor manufacturers like Arm and Qualcomm started adapting deep learning models in their System on Chip (SoC) and including specialized AI silicon in mobile SoCs. Nowadays the majority of mobile phones especially high-end phones have AI silicon in their SoC and even some phones include a separate chip designed specifically for deep learning and AI computations. This current and ongoing improvement in mobile computational hardware allows mobile phones to run more computationally demanding algorithms and allow pushing the limit of what is possible on mobile phones. But still, because of the compact size of the mobile phone, there are still computation limitations so you need to develop efficient and highly integrated models to be able to run them efficiently.

An important thing to keep in mind while developing a deep learning model for a device you need to consider what operations the mobile chip support. Additionally, you need to consider what operations mobile deep learning frameworks and APIs like TensorFlow Lite and Android NNAPI support because they are the gate between the application and mobile hardware. If you used an unsupported operation it will not run on the chip GPU which will end up increasing the execution time significantly. Because of these reasons, we need a benchmark to test how well different hardware perform with different deep learning model and task. AI Benchmark presented to solve this problem Ignatov et al. (2019b). AI Benchmark is a mobile application that can be installed on a mobile phone and tests the mobile hardware speed and ability to run different deep learning models for different tasks like image classification, detection, and enhancement. The benchmark applies 21 different tests and gives a score that represents the ability of the hardware for AI applications. They also provided the ability to import your own model to the benchmark and test it on your mobile hardware. They also provide different options for inference of the model like different inference APIs and different floating points. After running the benchmark they give you a detailed description of the inference information with time and memory consumption. These additions speed up the testing process of deep learning models on mobile devices and test the compatibility with different mobile hardware.

To push the research of learned-based ISP towards solutions that can work efficiently on mobile hardware, the leaned ISP challenge changed to include the efficiency of the proposed solution. The evaluation criteria changed to include the inference time in a specific chip (Dimensity 1000+ APU). So the new challenge final score is based on the quality of the output and inference time. The new challenge focused on developing more practical solutions instead of only focusing

on the accuracy of the output like in the previous challenges. The challenge was renamed Learned Smartphone ISP on Mobile NPUs with Deep Learning, Mobile AI 2021 Challenge Ignatov et al. (2021). Additionally, they changed the dataset of the challenge to include new sensors. They used Fujifilm GFX100 102 MP camera to capture the target images and Sony IMX586 Quad Bayer mobile camera sensor for raw input images. The mobile sensor was attached to the Fujifilm camera to shoot the photos synchronously to ensure the same content in both images.

The winning solution of this competition was SmallNet ISP. The model is a very shallow CNN architecture that consists of 3 CNN layers with 16, 16, and 12 channels respectively. After they used a pixel shuffle layer for upsampling to get the final result. Even though the solution didn't obtain the most accurate results (still very close), because of its small size of the model, it was the fastest model in the competition. The performance of this small model showed that with the right layer choice and training process you can achieve state-of-the-art results with very shallow models. CSANet Hsyu et al. (2021) was the winner of second place in the competition. The model consists of 3 parts, the first part is the downsampling part by utilizing a convolution layer with stride. The second part is the feature extraction part with double attention modules (DAM). DAM is the main processing block in the architecture that contains both spatial attention and channel attention modules. The last part is the upscale part using transpose convolution and pixel shuffle. Downsampling the image before feature extraction and utilizing DAM which contains lightweight attention modules for feature extraction increased the speed of the network. This gave the model a good mix between speed and accuracy.

The challenge continued the year after with Learned Smartphone ISP on Mobile GPUs with Deep Learning, Mobile AI, and AIM 2022 Challenge Ignatov et al. (2022c). For this year they used Qualcomm Snapdragon 8 Gen 1 mobile SoC as the target runtime evaluation platform. They also added a new track to evaluate the model based on their perceptual results using MOS without time constraints. For the first track (efficiency track), the overall winning solution is the enormous Re-parameter Convolution (eReopConv) model. The model uses blocks with a very large number of layers and then parameterizes them to a smaller block for inference. For their model, they used a training block with 10 branches and with different convolutions and different kernel sizes. For inference, they used a block with 1 convolution layer. They reparameterized the parameters of the training block to the inference block using linear transforms. By this approach, you give the model the opportunity to learn rich features from the images during training and then transfer this knowledge to smaller blocks that will be used during inference. For the second track (perceptual track), a 3 module model was proposed. The first module is Source Features Module to extract rough features from the input raw. The second module, Enhance Features Module, is for dense feature extractions using

lightweight multi-level feature extraction blocks. Lastly, Upsample Features Module generates the output in the desired size. They also utilized the re-parameterization technique though out their architecture to obtain an efficient model for inference.

As part of The challenge organizer work, different approaches for efficient learned-based ISP following the same dataset and evaluation criteria were proposed. One of state of the art architecture for learned ISP is PyNet Ignatov et al. (2020b). The problem with this architecture is its big size which makes it not usable with mobile hardware. A new lightweight version was developed, PyNet-V2 Ignatov et al. (2022a), that can process 12 MP images. They used the same multi-stage idea to process local and global operations at different stages. The model consists of 3 stages inverted pyramid network and each stage processes the input image with a different scale similar to PyNet Ignatov et al. (2020b). For the new architecture, they avoided large conv layers. They achieved that by only using 3x3 convolution layers in their model and using grouped residual blocks. These blocks split the input feature map into different convolution branches to decrease the feature map size of each convolution layer. This allows the branches to work in parallel which decreases the computational cost. For training they used the same multi-iterative training process as PyNet Ignatov et al. (2020b). The new model achieved close results to the original model with much faster inference time and much less memory consumption which allows the processing of high-resolution images.

A second approach proposed is an improvement over CSANet Hsyu et al. (2021) AIM 2021 winning solution. LAN: Lightweight Attention-based Network Raimundo et al. (2022). First, they replace the slow space-to-depth operation that converts the input raw image into 4 different channels for each color filter by a convolution layer with stride. This solves the problem of channel misalignment that happen when we split the raw image into 4 different channels. One of the biggest problems of the dataset used is the misalignment in the data. Because the data was collected with 2 different sensors and optical systems it is not possible to achieve perfect alignment which results in artifacts and miss-alignment in the dataset. To solve that issue the authors proposed a pretraining process by initially training the mode to demosaic the input raw image by its classically demosaiced image. By this, the model learns image reconstruction on aligned images which will help the model deal with the misaligned data. The modification applied on CSANet Hsyu et al. (2021) increased the performance of the model in both accuracy and efficiency.

One of the most important elements of processing high-resolution images on mobiles is ram consumption. For mobile inference, ram consumption is decided by the biggest convolution layer in your architecture. MicroISP Ignatov et al. (2022b) was developed around this idea by using the smallest convolution layer possible to process the highest resolution image possible. Model architecture can be seen in Fig. 3.3. The model consists of 3 branches each branch is responsible for the

reconstruction of one color channel of the final image. Utilizing this speculation, they can achieve the lowest number of filters possible for the used layers. For all the layers in the model, the number of filters is 4 which is the lowest number of feature maps possible to reconstruct the final image size (3, W, H) (double the input size (4, W/2, H/2)). They also used a channel attention block following the same layer size constraints for better global modifications. The network design they proposed allowed them to process 32 MP images in less than 1 second on MediaTek Dimensity 1000+ GPU with accuracy very close to state-of-the-art methods with much bigger size and memory consumption.

## 2.4 Night Image Rendering

Another important part of the image signal processing pipeline is rendering low-light and night images. For better night images it is important to differentiate between well-light images and low-light images or night images as the latter requires more tuning in the settings and much finer processing as they are more challenging to process. For example, when the phone detects low light conditions it tunes the phone to increase the exposure to capture more light. This can be achieved by increasing the ISO which introduces more noise or increasing the exposure time which introduces blurring and motion artifacts or increasing the aperture which decreases the area in focus. So it is important to deal with all these issues during processing which require developing additional processing for low-light images. This includes specific low-light modules and algorithms specific for the night images. Like algorithms developed to decide the capture setting in this environment which is very crucial for the final images. Additionally can include tone mapping and relighting algorithms to improve the brightness of the images. In addition to the extra modules some of the common tasks in all images require more attention and better processing in the case of light images. We can notice that noise is more in low-light images with different patterns than in well-light images. Similarly, illumination estimation is much more complex in low-light images because of the multiple illuminations and more complex and variant illumination maps. All that shows the importance of separate handling for night images as systems optimized for normal well-light images will not work well with this kind of image. Fig. 3.1 shows the different aesthetic issues of the night images. The developed learned-based ISP usually deals with well-light images only because the dataset created for this task only includes well-light images. This limits the learned-based ISPs' performance on low-light/night images.

To better study night image processing a Challenge was introduced in NTIRE 2022 workshop on Night Photography Rendering Ershov et al. (2022). The challenge is concerned with developing an ISP develop specifically to process night images

and produce a pleasing output. The challenge provides 50 raw night images with some meta date and asks the contestants to develop ISP for these raw images. Then they use other raw images from the same camera to test and choose the best solution. This challenge doesn't provide any supervised data which makes the challenge much harder and gives the contestants the freedom of the proposed solution (supervised vs unsupervised, classical vs neural network etc). The solutions in this contest were mostly split into 2 categories. The first category solutions created their own supervised dataset by creating an output GT by processing the provided night raw with photo editing software and a professional editor. The second category created an ISP with a sequence of different modules some of them general modules like demosaicing, denoising, ..etc, and other night-specific modules like tone mapping and low light enhancement similar to the conventional ISP approach. The contestants' solutions were tested using Mean Opinion Score MOS by conducting an online subjective experiment. Then the top 10 MOS solutions are evaluated by a professional photographer and the final ranking is decided.

The top solution of the year 2022 was part of the first category with a solution called Deep FlexISP Liu et al. (2022). The solution consists of 3 stages by separating the parts that suffer the most (Denosing and white balancing) in the night image rendering from the reconstruction network for a better rendering. The first stage is a raw denoising network with its own collected data. The second stage is the white balancing network using FC4 white balancing architecture Hu et al. (2017) trained on color Checker Dataset Gehler et al. (2008) and the NUS 8-Camera Dataset Cheng et al. (2014). The third stage is the reconstruction by a Bayer to RGB Network which was trained on the dataset they created by manually processing the given night raw images to create a GT. The third stage network is a modified version from MW-ISPNet Ignatov et al. (2020a). The second-place solution is part of the second category with a multi-handcrafted pipeline Li et al. (2022). The pipeline starts with denoising using the winning solution of NTIRE2020 RAW image denoising MWRCANet Abdelhamed et al. (2020). Followed by a demosaicing step using simple bilinear interpolation Then a white balance step using CAUnet from the illumination estimation challenge Li and Ma (2021). Afterward, they use the meta-data provided by the competition to apply the color correction. For the tone mapping they used 2 solutions the first one is the self-supervised Unpaired-HDR-TMO Vinker et al. (2021) model. The second solution by training their own network in a human-labeled dataset they created. to choose between the two solutions, they used an evaluator using Resnet34 He et al. (2016) to choose the best output. The cascade of modules for the top solutions consists of different and heavy models that make the pipeline not efficient at all and not suitable for real use in cameras. This is expected as for the challenge there were no efficiency constraints. Another approach was proposed by Zini et al. (2022) based on the

conventional ISP techniques. Their ISP consisted of the baseline steps provided by the challenge organizers followed by night-specific enhancement modules. the night-specific modules consist of a Local Contrast Correction algorithm followed by contrast and saturation enhancement then black stretch operation and denoising. Their approach was targeted for shallow and traditional processing that can be efficient for camera use but the performance was less pleasing than the other big non-efficient modules.

The same Challenge was proposed again in NTIRE 2023 workshop with the same camera sensor but with a new dataset and new solutions Shutova et al. (2023). This challenge was conducted during the master thesis timeline and we took part in it as part of the master thesis work. Our developed solution was one of the top 10 solutions and was featured as part of the final competition report. We described our experiments and our solution in detail in the Night Image Rendering chapter 3. The winning solution for this competition was proposed by the same team as the previous year's winners. It is an improved version of their solution in the previous year Liu et al. (2022). For their solution, the first 2 stages remained the same and the Bayer to RGB stage split into multiple modules. The modules consist of demosaicing, another White balance stage in the RGB stage for correction using the shade of gray algorithm. Color space transformation is then applied using the image's metadata then a tone mapping using a fixed curve. Finally, an enhancement model based on MW-ISPNet Ignatov et al. (2020a) is applied. The network is trained on the dataset they created by manually processing the provided night raw images. As we notice the solution is almost the same as last year with the addition of preprocessing steps before the construction part and with complex modules that are not suited for mobile hardware. The second place by Zini et al. (2023) was also an improved version of last year's previous work. The pipeline was developed using conventional ISP modules by using the same pipeline as last year with additional modules or improving the existing ones for better night processing like better tone mapping and white balance.

## 2.5 Datasets

In this section, we will talk about datasets available for Mobile Learned ISP. There are a lot of available raw datasets used for different tasks like denoising, relighting, and the components of the dataset depend on the target task. The majority of available datasets for raw reconstruction only have the input raw and the output of the same camera ISP. Also usually these datasets are captured with DSLR Cameras which have much fewer limitations than mobile cameras and much simpler ISP. Example MIT5K dataset Bychkovsky et al. (2011) which has the raw images of different DSLR Cameras and the output created by professional photo editors.

**(a)** *Phone RAW Visualized* **(b)** *Huawei P20 ISP* **(c)** *Canon 5D Mark IV*

**Figure 2.7:** *Example of full images from Zurich RAW to RGB dataset Ignatov et al. (2020b). As we notice from the full images there is a receptive field difference and miss-alignment between the full images.*

But our work focuses on developing a learning-based ISP for mobile cameras by imitating the output of professional DSLR cameras. For this kind of reconstruction, the datasets available are very limited and hard to create. This section will mention all the available datasets for this task.

The first dataset created for this task was Zurich RAW to RGB dataset in 2018. It was proposed as a dataset to develop a new approach to replace the conventional ISP with an end-to-end deep learning Network Ignatov et al. (2020b). For the DSLR camera, they used Canon 5D Mark IV camera with Canon EF 24mm f/1.4L fast lens and Huawei P20 smartphone as the mobile camera both in automatic mode and default settings. An example of the dataset can be found in Fig. 2.7. For the dataset, they collected 20 thousand images with the same scenes for both the DSLR and the mobile phone with a variety of scenes, illuminations, and weather conditions. Even though they capture the same scene the captured image pairs have a big miss-alignment and are not suitable to be used directly for this task. This miss-alignment is because of using 2 different camera systems with different optical systems and different sensors. To prepare the data the authors aligned the image pairs using SIFT keypoints detector Lowe (2004) and RANSAC algorithm then split the image pairs into patch pairs. This was done by two sliding windows moving in parallel in the two images. The window in the DSLR image was slightly adjusted with a small shift and rotations to maximize the cross-correlations between the patch pairs and account for the miss-alignments that still exist. To ensure that only patches with a good alignment are chosen for the dataset the authors used a cross-correlation threshold of .9 for the selected patch pairs in the final dataset. This process resulted in 48k Raw-image pairs (training/validation (46.8K) and testing (1.2K)) with a size of 448×448×1 for the phone raw image and 448×448×3 for the DSLR RGB image. It is important to notice all the modifications applied (alignment warping and patching window shifting) were only applied on the DSLR

images without any modification to the phone raw image not to destroy or timber with any of the raw information in the file. The online available data only contains the image patches for the training/validation part. But for the test set the authors provided the full images (148 images from each device) of all devices before the alignment (full phone camera raw, full phone camera ISP output, full DSLR RGB images) in addition to the aligned image patches. This dataset was used for the AIM 2019/2020 challenge on the learned image signal processing pipeline.

The second dataset is Fujifilm UltraISP Dataset Ignatov et al. (2022b) which was created as an upgrade to Zurich RAW to RGB dataset with upgraded sensors and better quality cameras. For the professional GT camera, they used Fujifilm GFX100, a medium format with a 102 MP sensor. This camera provides very sharp high resolution and noise-free GT images with high dynamic range and good low light performance capturing which make it more than suitable as a GT for the required task. For the phone camera, the Sony IMX586 Quad Bayer camera sensor attached to MediaTek Dimensity 820 development board was used. The Sony IMX586 sensor is very popular and can be found in different mid-range and high-end cameras which makes it suitable as a representation of the current phone camera's hardware. The camera sensor is used to capture both the phone's raw images and the raw images processed by the internal ISP MediaTek Dimensity 820 chip (used as a baseline and comparison to the network output). Both cameras were attached together using developed software to ensure a synchronized image capture to make sure of identical content between the image pairs, especially with the moving objects (this was a problem in the previous dataset). They captured 6k image pairs in daylight across several weeks with a good variety of scenes, weather conditions, and illuminations. To align the image pairs the authors used a SOTA dense matching algorithm based on deep learning from Truong et al. (2021) to insure accurate pixel-wise alignment. Then the images split into patches of size 256×256 directly without any additional processing and without any patch pairs elimination. The dense matching achieved an accurate pixel-wise matching that didn't require patch elimination and increased the dataset size immensely compares to the method used in the previous dataset. But the dense matching still resulted in some warping problems as we see in Fig. 4.1. Additionally applying dense matching in this size of images requires a huge amount of hardware resources which is not usually available. Similar to the previous dataset all the alignment operations were applied on Fujifilm RGB images to ensure unmodified raw images. This process resulted in 99k image patch pairs split into training (93.8K), validation (2.2K) and test (3.1K). The dataset was made available as part of the 2021/2022 Learned Smartphone ISP on Mobile NPUs with the Deep Learning challenge. Only the training set patches were made available to the competition contestants and the validation and test sets weren't made available. Instead, the methods were

evaluated on these sets using a remote server to keep the sets hidden from the participants. During our master thesis work the computation was not running and the submission server was not available which made it not possible for us to use this dataset. Because of the missing parts in the dataset, it is nonusable outside of the competition purposes.

The last dataset is a new one called ISP in the Wild (ISPIW) Shekhar Tripathi et al. (2022) and it was made available in the middle of 2023. It is very similar to Zurich RAW to RGB dataset with some new hardware. For the DSLR Camera, they used the same camera, Canon 5D Mark IV DSLR camera with a lens of focal length 24mm. But they used a new phone camera with Huawei Mate 30 Pro mobile phone which is a new up-to-date phone. For the DSLR settings, they use an ISO of 100 to ensure less noise and a small aperture to ensure better sharpness and automatic mode for the mobile phone. They also captured different images for the same scene with the DSLR with different exposure values (EV -1, 0, 1) but only used EV 0 for the created dataset. The authors captured a total of 200 images for the dataset and split them into 160, 20, and 20 for training, validation, and testing. To align the image and create the patches for the dataset they followed the same process as Zurich RAW to RGB dataset but they used a much lower cross-correlation threshold for the eliminated patches (.5 threshold). This results in a more miss-aligned dataset but a bigger number of patches. Additionally, they used a smaller patch size of $320 \times 320$ with an overlap between the patches (stride of 160) to increase the number of patches because of the small number of images. This dataset is by far the most comprehensive with respect to the available components. The authors provide all the full images for the raw image and the DSLR images but lack some others like the output of the phone ISP and the output phone raw in the original format.

As we presented there are not many options for this kind of task we are working with. The datasets are very limited, and not all the parts of the dataset are publicly available which makes it limited in its use. This additionally limits the ideas that can be applied to this task. Additionally, the dataset is focused on one-to-one mapping without any variety in the setting (only automatic mode images). This makes the developed method not able to process different raws generated with different setting which limit the creative part of taking photos with the phone. All the datasets are captured in the daytime, limiting the developed models' performance in challenging conditions like low light or night images. A more comprehensive and variable dataset is much needed with a variety of settings, lighting conditions, and scenarios. That will allow the development of a robust and flexible ISP similar to the conventionally used ISPs without needing the same amount of tuning and calibration.

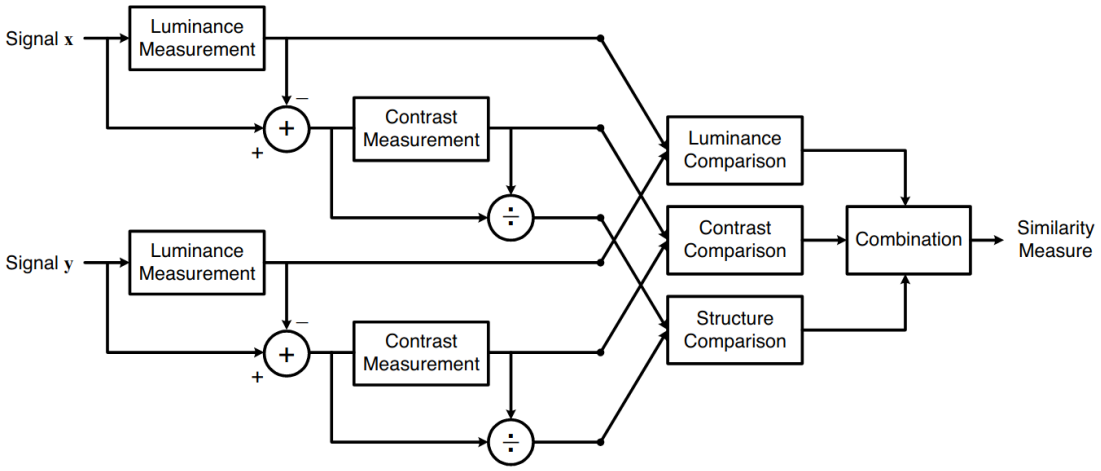


**Figure 2.8:** *The diagram of the structure similarity measurement metric. Image from Wang et al. (2004)*

## 2.6 Evaluation Metrics

The most commonly used metric to compare the model's output to the GT in learned ISP networks is the peak signal-to-noise ratio (PSNR). It is used to quantify reconstruction quality metric between lower-quality images (the output of the ISP Network in our case) and the high-quality GT. The higher the ISP the closer its to the GT the better the quality is. It is similar to Mean Square Error as it applies the comparison pixel-wise so any miss-alignment will largely affect the evaluation. The difference is MSE computer the cumulative error between the image pair, but PSNR represents a measurement of the peak error. It is calculated using the following equation. First We calculate MSE between the image pair and use it to calculate the PSNR.

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij} - y_{ij})^2 \tag{2.1}$$

$$PSNR = 10\log_{10}(\frac{R^2}{MSE}) \tag{2.2}$$

Where $m, n$ is the height and width of the image. $x, y$ is the image pair. $R$ is the max value of the image which is 255 in the case of 8-bit images.

The second and also commonly used metric is the Structural Similarity Index (SSIM) Wang et al. (2004). It is a full reference metric (similar to PSNR) that is used to measure the perceptual difference between image pair of the same scene. Different from PSNR, SSIM is based on the comparison between the visible

structures in the images and not just pixel comparison between the image pair. We can see the diagram of SSIM in Fig. 2.8. As we see from the diagram the SSIM is calculated from a comparison between 3 different components extracted from the images, luminance (l), contrast (c), and structure (s). This comparison is applied between the statistics of 2 windows from the image pair which make it less affected by miss-alignment in comparison to the pixel-wise comparison. For our experiments, we used the open-source implementation of SSIM from scikit-image van der Walt et al. (2014) python library to evaluate our models.

For last, we used a Color Difference CD metric to better evaluate the color difference between the reconstructed image and the GT. We chose CIEDE2000 Luo et al. (2001) as our CD metrics based on the evaluation study in Wang et al. (2023). They evaluated 33 different CD metrics on perceptual color differences in natural images. CIEDE2000 was one of the best-performing metrics with good robustness against miss-alignments between the compared images. CIEDE2000 is a perceptually uniform color metric that is calculated between 2 different colors in the LAB color space. Following the study from Wang et al. (2023) we used the available Python implementation from Ortiz-Jaramillo et al. (2019) to calculate our CD metric for our evaluations.

# 3 Night Image Rendering

In this Chapter, we will talk about our proposed solution for night image rendering. This solution was developed and submitted to NTIRE 2023 Challenge on Night Photography Rendering Shutova et al. (2023).

## 3.1 Problem definition

Night image rendering is concerned with the raw image Reconstruction of the images that were captured under challenging lighting conditions, especially at night. It is much more challenging to deal with images that are captured at night because of the dark environment which will require increasing the exposure of the camera to collect enough light to capture the scene. Increasing the exposure is done by either increasing the camera ISO, exposure time, or camera aperture. This increases the noise level and can introduce different problems like light flare and motion blur. Because of the nature of the night images we need to address the processing of this kind of images separately as a normal processing pipeline developed for day images will not work effectively. For example, in night images there are more different light sources with different colors, intensities, and directions which makes the white balancing and color correction much harder. This can introduce problems like over-saturation and contrast if it was processed by a daytime-developed ISP. Fig. 3.1 shows some of the aesthetic issues in night scenes. This is why we need to develop a pipeline that is specifically developed to render night scenes. This is what is commonly done in cameras, especially on mobile phone cameras which are also bound with their limited hardware.

## 3.2 Challenge Description

For the challenge, the organizers provided raw images of night scenes captured by Canon 7D camera. The provided images are encoded in 16-bit PNG format and some additional meta-data information for each image is included like color

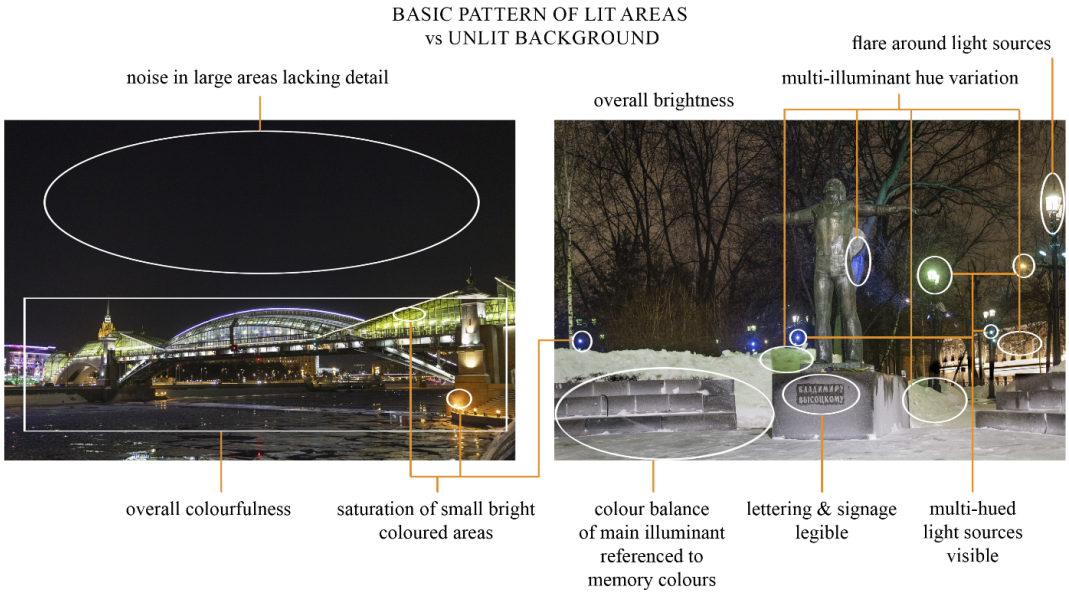


**Figure 3.1:** *Different aesthetic issues in night scenes. Image from Ershov et al. (2022)*

correction matrix, and noise level. The first stage includes 50 raw images used to develop the imaging pipeline. The challenge includes 2 validation stages which can be used by participants to submit their solutions and get an evaluation and feedback before final submission. each of the evaluation stages provides additional 50 raw images that are used as the submission for that stage. For the final submission, another 50 raw images are provided for the participants to process using their final solution and submit in addition to another hidden 50 images used for the final evaluation. The dataset in total is 150 raw images for development and evaluation and 100 images (50 available, 50 hidden) for testing as the final submission. The dataset provided has no GT reference.

For evaluation and results the organizers used Mean Opinion Score (MOS) from a subjective evaluation experiment that was conducted using Toloka (a service similar to Mechanical Turk). The scores were collected by pair comparison by providing two images and the evaluator has to choose which image they prefer. All the images compared are the solutions processing output, no GT existed. In the final stage, the 10 solutions with the highest MOS are evaluated by a professional photographer to determine the final ranking.

**(a)** *Basic Processing Pipeline Output*

**(b)** *Enahnced Image Using Zero-DCE Guo et al. (2020)*

**(c)** *Enhanced Image Using DSLR Lim and Kim (2020)*

**Figure 3.2:** *The results we obtained by using Low Light Image Enhancement Methods*

## 3.3 Methodology

In this part, we will represent our tested approaches and which solutions we submitted for the final testing stage.

### 3.3.1 Using Low Light Image Enhancement Algorithms

Because the challenge doesn't include a GT and only provides raw images without reference we wanted to develop an approach that is not fully supervised and doesn't depend on annotated data. Our first approach was to depend on a basic processing pipeline to reconstruct the RGB image from the raw image and the provided metadata. Then utilize a low-light image enhancement (LLIE) approach to process and enhance the reconstructed RGB image. By utilizing this approach we will not need specific sensor training data as we will use a basic unsupervised general processing pipeline to get RGB images and then we can use general RGB data for the enhancement part.

In this experiment, we utilized the basic processing pipeline that was provided by the challenge organizers to obtain the RGB image. This pipeline consists of black-level normalization, denoising, white balancing, color correction, gamma correction, and tone mapping. For the LLIE algorithms, we utilized 2 different deep learning-based approaches. We choose the deep learning-based approaches because they are the current state of the art in LLIE. The first approach we chose is Zero-DCE Guo et al. (2020) which treats the low light enhancement problem as multi-light-enhancement curves. This method is a Zero reference method which means it needs no GT and depends on the statistics of the output image for optimization. The second method we chose is DSLR Lim and Kim (2020) which is

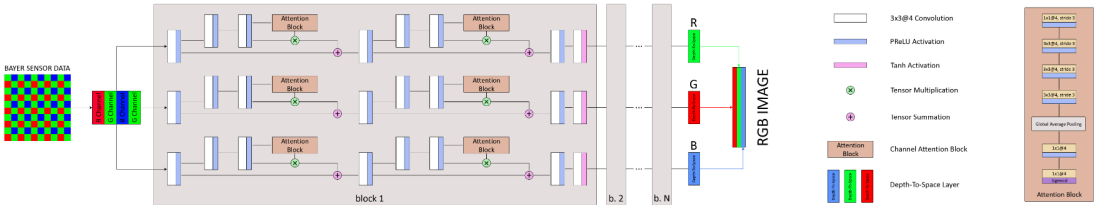


**Figure 3.3:** *Architecture Details of MicroISP Model. Image from Ignatov et al. (2022b)*

a fully supervised method that depends on three stages Laplacian pyramid network for enhancement.

The results we obtained from this method are shown in Fig.3.2. As we see in the results the final images are overexposed noisy and over-saturated. This is because the LLIE algorithms are mainly concerned with relighting and enhancing the under-exposed and dark images. This is because of the construction of the methods and the datasets they were trained on. As we see in the processing pipeline output in Fig. 3.2 The challenge data are not under-exposed as the images were captured with proper exposure. But the task is focused on how to process these well-exposed images and how to overcome the problems of night images like high noise and hard white balancing. Because we are using a basic processing pipeline the produced RGB suffer from high noise and bad white balancing and LLIE algorithms don't deal with these issues. The LLIE only focus on image relighting so they will not be suitable for this problem. So we need a method that was developed for the reconstruction of night images from the raw images.

### 3.3.2 Knowledge Distillation

The hardest part about this challenge is the lack of GT data to use for training or evaluation. the methods proposed for this challenge either use a complex and highly tuned Image Processing Pipeline (ISP) to deal with night images or utilized a professional photo editor to create their own training data from the raw images provided. Please refer to the Night Rendering section in the Literature Review 2.4 for more information about the previous solution. For our approach, we wanted a method that doesn't require a lot of tuning and engineering like the conventional ISP pipeline and we don't have the resources to create our own GT for the data provided. Additionally, because there are no size or time constrain in this challenge the proposed solutions mostly consist of big networks or long chains of different modules which make them not feasible to run on camera or mobile hardware. Because our focus in this project is efficient methods we wanted our approach to work directly on mobile and camera hardware.

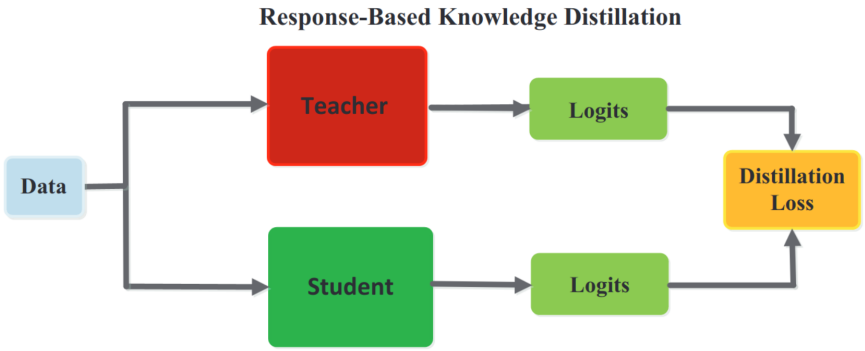


**Figure 3.4:** *Diagram of Response-Based Knowledge Distillation. Image from Gou et al. (2021)*

To overcome the problem of missing training data we utilized a knowledge distillation. knowledge distillation is a process utilized to transfer knowledge from a large model to a smaller one. Large models are more capable and usually achieve better results when you have enough data. But they are sometimes not fully utilized because the dataset or that task doesn't require that big model and the model is usually big to run on limited hardware. This is why knowledge distillation is a very useful tool to create efficient models with a close performance to big models. Usually, the big model is referred to as the teacher model and the small model as the student model. For our teacher model, we used Deep-FlexISP Liu et al. (2022) which is the winning solution of the NTIRE 2022 Challenge on Night Photography Rendering challenge. The authors of Deep-FlexISP provided the pre-trained model which is very crucial for our approach as we will not need to annotate our dataset and we can only depend on the pre-trained model they provided. Deep-FlexISP consists of 3 different models, denoising, white balance, and reconstruction that are applied sequentially to the input raw image. For the student network, we used MicroISP architecture Ignatov et al. (2022b) which is a very memory efficient and fast architecture that can process 32MP images on mobile devices hardware. The Model architecture is shown in Fig. 3.3.

Because the teacher network consists of different models and we don't have the dataset that the model was trained on, only the pre-trained weights, we chose a simple form of knowledge distillation called Response-Based Knowledge Distillation Fig. 3.4. This form focuses on the output of the models as the student network will be trained to mimic the output of the teacher network. For our method, we started by creating the dataset that the student network will train on. We did that by running the raw dataset on the teacher network and then using it to train the student network to mimic the teacher network. With this process, we have a fully supervised dataset represented by the raw images provided by the challenge

and the GT represented by the teacher network output. We used the 150 images provided in the initial stage and the 2 evaluation stages for training and evaluation by using a holdout 80/20 split. Then for training the student network, we split the full images into batches of size 256 × 256. We trained the model for 500 epochs using Adam optimizer Kingma and Ba (2014) with a learning rate 1e-5 using an NVIDIA RTX 3090 Ti (4 hours of training).

## 3.4 Results

We first did a preliminary study to decide on the best loss functions that will allow the student model to better mimic the teacher network. We will first represent the effect of the loss function on the student model accuracy and we will represent the results of the challenge that compare our method to other methods in the challenge.

### 3.4.1 Performance of Student Networks

For the loss function, we investigated 3 different setups. First using only Mean Square Error Loss (MSE Loss), second MSE Loss in addition to VGG loss Johnson et al. (2016) and SSIM loss Wang et al. (2004) (perceptual losses) for better texture and content information. Third MSE, VGG, and SSIM, in addition to Color loss. VGG loss uses VGG Network trained on ImageNet to compare the features extracted from both images. If the features extracted for both images are similar this means both images have the mean content. This loss is usually used to ensure content consistency in image modifications application. SSIM loss calculates the SSIM between the produced image and the GT and it is useful to ensure texture and structure similarity for better perceptual reproduction. For color loss we used the color loss proposed by Ignatov et al. (2017). For the color loss, they apply a Gaussian blur filter on the image to remove texture information and focus more on the color information then convert the RGB image to YCrCb image and use the difference between the color channels CrCb of the images as the color loss.

**Table 3.1:** *Student net performance based on the loss function. Evaluation in comparison with the teacher output (GT)*

| Loss | PSNR | SSIM |
|---|---|---|
| MSE Loss | 27.33 | 0.8628 |
| MSE + VGG + SSIM Loss | 27.39 | 0.8783 |
| MSE + VGG + SSIM + Color Loss | 27.35 | 0.8780 |

**Table 3.2:** *Efficiency Comparison between student network and teacher network (The reconstruction Stage Only)*

| Network | Numb of Parameters | Inference Time (on Nividia 3060) |
|---|---|---|
| Teacher Network (Deep FlexISP) | 23.5 M | 0.33 s |
| Student Network (MicroISP) | 0.017 M | 0.027 s |

**(a)** *Teacher Network output (GT)*

**(b)** *Student Network with MSE Loss*

**(c)** *Student Network with MSE + VGG + SSIM Loss*

**(d)** *Student Network with MSE + VGG + SSIM + Color Loss*

**Figure 3.5:** *The results of the student network using different loss setups in comparison with the teacher network output (GT)*

The output of the different loss setups can be seen in Fig. 3.5 and the quantitative results are shown in Table. 3.1. As we notice from the results and the images there is not much difference between the different loss setups. That is mostly related to the dataset being fully aligned so the pixel-wise loss (MSE) Loss will be enough. But we also notice some improvement with the addition of content and structure losses (SSIM and VGG Loss). These losses are spatial-wise losses that account for the relation between the neighboring pixels and not only pixel-to-pixel comparison and we can notice the improvement in the SSIM results. The color loss didn't add any improvement as it also depends on pixel-wise comparison which is

also included in the MSE Loss. For our final method, we choose the model that was trained with MSE + VGG + SSIM Losses. As we also notice The student network results are close to the teacher network results. The student Network matches the teacher network with 27.39 dB and SSIM of 0.8783 as we see in Table. 3.1. We can also see the efficiency comparison with the teacher network reconstruction stage (this is only 1 stage from 3 stages with other big models) in Table. 3.2. The student network has 1000x fewer parameters than the teacher network and this is crucially important especially when dealing with big images with a limited memory size like the ones in mobile phones. Also, the smaller size result in a much faster speed with 10x faster time, and this difference will be much bigger when computing on mobile hardware with a much less number of cores. But on the other side as we notice from the difference in Fig. 3.5, the student network overall is less bright and the white balancing is less accurate. We trained the student network with image patches so it lacks global information which makes it less accurate in global tasks like white balancing and more prone to global artifacts like vignetting.

### 3.4.2 Comparison between our method and other NTIRE 2023 Challenge methods

Our submitted solution got compared to other methods in the competition through a subjective experiment. For the first evaluation stage, the organizers will compute the Mean Opinion Score (MOS) of the submitted methods. They will calculate MOS by using Toloka (an online service similar to Mechanical Turk) by conducting a pair comparison experiment by choosing 2 images from the output of the challenge methods and asking the participant to choose the best one. It is important to notice that experiment is a non-controlled experiment as the organizers didn't have any control over the viewing setup of the participant. But this experiment represents a wide variety of environments and viewing conditions in real life. So having an uncontrolled environment will be suitable to evaluate what will the users see in their normal viewing environment. After computing the MOS of the submitted methods the top 10 methods will be evaluated and ranked by a professional photographer.

The MOS is shown in Table. 3.3. Our team represented by JMUCVLAB achieved 9th place in the challenge out of around 50 participants. and we achieved 10th place in the professional ranking Table. 3.4. It is important to mention that our method is the only method in the challenge that focuses on efficiency.

We couldn't include the teacher network in this comparison as the MOS is calculated with a subjective comparison between 2023-year solutions. So we can't compare it to last year's 2022 solution with MOS which was calculated compared with other solutions. But the teacher network used was last year's winner and it is the same team as the 3rd-place in this year's competition with an upgraded

**Table 3.3:** *People's choice ranking results. Table From Shutova et al. (2023)*

| Rank | Team | Mean Score |
|---|---|---|
| 1 | IVLTeam | 0.67 |
| 2 | DH_ImageAlgo | 0.645 |
| 3 | MiAlgo | 0.626 |
| 4 | BSSC | 0.606 |
| 5 | DH-AISP | 0.583 |
| 6 | Manual image enhancement | 0.491 |
| 7 | OzUVGL | 0.453 |
| 8 | The Majestic Mavericks | 0.444 |
| 9 | JMUCVLAB | 0.439 |
| 10 | NTU607 | 0.376 |
| 11 | Baseline ISP | 0.345 |

**Table 3.4:** *Professional choice ranking results.Table From Shutova et al. (2023)*

| Rank | Team |
|---|---|
| 1 | MiAlgo |
| 2 | DH_ImageAlgo |
| 3 | IVLTeam |
| 4 | The Majestic Mavericks |
| 5 | BSSC |
| 6 | NTU607 |
| 7 | DH-AISP |
| 8 | Manual image enhancement |
| 9 | OzUVGL |
| 10 | JMUCVLAB |

version of Deep FlexISP (MiAlgo). Even though the student network was trained on image patches with a small amount of data it was comparable to the teacher network which shows the potential of this approach to create more efficient models from the more big and complex models.

## 3.5 Discussion

In this chapter, we proposed our efficient night rendering model that was one of the top solutions in the Night image rendering challenge. Our approach was the only approach that focused on efficiency because of not having time constraints on the proposed solutions. By utilizing Knowledge distillation, our method didn't need

any tuning or multiple modules and didn't require annotating our own dataset. our method has 1000x fewer parameters than the last year's model winner Deep FlexISP Liu et al. (2022) and can run efficiently on mobile hardware.

For the Challenge, it is important to have constraints on the model size and inference time to have usable solutions that can be used with the camera's hardware. Also, a more comprehensive night imaging dataset with a variety of sensors will be available, especially mobile camera sensors. Because of the hardware limitations, rendering night images from mobile phones will be more challenging so a dataset to work on that will be important to improve the night rendering capabilities of mobile cameras. There is no dataset available for learned-based ISP that includes a good variety of lighting conditions, especially night images. So the availability of a dataset with GT for night images will help in developing learned-based ISP Networks that can deal with different environmental and lighting conditions.

For our approach, the biggest drawback pointed out by the professional evaluator was the overall dim image and non-accurate color reproduction. For that, we need to integrate the global information of the images during training in addition to the image patches and not only depend on the image patches. We discussed this issue in a later chapter 5. Even though MicroISP arch is very efficient but because of the architecture constraints to reaching that efficiency it lacks in performance compared to other ISP architectures. It also requires a long time of training and more tuning than other efficient models. It will be important to investigate other efficient ISP architectures that have a better compensation between accuracy and efficiency.

# 4 ISP Data Limitations

In this chapter, we will talk about the problems of the creation of the datasets for training a leaned based ISP For mobile cameras. We will propose our solution to overcome these issues. We will mostly focus on the Alignment issues in the datasets and the limitations of the dataset sizes and variety.

## 4.1 Problem Definition

The direction developed to train end-to-end ISP for mobile phones was developed by creating a network that learns to imitate the output of a DSLR camera. This intuition came from the biggest limitations of mobile phone cameras which is hardware limitations that mobile manufacturers try to overcome by heavy image processing pipelines. So the idea was to create a data set to develop a model that will overcome these hardware limitations by imitating DSLR cameras which have much better hardware and much fewer size limitations which allow for better more accurate captures.

The datasets were created by capturing the same scene with a DSLR camera and a mobile phone so we have a matched scene for both DSLR and Mobile. This dataset is used to train a network that takes the input as the raw image from the mobile camera and outputs something similar to the DSLR image. But because of the different optical systems and sensors in both cameras, you will not have a good match between the two cameras' output to be able to use the dataset in a supervised way. To fix that issue the dataset is aligned using SIFT keypoints detector Lowe (2004) and RANSAC algorithm or using deep learning dense matching algorithms like the one from Truong et al. (2021). Then split the images into paired batches (the image pair consists of a phone raw image patch and a DSLR patch) and the patches with the high correlation are chosen for the final dataset. Because of this process, a lot of patches from the dataset are ignored and not used for not being highly correlated. We can notice that in Zurich RAW to RGB dataset Ignatov et al. (2020b) as they only extracted 48043 RAW-RGB image pairs from 20 thousand full photos collected. Additionally, after this alignment process, we can't achieve a

(a) *Reflection Miss-Alignment* (b) *Moving Objects* (c) *Warping Distortion*

**Figure 4.1:** *Example of alignment problems is ISP training data. Images from the Fujifilm UltraISP dataset Ignatov et al. (2022c) and Zurich RAW to RGB dataset Ignatov et al. (2020b)*

fully aligned dataset and there is no perfect alignment method that can account for this big difference between the optical systems. There will always be a small miss alignment between the images. There are also differences coming from moving objects and differences in reflection because of the difference in capturing angle. Additionally, there will be some distortions from the alignment operation. We can see examples of different distortions in Fig. 4.1. These miss alignment problems affect the training of the network as the network learns the artifacts in the dataset and results in less detailed images. Additionally because of the miss alignment pixel-wise losses like MSE Loss will not be sufficient as the pixel-wise matching in the dataset is not correct. So including contextual and structural losses like SSIM loss and VGG Loss is needed.

To solve this problem Zhang et al. (2021b) proposed a way to refine the alignment during training time for better-supervised training. They first reconstruct a simple version of the input raw image and compute the optical flow between the reconstructed raw and the DSLR Ground truth image. Then they use the optical flow to densely align the DSLR GT patch perfectly with the raw image.

**(a)** *Input Phone Raw Visualized* **(b)** *DSLR Image GT Aligned* **(c)** *Without densely aligned patches* **(d)** *With densely aligned patches*

**Figure 4.2:** *The effect of badly aligned training data on the model output. Images from Zhang et al. (2021b)*

Training with better-aligned data results in better and less blurred images as we notice in the output in Fig. 4.2. The problem with dense matching it sometimes results in warping artifacts like the ones in Fig. 4.1. But because they are trained on small patches and the patches are moderately aligned these artifacts are not that effective. This method works on the batches that were already aligned so the alignment process to create the batches is still required. This result in a lot of patches being eliminated because only the highly correlated image pairs from the initial alignment process are used.

In this section, we will propose our own process to create a fully aligned dataset from the existing weakly aligned dataset. Our method also extends to creating a new synthetic dataset for DSLR images only without phone captures. This will overcome the problem of data limitation and the complexity of capturing and creating this kind of dataset.

## 4.2 Raw to Raw Mapping

Our idea was to use the weakly aligned dataset to create a process that will allow us to create a fully aligned high-quality synthetic dataset that can be used to train the ISP Network.

Our initial idea as shown in the diagram 4.3 is to develop a mapping method to take the raw images of the DSLR Camera and apply the mapping to convert their color space to the phone color space. By this, we will have a fully aligned dataset with raw images that have the same characteristics as the phone's raw images. The common way to apply this color space transfer is using the color conversion matrix that was calculated using some calibration process and color standard objects. This color conversion matrix is usually included in the metadata of the raw images but in the available dataset, the raw images provided for the mobile phone don't

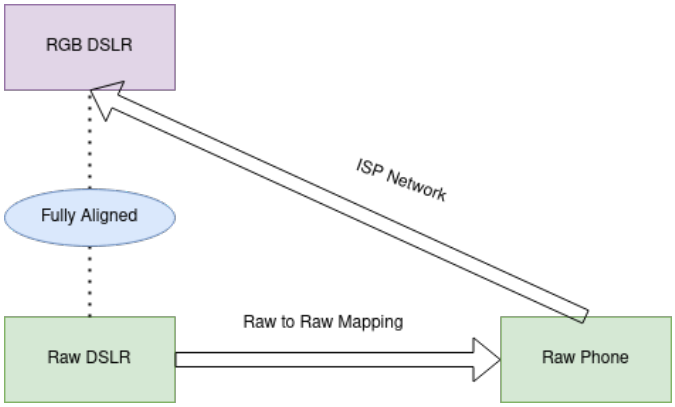


**Figure 4.3:** *The Pipeline of creating a Fully Aligned dataset using Raw to Raw Conversion. We take the raw DSLR image and apply raw-to-raw mapping to convert it to the color space of the phone raw. The generated raw will be fully aligned with the DSLR image*

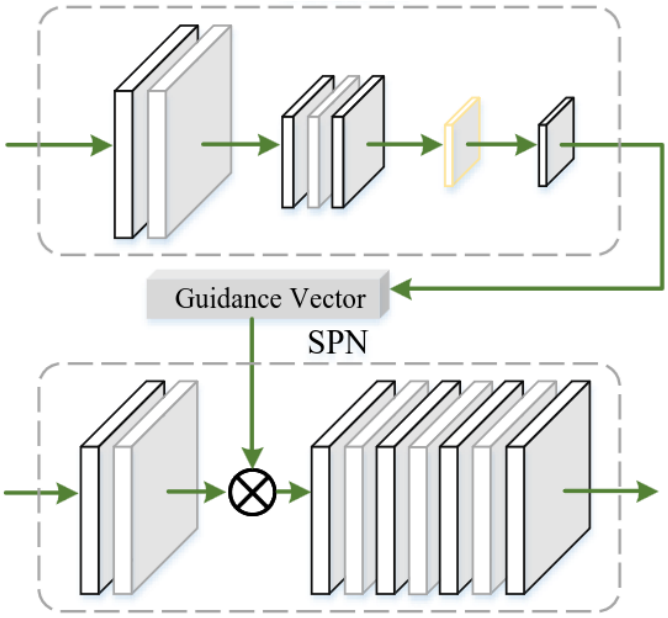


**Figure 4.4:** *Raw to Raw mapping Network. inspired from Zhang et al. (2021b)*

**Figure 4.5:** *The output of the network trained of the synthetic dataset generated by raw to raw mapping on real raw images*

include metadata information. To overcome this issue we developed a pixel-wise network to learn the mapping from the DSLR color space to the Phone Color Space. The network we used is inspired by the pixel-wise network from Zhang et al. (2021b) that they used as part of their architecture. The network consists of pixel-wise modifications that are applied by using 1x1 convolution layers. The global guidance branch is used to compress the full image to a guidance vector to have a full perspective of the input image. The full Architecture is shown in Fig. 4.4. It was important in the network to force the modifications to be only pixel-wise to simulate the traditional color conversion process and to ensure there will be no context and structural modifications to the raw image. This will ensure the output of the network is still fully aligned with the DSLR image.

To train the raw-to-raw mapping network we used the raw images from the ISP dataset. Similar to the creation process of the ISP dataset we took the raw DSLR images and the raw Phone images and aligned them using SIFT keypoints detector Lowe (2004) and RANSAC algorithm. We then split the images into smaller patches and took the patches with high correlation as the training dataset. We then trained the network on the dataset to map the DSLR raw image to the Phone raw image using MSE loss and we achieved a PSNR of 31.5 dB.

To test the quality of the mapping and the generated data we will train an ISP network on the generated fully aligned dataset and then test it on the real dataset. The closer the performance of the ISP model trained on the synthetic dataset to the ISP model trained on the real dataset the higher the quality of the synthetic dataset. If we can create a synthetic dataset of a closer quality to the real dataset we will solve the problem of miss alignment as the synthetic dataset generated is fully aligned with the DSLR Ground Truth.

As we notice from the output of the trained ISP network on the real dataset (Fig. 4.5) the quality of the synthetic dataset is not sufficient. The color characteristics of the synthetic dataset are still different than the real raw phone images. Even though we achieved a good PSNR on the raw-to-raw mapping network the dataset we trained on also suffers from the problem of miss alignment. Because the dataset

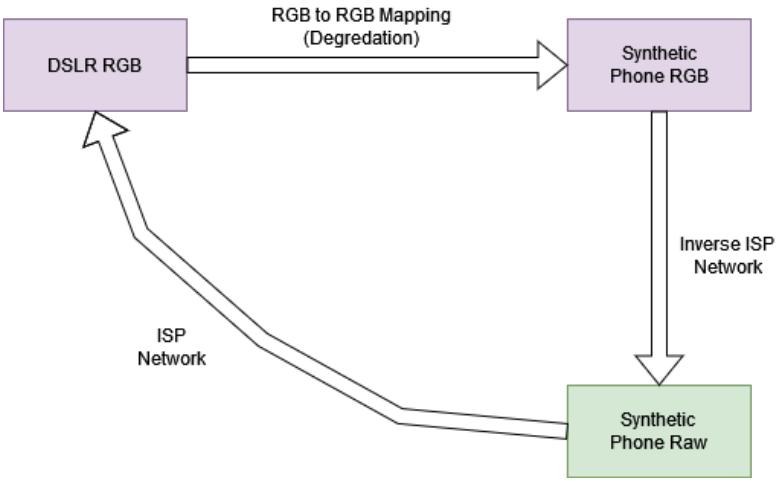


**Figure 4.6:** *The Pipeline of creating a Fully Aligned dataset using RGB to RGB Conversion. We apply RGB to RGB mapping to generate from the DSLR RGB Image an RGB image with the same characteristics as the phone RGB. Then we use an Inverse ISP network to generate a synthetic phone raw image that is fully aligned with the DSLR RGB (GT).*

is also generated in the same way as the ISP dataset and it is from 2 different cameras with 2 different optical systems there will always be some small miss alignment even with the highly correlated patches chosen. This miss alignment is even more sensitive with raw images as the pixels are not in the same band as in the case of our CFA cameras. So with the miss alignment, the network can learn to map different bands to another one which will ruin the pattern information and the arrangement in the raw images. Because the raw images are just light measurements that didn't get processed yet they will be much more sensitive to any modification differently from the RGB images. Additionally, by this method, we only learn the conversion of the color space and ignore the other quality differences between the 2 camera systems like sharpness, dynamic range, and sensor quality. This resulted in a bad synthetic dataset that will not be fit for training.

We then modified this pipeline to apply the mapping on the RGB images instead of the raw images and account for the other difference between the 2 camera systems. We describe this method in the next section.

## 4.3 RGB to RGB Mapping

Because of the problem of the raw to raw mappings and the problems of modifying the raw images which can easily destroy the construction of the raw image we

changed the pipeline to apply the mapping in the RGB images. We will apply the mapping between the DSLR and the phone on the RGB image. As we see in the pipeline diagram in Fig. 4.6 We take the DSLR RGB Image and apply an RGB to RGB mapping (degradation) to generate a synthetic RGB image that represents the RGB image generated from a phone raw image. This phone's RGB image can be the output of the mobile ISP for example. Then we take the synthetic Phone RGB image and apply an Inverse ISP Process that generates a synthetic phone raw image that is fully aligned with DSLR RGB image (GT). If the RGB to RGB Mapping process is accurate we will generate a synthetic RGB Image that is similar to the RGB image that the phone will generate if it captured the same scene. Additionally, if the reverse process is accurate we will generate a synthetic phone raw image that is fully aligned with DSLR RGB image (GT) and have the same characteristics as the phone raw images. By applying the mapping process in the RGB domain we overcome the issues with the Raw to Raw mapping as the RGB image are processed and include structural and context information. So the RGB to RGB Mapping will include also mapping the difference in texture, sharpness, and other image quality images in addition to color. This will result in more accurate synthetic raw and will also avoid the destruction of the pattern and the bands' distribution in the raw image. For this pipeline, we need a representation of the phone RGB output so we can learn the mapping between the DSLR RGB and Phone RGB. For that, we chose the output of the mobile phone's internal ISP. The only dataset that includes both the DSLR RGB Images, Phone ISP Images, and Phone Raw Images is a small set of Zurich RAW to RGB dataset Ignatov et al. (2020b) with 148 full images. we used this dataset as a proof of concept for this pipeline. Next, we will describe each module of the pipeline and the training process of each part.

### 4.3.1 Inverse ISP Network

As we mentioned before we will apply the mapping between the DSLR and the Phone on RGB images. So in order to have the synthetic raw images that we need for training the ISP Network we will need to convert the Phone's RGB image back to raw. This requires reversing the process of the ISP. Most of the mobile ISPs are consistent of complex series of different processing modules and these ISPs are closed sources and not accessible. Additionally, these ISPs contain nonreversable processes which makes reversing the ISP a very hard process. In the ISP to generate the final RGB image the image goes through some lossely process that removes a lot of the information in the original raw image to have a reusable small-size image. In order to reverse this process you need to estimate the lost information during the ISP process. The process of reversing the ISP is still an open and very

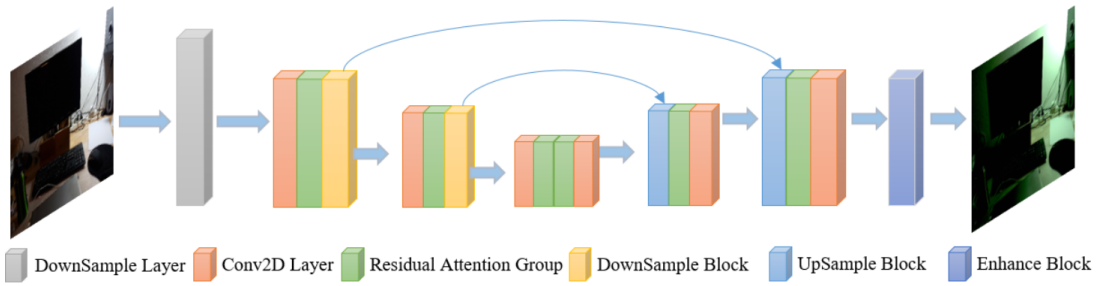


**Figure 4.7:** *Inverser ISP Network Architecture of the winning solution (MiAlgo) of AIM 2022 Challenge Reversed Image Signal Processing and RAW Reconstruction. Image from Conde et al. (2022)*

important problem as the majority of the images that we have is the processed RGB images. We lack enough raw images for the processes that require raw images like denoising. So if we can reverse the ISP processes we can use the available huge amount of RGB images to generate raw images that we can use for a lot of other tasks like denoising, HDR Imaging, white balancing, and more. Because of the power of neural networks as a function estimator, we can use it to estimate the reverse function of the ISP. A new solution direction of the reverse ISP problem is to estimate the reverse function of the ISP using a Neural Network and raw image with their ISP-processed RGB images. In this part, we will talk about the 2 reverse ISP networks we used for our experiments and later we will represent the evaluation process and the performance of each one and which one we used for our experiments.

#### 4.3.1.1 Reverse ISP

For the first reverse model we used the winning solution from AIM 2022 Challenge Reversed Image Signal Processing and RAW Reconstruction MiAlgo Solution Conde et al. (2022). The Network architecture is shown in Fig. 4.7. The network consists of an encoder-decoder in an Unet-like architecture. The network uses residual attention blocks with spacial and channel-wise attention and residual connections for better feature extraction and processing. To train the network we used the small set of Zurich RAW to RGB dataset Ignatov et al. (2020b) that contains both the phone Raw images and Phone RGB images generated by the phone's internal ISP. This small set contains 148 full images with a small miss-alignment between the Raw image and the RGB image because of some processes in the mobile ISP like super reduction and image enhancement. To solve this slight miss-alignment we split the images into small patches of size 256x256. We did that by using two windows sliding similarly in both the RGB and Raw image and the position of the

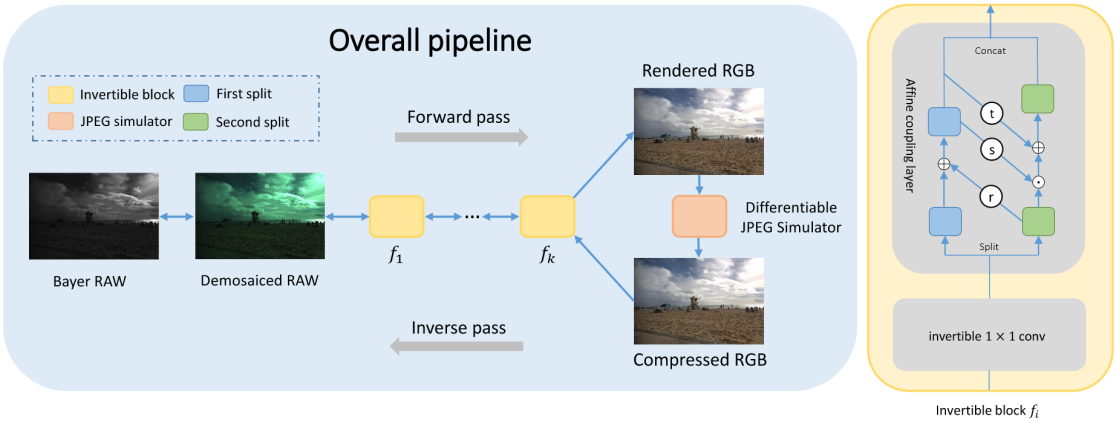


**Figure 4.8:** *The Inevitable ISP Architecture using Normalizing Flow. Image from Xing et al. (2021). The network consists of a series of invertible blocks that are used to process the input image. The architecture of the invertible block is shown on the right side where t is translation, s is resizing, and r is an arbitrary function that is used to enrich the representation learning power of the architecture.*

window in the RGB image is adjusted slightly with a small shift in all directions to minimize the correlation between the patch of the RGB and the Raw. Then we used this dataset for training by using the RGB patch as an input and the Raw patch as output. For the Raw image, we used a simple demosaiced version of the Raw patch as GT computed by linear interpolation to avoid any problems with the pattern structure in the Raw image. We trained the model on the original model setup for 100 epochs and achieved a PSNR of 34.9 dB.

#### 4.3.1.2 Invertible ISP

The second architecture we used for the RGB to Raw conversion is the invertible ISP Xing et al. (2021). The unique thing about this architecture is it is invertible which means that the reverse function of this network is easily computable. They achieved that by replacing the classical neural network layers which are non-invertible by affine coupling layers Dinh et al. (2016) which consists of scale and translation functions. Invertible 1x1 Convolution layers Kingma and Dhariwal (2018) were also utilized to control the number of features in the feature layer in each stage of the network. The authors also developed a differentiable JPEG simulator that simulates the lossely JPEG compression process that happens in the ISP. For the training process, we used the same dataset from the other Reverse ISP architecture we investigated. In the training process, we optimize for both directions in the invertible architecture. The forward pass uses the input as the simply demosaiced

version of the Raw image and output GT as the RGB Image processed by the mobile Internal ISP. For the backward pass, the input is the RGB Image and the output GT is the Raw demosaiced image. The architecture optimized the network parameters for both tasks using L1 loss as described in the next equation.

$$L = ||f(x) - y||_1 + \lambda||f^{-1}(y) - x||_1$$

Where $L$ is the loss function $f$ is the ISP process $x$ is the demosaiced Raw patch and $y$ is the RGB patch. We trained the model on the original model setup for 50 epochs and achieved a PSNR of 30.53 dB for the backward pass (RGB to Raw) and 22.23 dB for the forward pass (Raw to RGB).

It is important to notice that the network is invertible. So during the optimization process in both forward and backward passes, the same parameters are being optimized and used for both tasks. So if you processed the output of the network in the other direction you will get the exact same input you gave for the network. By utilizing the invertibility feature of this architecture we can train the RGB to RGB mapping on the output of the forward pass of the invertible network (Synthetic RGB image simulating the mobile RGB output). This will ensure the backward pass to get the raw image without error. By training the RGB to RGB mapping on the output of the invertible ISP we eliminate the error of the reverse part in the pipeline as the architecture is invertible. This insure a fully accurate backward pass and the only error will be the RGB to RGB mapping between the DSLR image and the Mobile RGB image generated by the Invertible ISP Network.

### 4.3.2 RGB to RGB Mapping Network

For the RGB to RGB Mapping, we will be using the architecture from Ignatov et al. (2018) which is the Network that they used for Image Enhancement. Architecture shown in Fig. 4.9 The Network consists of a series of residual blocks that are used to modify the input image to achieve the desired output. For this task, we use the input image as the DSLR image and the desired output is the mobile image. Because the DSLR image is considered the better quality one we consider this network and a degradation network to decrease and adjust the quality of the DSLR image to produce a mobile-like quality and colors. In the training of RGB to RGB mapping we will deal with the data from 2 different imaging systems. That will have the same alignment problem as the problem with the ISP data we discussed before. We need to avoid the network learning the misalignment between the 2 imaging systems which will result in carrying this miss alignment to the synthetic raw images we will generate. This will result in the generated synthetic raw images not fully aligned with the DSLR GT images which is what we are trying to achieve. To avoid this issue we didn't use any pixel-wise losses during the training process.

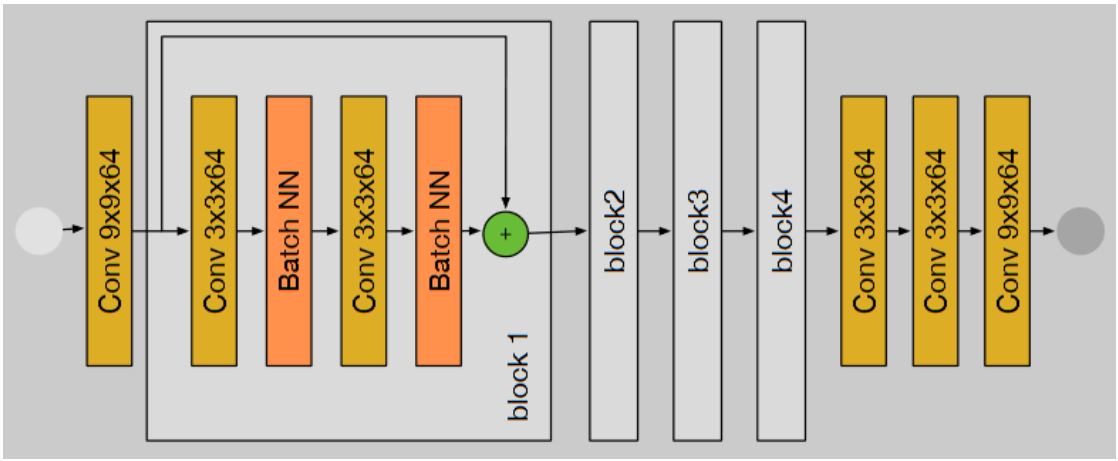


**Figure 4.9:** *The RGB to RGB Mapping Architecture. Image from Ignatov et al. (2018)*

We used SSIM loss to focus on the texture, structural and perceptual components of the images, and VGG loss to ensure content consistency between the input and the output. We also used the color loss we describe in the Night Image Rendering chapter 3.4.1 by blurring the output and the GT images to remove the textural information and decrease the pixel dependency and convert the images to YCrCb. Then we use the difference between the color channels CrCb of the images as the color loss. The loss function we used are fully supervised loss functions but they don't depend on the pixel-wise comparison which is beneficial for the miss alignment issue.

For the training process of the RGB to RGB, data is different for each one of the reverse ISP networks we used. For the Reverse ISP Network from Reversed Image Signal Processing and RAW Reconstruction Challenge (MiAlgo) solutionConde et al. (2022), we trained the RGB to RGB mapping network to map from the DSLR Image to the mobile image that was produced using the mobile internal ISP. To create this dataset we used the 148 full images from Zurich RAW to RGB dataset Ignatov et al. (2020b) and aligned the DSLR image and Mobile ISP image using the same process as the ISP dataset by using SIFT keypoint detector Lowe (2004) and RANSAC algorithm. Then we split the images into smaller patches. Because the RGB to RGB network we created doesn't use pixel-wise loss we decreased the threshold of the cross-correlation between the patch pairs. We choose a threshold of .5 to increase the dataset size and neglect fewer patches. Then we used this dataset to train the RGB to RGB Mapping. We refer to this pipeline as **Pipeline 1**

For the Invertible ISP network Xing et al. (2021), we used a different process. We trained the RGB to RGB mapping network to map from the DSLR image to

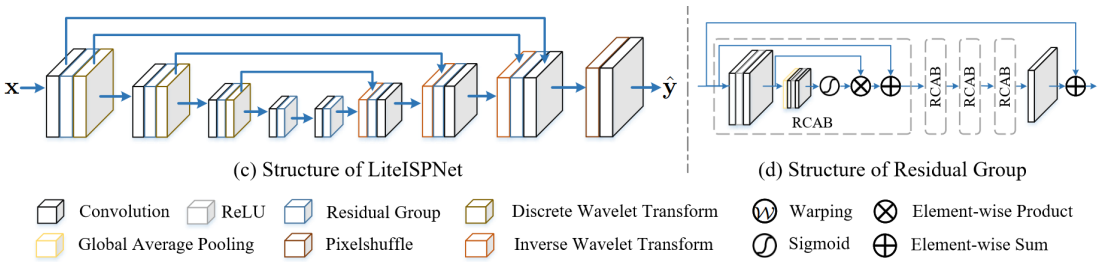


**Figure 4.10:** *Lite ISP Architecture. The Network consists of Residual groups that consist of Residual channel attention groups (architecture shown on the right side). Discrete Wavelet Transform is used for feature decomposition to increase the receptive field of the network. Image from Zhang et al. (2021b)*

the output of the forward pass in the Invertible ISP network. By using this process we decreased the error accumulation between the RGB to RGB mapping and the Reverse ISP. If we got a similar output to the output of the Invertible ISP we ensure a fully accurate reverse step because of the invertibility of the network. So to create the dataset for this training we use the patches of the DSLR image and the Phone Raw images from Zurich RAW to RGB dataset Ignatov et al. (2020b). We input the phone raw patches to the Invertible ISP network forward pass to get the output of the Invertible ISP. Then we train the RGB to RGB Mapping network on the DSLR patches and the output of the Invertible ISP forward pass. We refer to this pipeline as **Pipeline 2**.

We trained both versions of the RGB to RGB mapping with the same hyper-parameters (train for 100 epochs with a learning rate of 1e-4) and the same loss functions. The difference was the dataset used. We tested the 2 processes and we chose the best one for further experiments. For the first pipeline, the RGB to RGB mapping network achieved 21.19 dB, and for the second pipeline 22.29 dB.

### 4.3.3 ISP Network

For the ISP network we used for our experiment, we used LiteISP Network from Zhang et al. (2021b). The network architecture is shown in Fig. 4.10. LiteISP is a lite weight-efficient network that consists of Unet-Like Ronneberger et al. (2015) multi-level wavelet architecture. The network consists of residual group blocks which consist of residual channel attention blocks for accurate feature extraction. The network also uses Discrete Wavelet Transform for feature decomposition Liu et al. (2019) for down sampling and up sampling to increase the receptive field in the network. We used this architecture to train an ISP Network with the synthetic data we created using our pipeline and compare it to the same model trained on a real dataset.

# 4.4 Results

## 4.4.1 Synthetic Data quality

First, we tested the quality of the synthetic datasets generated by the two RGB 2 RGB mapping pipelines. We tested the quality by training an ISP Network on the synthetic dataset and evaluating it on the real dataset. To create the synthetic dataset we used for training we took the 148 DSLR full images from Zurich RAW to RGB dataset Ignatov et al. (2020b) and applied the RGB to RGB mapping to obtain synthetic mobile phone images. Then we applied the reverse ISP network on these images to get the synthetic raw images (same process for both RGB to RGB mapping pipelines). Because the DSLR images (GT) are fully aligned with the generated synthetic raw images we split the images directly into patches of size 448x448 without any additional steps and without eliminating any pairs like the process in the read ISP dataset.

For the real dataset, we used the same dataset of the 148 DSLR and Phone raw full images from Zurich RAW to RGB dataset Ignatov et al. (2020b). We processed the dataset by following the same process as the other ISP dataset by aligning the images using SIFT keypoint detector Lowe (2004) and RANSAC algorithm. Then spliting the imaging into patches and choosing the patches with a certain correlation to the final dataset. For this dataset, we choose a cross-correlation of .5 for the selected pairs because of the small size of the dataset to eliminate fewer patches and also to show the effect of high miss-alignment on the output of the ISP Network. We split this dataset into training and testing (90% and 10% respectively) and we used the test part to test the networks trained on the synthetic datasets. We called this dataset **Zurich RAW to RGB Small**.

From the difference in the dataset size between the real dataset and the synthetic dataset we can notice the problem of miss-alignment of reducing the amount of usable data. We need to eliminate a lot of the patches in the process of creating the ISP dataset because of the big miss-alignment. For example, the real ISP dataset size we created from the 148 images contains around 6K patches only but for the synthetic dataset because it's fully aligned we don't remove any patches. The same 148 images produce 25K fully aligned image patches. So if our pipeline created a synthetic dataset that is close to the real dataset we will have a much bigger dataset that will improve the model performance and decrease the requirement for a bigger dataset.

The results are shown in Table. 4.1. As we see from the table the networks that were trained on the synthetic datasets are comparable in the quantitative results with the network that was trained on the real dataset. As expected the network trained on the real dataset achieves the best quantitative results on the real dataset


**(a)** *DSLR Image (GT)* **(b)** *ISP on synthetic Pipeline 1 4.3.1.1* **(c)** *ISP on synthetic Pipeline 2 4.3.1.2* **(d)** *ISP on the Real dataset*

**Figure 4.11:** *The output of the network trained on the synthetic RGB to RGB mapping dataset on real raw images in comparison with training on real dataset.*

**Table 4.1:** *The ISP Network performance trained on synthetic data and evaluated on a real dataset. The first network is trained on the real dataset. The second network is trained on the synthetic dataset created by the Reverse ISP Network from MiAlgo 4.3.1.1. The third network trained on the synthetic dataset created by the Invertible ISP 4.3.1.2.*

| Train Data | PSNR | SSIM |
|---|---|---|
| Zurich RAW to RGB Small | 18.30 | 0.09 |
| DSLR to Mobile ISP Mapping + MiAlgo Reverse ISP (Pipeline 1) 4.3.1.1 | 15.46 | 0.06 |
| DSLR to synthetic Mobile ISP Mapping + Invertible ISP (Pipeline 2) 4.3.1.2 | 15.76 | 0.08 |

because the network is optimized in this dataset and tested on the same dataset. For the difference between the 2 pipelines used for the synthetic datasets for each pipeline. The pipeline that uses the invertible ISP achieved better results. This is because we eliminated the error of the reverse ISP step by training to RGB to RGB mapping to map to the output of the Invertable ISP network and we use the invertibility feature of the network to get the synthetic raw.

To also better understand the performance of the compared models we also investigated the output of each network as shown in Fig. 4.11. Differently from the quantitative results, we noticed the output of the network trained on the synthetic datasets is better than the network trained on the real dataset. We can notice the effect of high miss alignment in the dataset in the output of the network trained on real data. The output images are very blurred and washed out with worse details and bad color reproduction. But this model achieved better quantitative results because the test data have also the same misalignment problem as the training data. The washed-out blurred images will get better quantitative results in this case because of the pixel-wise comparison nature of the PSNR metric (We also noticed the same when we tested on a highly aligned dataset). On the contrary, the images from the networks trained on the synthetic dataset are not blurred or washed out especially the Invertible ISP pipeline because the synthetic data is fully aligned. For the network that was trained on the synthetic dataset of the first pipeline (MiAlgo pipeline), the images are darker and the color reproduction is less accurate. We can also notice some color artifacts because of the error in the reverse ISP network. For the network that was trained on the synthetic dataset of the second pipeline (Invertible ISP pipeline), the images are the best looking and they have good color reproduction and good details without any blurring. This shows the effect of eliminating the error of the reverse process we used in this pipeline. But this network struggles with fine details like accurate texture rendering because of the error in the RGB to RGB mapping process. This can be a problem of the small dataset used (only 148 full images) to train the RGB to RGB mapping and the simplicity of the network used for this task. Additional data and a better

network for RGB to RGB will increase the quality of the synthetic dataset. We also train the RGB to RGB mapping on image patches which will make the model miss the global information and decrease the mapping accuracy similar to the problem we discussed in the night image rendering. A way to combine training for local and global information will be required for all networks used so far.

### 4.4.2 Synthetic Data Generation

An important feature of our pipeline is it can be used to generate more data from only DSLR images. So in order to generate more data we don't need to capture both DSLR and Phone Raw and make sure they have similar aspects which is a very hard an time-consuming process. We just need to capture DSLR images with the same camera and process it with our pipeline and the pipeline will generate fully aligned Phone raw images. This will make the process much easier and will generate a much bigger dataset. Also, the DSLR RGB images are available online and easily accessible which will allow to generate much bigger dataset.

In this part, we will show that we can generate a synthetic dataset with data different from the one used to develop the pipeline with similar quality. This will show the ability of the pipeline to generalize. For this experiment, we used the 192 full DSLR images from the ISPIW dataset Shekhar Tripathi et al. (2022) which uses the same DSLR camera for the dataset.

**Table 4.2:** *The ISP Network performance trained on ISPIW synthetic data. The first network is trained on a real dataset. The second network is trained on the synthetic dataset only. The third network is trained on the synthetic dataset and the real dataset combined. The fourth network pre-trained on the synthetic dataset and fine-tuned on the real dataset*

| Train Data | PSNR | SSIM |
|---|---|---|
| Zurich RAW to RGB Small | 18.30 | 0.09 |
| Synthetic ISPIW dataset | 15.95 | 0.09 |
| Synthetic ISPIW dataset + Zurich RAW to RGB Small | 18.54 | 0.09 |
| pre-trained on synthetic ISPIW dataset and finetuned on Zurich RAW to RGB Small | 18.47 | 0.09 |

The results of training ISP Network on the ISPIW synthetic dataset are shown in Tab. 4.2. We first trained a network on ISPIW synthetic dataset and compared it to the other synthetic dataset generated from the data we used to develop the pipeline to see if the pipeline is able to generalize or not. We then tested 2 different ways to use the synthetic data as extra data. The first approach is to use the synthetic dataset combined with the real dataset. the second approach is to use the synthetic dataset as pre-training for the ISP model.

**(a)** *ISP on the Real dataset* **(b)** *ISP on ISPIW synthetic dataset* **(c)** *ISP on ISPIW synthetic and Real dataset* **(d)** *ISP pre-trained with ISPIW synthetic*

**Figure 4.12:** *The output of the network trained on the synthetic ISPIW dataset with different training processes*

As we see from the results the model trained on ISPIW synthetic dataset is even performing better than the one the synthetic dataset from Zurich RAW to RGB Small dataset. This can be because the ISPIW dataset is bigger (192 full images vs 148 full images). This shows that our pipeline was able to generalize for the DSLR dataset that wasn't included in the development process of the pipeline and produce synthetic data with similar quality. We can use this pipeline to generate more data for free.

When we add the synthetic dataset as an additional dataset the bother techniques slightly improved the quantitative performance. As we discussed before the quantitative results are not accurately reflecting the visual results so we also compared the visual results as shown in Fig. 4.12. As we see the network trained on the synthetic ISPIW dataset only performs the best and has a similar quality to the network trained on the synthetic Zurich RAW to RGB Small dataset. The addition of fully aligned data with the miss-aligned data increased the blurriness of the output. This addition confused the model more as the model is now optimizing for both the miss-aligned and the fully aligned data. The addition of a synthetic dataset which is easy and cheap to create with our pipeline can boost the performance of the model. But the quality of the real dataset is still the most important and will affect the model performance no matter how small it is.

## 4.5 Discussion

As we saw from the quantitative and the visual results the proposed pipeline with simple networks and a small amount of data can still generate decent-quality synthetic data. Training on this synthetic data can even surpass the performance of the model trained on the real dataset when the dataset suffers from a noticeable misalignment problem. These results show the effectiveness of applying the mapping on the RGB images instead of the Raw images as they are less sensitive to modifications. Also, the processed RGB images contain context and structural information that can allow the mapping of the difference in the image quality and not only the colors. But the RGB to RGB mapping network is still not very accurate and hinders the quality of the synthetic data. The synthetic data is still not that similar to the real data and we can notice that in the problems that appear when we process on a real dataset, especially with texture and fine details. An improvement on the RGB to RGB mapping is still required.

Additionally, the contradiction between the quantitative and visual results can be noticed. Because the test dataset suffers from miss-alignment and the model trained on a real dataset outputs blurred images, the quantitative results are better for the blurred images because of the nature of the metric used (PSNR). The models trained on the fully aligned synthetic dataset didn't optimize for the miss

alignment. That produces better images but worse numbers on the evaluation metric. This shows the need for better testing datasets with high alignment which needs more full-sized images that are not available in the existent datasets. Also, better testing metrics will be required as even with high aligned dataset the PSNR still prefers blurred images. No matter what the quality of the alignment is there will still be a degree of miss-alignment and the PSNR is very sensitive to that as the metric depends on the pixel-wise comparison. Also, the collection of a bigger dataset for the pipeline development will be needed especially to train the RGB to RGB mapping because of the lack of datasets suitable for our pipeline.

Additionally, we showed that our pipeline can be used to generate new data and is not limited only to the dataset used to develop the pipeline. An additional extension to this pipeline would be to make the RGB to RGB mapping fully unsupervised by utilizing unsupervised losses like GAN loss Ignatov et al. (2018). This addition will require a much bigger dataset and a more complicated process and model as it will require the use of the full image to be able to map between 2 different camera systems fully unsupervised. The available datasets are not enough so the creation of a new dataset to achieve that will be required. But if we managed to achieve this addition we can use any high-quality RGB images to create synthetic data which will open an unlimited amount of synthetic data.

This experiment was conducted as an initial investigation of a novel approach to solve the problem of miss-alignment of the ISP dataset. In addition to a way to generate a synthetic dataset that can be used as a cheap and easy alternative to the hard and time-consuming process of ISP data collection. Our initial experiments showed very promising results for this approach and also showed the importance of additional experiments to better utilize this approach.

# 5 Color Reproduction

In this chapter, we will discuss the problem of color reproduction and the lack of global information in ISP Networks. We will propose a novel idea to combine the use of local and global information to improve the performance of ISP Networks.

## 5.1 Problem Definition

One of the biggest drawbacks of the current ISP Networks is the problem of color reproduction and other global-related artifacts like lens shading. Part of this problem is utilizing only one network to do the full ISP pipeline which hinders the ISP performance. Even though there is some similarity between different tasks of the ISP but there are also different tasks that require different processing. White balancing and denoising are global tasks that require global information and big receptive filed but other tasks like texture processing require more local information. But the biggest problem is the training on image patches which doesn't allow the model to learn from the image's global information and limits the network performance on tasks that depend on global information. We can notice that with White Balancing as the illumination information required for a specific batch can be in another image patch from the same image far away from this patch (ex: illumination information is included in the sky part of the image but required to process other parts of the image like the ground).

The ISP Networks are limited to patch training because of the nature of the dataset used for this task. Because the ISP dataset depends on 2 different camera systems with a wide difference in components even with fine calibration the captured image pairs are not aligned. So to create the dataset the 2 full images have to be aligned and split into patches and only the patch pairs that are highly correlated are chosen for the final dataset. So the full images even after the alignment are not aligned enough to train on them directly. Additionally, training on the full-size image is not feasible because of the Raw image size (3968 x 2976 pixels in the case of Zurich Raw to RGB Dataset Ignatov et al. (2020b)). It is not possible to train on the full image with this size because of all the meta information and

gradient information that is required during the training of the Neural Network. This information is proportional to the input size so training on a full image with this size will require a huge computation power with a very small network.

In the other tasks that require global information, the input image is either resized or the model is trained on a random crop of the input. At our task resizing the input image will destroy the details in the input raw image and will decrease the quality and the actual sensor size which will result in a low-quality final image. Because our task is a raw image reconstruction we need to work on the image with full size without any resizing or modification to the input raw image. The other solution involves training on random crops which also doesn't work on the full global information of the input image. Another common design to process full images is to use an encode-decoder network design like UNet-like design Ronneberger et al. (2015) with a high down-sampling rate. This decreases the size of feature maps in each stage and also increases the model receptive fields which can be beneficial for global tasks. The problem with this design it has a problem with fine details when the down-sampling rate is high. The down-sampling reduces the fine details in the decoder and the model tries to restore them in the encoder with the help of step connection branches between the encoder and decoder. But the performance still suffers when the down-sampling is high which will be the case when we want to process this huge size of Raw images in full resolution.

To overcome this issue we developed a novel method that trains with both the full image and image patches with a separation for global information tasks and local information tasks. We use the resized version of the full image to extract global information and use it to reconstruct the input image patch. This allows us to work with the full details from the images patch and still include global information from the resized full image. The full details and information about the network will be described in the next section.

## 5.2 Methodology

For our architecture, we started with the ISP architecture from Zhang et al. (2021b) paper as our baseline. Their architecture consists of an ISP Network called LiteISP trained with an alignment loss to solve the miss-alignment problem in the isp dataset (discussed in detail in chapter 4). We used this as the initial network and added our modification to improve the model color production, global modifications, and Raw reconstruction. First, we will explain the architecture from Zhang et al. (2021b) and then explain our modification and represent the results of these modifications.

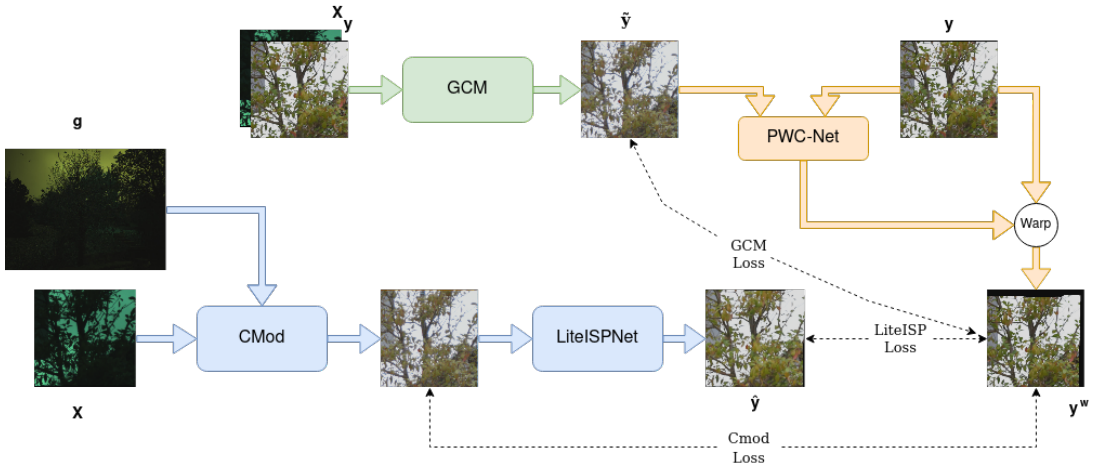


**Figure 5.1:** *The full pipeline for our model modified from the work of Zhang et al. (2021b). We added the CMod for a better color reconstruction by utilizing the information from the full raw image. The pipeline represented is the training pipeline and only the blue part will be used for inference. The pipeline uses the GCM to create a colored version that is used to calculate the optical flow (using PWC-Net) with the GT to align the GT with the input raw. The ISP Model takes the demosaiced input raw patch, and the resized full raw image. CMod applies pixel-wise color reconstruction and liteISP takes this and processes it to produce the final output.*

### 5.2.1 Baseline

Zhang et al. (2021b) pipeline's main contribution is to create an alignment loss. Alignment loss densely aligns the DSLR GT with a color-reconstructed version of the input by the computed optical flow between them. This allows them to have a better aligned GT that will reduce the effect of miss-alignment which results in less blurred output as we see in Fig. 4.2. Their pipeline consists of 3 components, all the components in addition to the new parts can be seen if Fig. 5.1.

The first component is the Global Color Module GCM (Fig. 5.2). The module is concerned with reconstructing a colored version of the input raw with the same color characteristics as the DSLR GT and without any spatial position shifts of the pixels of the input raw. GCM consists of 2 parts a spatially preserving network (SPN) which is a stack of 1 x 1 convolutional layers to process the input raw and produce the DSLR color-like output ($\widetilde{y}$). Additionally, a guide network takes the DSLR GT and the input raw image to better encode the target color characteristics (DSLR Colors). This guide network consists of big receptive field layers (7 x 7 convolutions) and average pooling to generate a guidance vector that gets multiplied

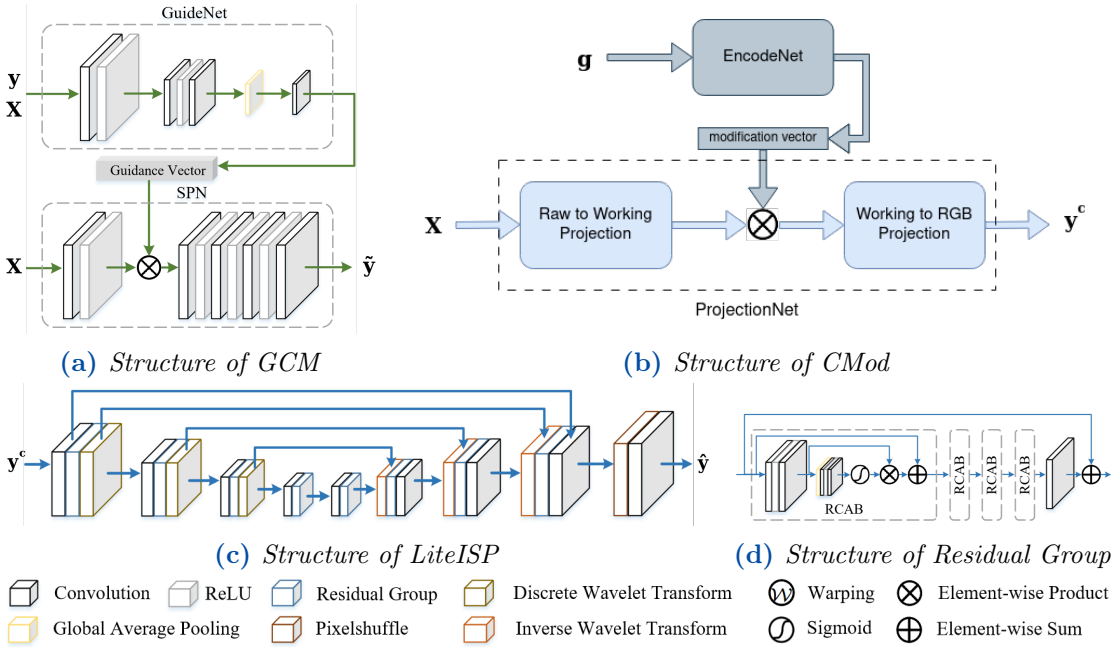


**Figure 5.2:** *The figure shows the structure of the different components in our pipeline. Some figures modified from Zhang et al. (2021b). (a) the structure of GCM which consist of a guidance network and SPN and the demosaiced raw patch (X) and the GT (y) as input. (b) the structure of CMod Which consist of EncodeNet similar to GuideNet in GCM and ProjectionNet Net which consists of 1 x 1 convolutions, the guidance image (g), and the demosaiced raw patch (X) as input. (c) the structure of the liteISP network which consists of Residual groups and DWT blocks. (d) the structure of the residual group which consists of Residual Channel Attention blocks.*

by the feature map in the middle of the SPN. It is important to mention that the GCM module is only used in the training stage and not the inference stage. It is only used to generate a colored version of the input raw that will be used to compute the optical flow to align the DSLR GT. This module is not part of the ISP Network which asserts no data leakage by using the DSLR GT as input to the GCM. Additionally, for the GCM module, they use a simply demosaiced version of the input raw as the network is pixel-wise to have the RGB information in each pixel. This part is described in the next equation.

$$\widetilde{y} = C(\hat{x}, y; \theta_c) \tag{5.1}$$

Where $\widetilde{y}$ is the reconstructed version of the input raw with the DSLR GT colors. $\hat{x}$ is the demosaiced raw image, $y$ is the DSLR GT, and $\theta_c$ is the weights of the GCM.

The second part is the Optical Flow Network. They used the PWC-Net pre-trained model of Sun et al. (2018) to calculate the optical flow between the

reconstructed color version ($\widetilde{y}$) and the DSLR GT ($y$). Then the optical flow is used to warp the DSLR GT ($y$) to get a new GT ($y^w$) that is much better aligned with the input raw image. This new GT will be used as the supervision to train all the modules in the pipeline. It is important to notice, PWC-Net used is the pre-trained network, and its weights are fixed and not optimized during training. The process can be described in the next equations.

$$\Psi = F(\widetilde{y}, y) \tag{5.2}$$

$$y^w = \omega(y, \Psi) \tag{5.3}$$

Where $\Psi$ is the Optical flow, $y$ is the DSLR GT, $\widetilde{y}$ is the output of the GCM, and $y^w$ is the warped DSLR GT.

The last part is the ISP Network. The network used is called LiteISP which is a simplified version of MW-ISPNet Ignatov et al. (2020a) to make it more efficient. We described the network in the previous chapter 4.3.3. The output of the LiteISP will be the final constructed raw image ($\hat{y}$) that will be considered as the final output.

For the loss functions that are used during training all the loss functions are calculated using the warped DSLR GT to avoid the miss-alignment between the input and GT. This helps to avoid the learned shift so the output will be more sharper and accurate. The used loss functions are shown in the following equations.

$$m_i = \begin{cases} 1, [\omega(1, \Psi)]_i \geq 1 - \epsilon \\ 0, otherwise \end{cases} \tag{5.4}$$

$$L_{GCM} = ||m \circ (\widetilde{y} - y^w)||_1 \tag{5.5}$$

$L_{GCM}$ is the loss computed for GCM. $m$ is the mask indicating the valid positions of the computed optical flow to eliminate the bad areas (like the black regions) from the loss computations. $\epsilon$ is a threshold for the optical flow mask computations and set to .001. $\widetilde{y}$ is the output of the GCM, and $y^w$ is the warped DSLR GT.

$$L_{ISP} = \lambda_{l_1}||m \circ (\hat{y} - y^w)||_1 + \lambda_{VGG}||m \circ (\phi(\hat{y}) - \phi(y^w))||_1 \tag{5.6}$$

$L_{ISP}$ is the loss used for the ISP Network training. The first term is the $l_1$ loss computed between the ISP Network output $\hat{y}$ and the warped DSLR GT $y^w$. The second term is the VGG loss we described in the Night Image Rendering chapter3.4.1 and computed between the same terms as well to compare content similarity.

## 5.2.2 Modifications

The alignment loss used in the baseline was helpful to eliminate the effect of miss-alignment in the dataset and produce sharper and less blurry images. The

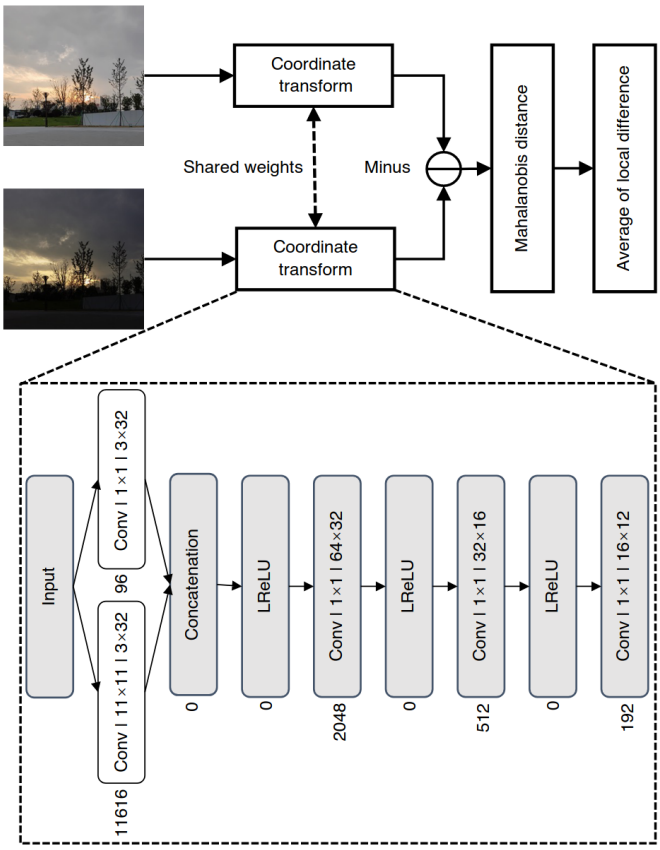


**Figure 5.3:** *Architecture details of CDNet. Image from Wang et al. (2023).*

ISP Network developed with alignment loss achieved Stat-of-The-Art results so we used it as our baseline. Our focus is to improve this model through better integration of global information to improve the model output and achieve better color reproduction. We introduced 3 different components for that purpose and we described each one separately in the next part.

### 5.2.2.1 Color Loss

For better color reproduction there should be a component that focuses on the accuracy of the produced colors in comparison to the target colors. This can be in the form of color loss that measure the color difference (CD) between the output and the target. Even though the color difference is part of the pixel-wise comparison losses like $l_1$ loss but they are very dependent on the pixel comparison so any small miss-alignment will result in a wrong color difference. Also, the difference is not very uniform with respect to the actual perceived color difference.

Some other specific color losses were developed to tackle this problem but they are either not truly representative of the perceived color difference or they are very timely consuming (like histogram color loss Afifi et al. (2021b)). Additionally, some of the other color difference metrics used for color difference applications involve non-differential operations that don't allow them to be used as loss functions.

Following a recent study on the color difference in natural images Wang et al. (2023) most of the color metrics developed were developed based on uniform color patches. There was a lack of information about the generalization of these metrics for natural images with their complex structures and content. Additionally, there is a lack of high-scale natural color difference datasets that can be used for evaluating the different color difference metrics. The authors of Wang et al. (2023) created a perceptual color difference natural images dataset named SPCD Dataset and it is by far the largest available dataset for this task. the dataset consists of 15,335 color images with 1,000 distinct natural scenes collected using 6 different flagship smartphones. They created the perceptual color difference scores of the dataset by conducting a large-scale psychophysical experiment in a carefully controlled laboratory environment. The experiment was done by displaying 2 images with the same content and different color appearance whose color differences are decided by scale-and-slider applet with respect to 5 different grayscale pairs. The grayscale pair patches have known perceptual color difference scores in CIELAB ($\Delta E^*_{ab}$). These scores were used to generate the perceptual color difference scores ($\Delta E^*_{ab}$) of the image pairs in the dataset. The final dataset consists of 30,000 image pairs (with some non-perfectly aligned image pairs). Ten male and ten female observers participated in this experiment and were asked to give a CD score for all image pairs and the experiment lasted for 4 months.

The authors used the dataset to develop their own learning-based CD metric (CD-Net) and compared it to 33 of the commonly used CD metrics. The method they developed consists of 2 parts. The first one is a coordinate transform which takes the image pair and converts each image to a new vector space that will be used to measure the difference between the pair. Secondly is the CD calculation which involves the comparison between the pair in the new vector space to give a final single CD score. The full architecture is shown in Fig. 5.3. The Coordinate Transform is a generalization of the color space transformation in a learnable way by using Neural Networks. The Network consists of a filter bank created by different CNNs with different kernel sizes to integrate spatial information into the transformation. Followed by different 1 x 1 CNNs to learn fine-scale CD maps with different sizes and also maintain the spatial size of the image without shifting to allow for pixel comparison for CD calculation. For the CD Calculation, we take the output of the Coordinate Transform Network for each image of the pair and we apply a pixel-wise comparison using Mahalanobis distance between them using

the following equation.

$$\Delta E(x_{ij}, y_{ij}) = \sqrt{(f(x)_{ij} - f(y)_{ij})^T S^{-1} (f(x)_{ij} - f(y)_{ij})} \tag{5.7}$$

Where $S$ is a learnable matrix that is part of the optimization process. Then the average of $\Delta E(x_{ij}, y_{ij})$ for all pixels is used as the Network CD Score. The full CD-Net is then optimized using MSE loss between the Network CD Score and the GT CD Score ($\Delta E^*_{ab}$)

The developed method achieved better results than all the tested 33 CD metrics both on aligned and non-aligned image pairs. One of the big advantages of the CD-Net for us is being totally differentiable which allows us to directly integrate it as color loss in our ISP Network without worrying about any non-differentiable steps in the calculations like other metrics. The integration of this as a color loss will allow the model to optimize for color difference separately and should produce perceptually closer colors to the desired output.

#### 5.2.2.2 Color Module

We created a Color module (CMod) that is added before the ISP network for the purpose of color reconstruction. We separated the color reconstruction task as it is mainly a global modification task that depends on the global information in the image and training only on a patch image will limit this global information. These global modification tasks mainly consist of white balancing and color correction. These tasks are usually applied as pixel-wise modifications and applied on each pixel separately. To follow this process we used a pixel-wise network consisting of 1x1 Convolution layers to ensure only pixel modification and eliminate the pixel shit that can happen from miss-alignment of GT and input.

Our module consists of 3 parts the first part project the input raw image into a working space that embeds each pixel into a k-dimensional vector by 1 x 1 convolution layers. The second part is the modification vector that was created by an encoding network that takes the guidance image and encodes it to a k-dimensional vector. That modification vector will be used to apply the color modification to the input image projected into the working space by direct multiplication. The modification vector is created by EncodeNet which is a similar structure to GuideNet with big kernel convolution and average pooling. After the multiplication operation, We project the image again to the RGB space using another projection network. The projection process to higher feature space allows us to apply different color modification operations like white balance and color correction with just a simple vector multiplication. For our working space size, we chose 64 as the size of the vectors.

For the guidance image, we first used the raw image patch as input to test the benefits of applying color reconstruction with a separate module. But because color reconstruction is a global operation that depends on the information from the whole scene, in our final model we used the resized full raw image as input. Our design allowed us to encode the full raw image into a modification vector and use it to apply the color modification. This allowed us to utilize the global information from the full raw image with a small computation cost. The use of the full raw image as guidance for global information was the third modification we proposed. The combination between the CMod and the full raw image guidance increased the performance of the network immensely and allowed for much better color reproduction as we will describe in the results. The next equations describe the operations in the CMod.

$$X_v = P_{raw \rightarrow working}(X) \tag{5.8}$$

$$mv = E(g) \tag{5.9}$$

$$X_{mv} = X_v \times mv \tag{5.10}$$

$$y^c = P_{working \rightarrow RGB}(X_{mv}) \tag{5.11}$$

The Final ISP Network consists of CMod that reproduces a colored image without any texture or content modifications just pixel-wise color adjustment then the colored image input to LiteISP that reconstructs the final image. The input for our model is a simple demosaiced version using bi-linear interpolation of the input raw patch. We used a demosaiced image to insure the same size throughout the network to be able to compare the output of the CMod Network with the Algined GT. For the guide image in our final model, we used a resized version of the full raw image. To resize the raw images we split the image into 4 channels one for each filter array (RGBG for Bayer filter), and resize each band separately to avoid mixing information of different bands and destroying the raw pattern. For the final inference on the full image, the input will be the resized full raw image as a guide image and the full raw image in full resolution for the ISP Network.

### 5.2.3 Loss Functions

For the loss functions each module has its own loss term and they are joined at the end to optimize the network end to end. For the GCM, we use the L1 loss between the output of GCM $\widetilde{y}$ and the aligned GT $y^w$ as GCM loss. m is the alignment mask indicating valid positions of the optical flow and $\circ$ is element-wise multiplication.

$$m_i = \begin{cases} 1, [\omega(1, \Psi)]_i \geq 1 - \epsilon \\ 0, otherwise \end{cases} \tag{5.12}$$

$$L_{GCM} = ||m \circ (\widetilde{y} - y^w)||_1 \tag{5.13}$$

For Cmod Loss we used the color difference loss as CDNet output in addition to l1 loss. The loss is computed between the output of the CMod $y^c$ and the aligned GT $y^w$

$$L_{CMod} = \lambda_{CD} CD(y^c, y^w) + \lambda_{l_1} ||m \circ (y^c - y^w)||_1 \tag{5.14}$$

Finally, for the ISP we used the same loss term as the baseline in addition to CD loss with CDNet between the output of the ISPNetwork and the aligned GT.

$$L_{ISP} = \lambda_{l_1} ||m \circ (\hat{y} - y^w)||_1 + \lambda_{VGG} ||m \circ (\phi(\hat{y}) - \phi(y^w))||_1 + \lambda_{CD} CD(\hat{y}, y^w) \tag{5.15}$$

The overall loss term that is used for the optimization is the combination of all the loss terms.

$$L = L_{GCM} + L_{CMod} + L_{ISP} \tag{5.16}$$

For our experiments, we trained all models with the same setup to ensure measuring the difference of our additional components only. We trained our models for 200 epochs with ADAM Optimizer with lr of 1e-4 with a lr decay to half every 40 epochs. The models were trained with a batch size of 4 on NVIDIA GeForce RTX 3090 24 GB GPU.

#### 5.2.3.1 Dataset

Because our idea depends on the existence of the full raw image in the dataset we were very limited with the dataset we can use. The only datasets that contain the full raw images are the test set from Zurich RAW to RGB dataset Ignatov et al. (2020b) with 148 full images which is the same dataset we used in ISP Data Limitations chapter experiments 4. We processed it in the same way we described there and it is called **Zurich RAW to RGB small**. In addition to ISPIW dataset Shekhar Tripathi et al. (2022) that the authors provided the full images and image patches after alignment for (192 full images). The problem with ISPIW the authors didn't provide the processing code of the dataset and the dataset missed some details like the split the authors used. So for our experiment, we focused on the use of Zurich RAW to RGB small.

## 5.3 Results

In this section, we will present the improvement of each of our modifications and the performance of our final model compared to the baseline. The ISPIW dataset is a new dataset and the processing code for this dataset and training/validation/test splits are not available which doesn't allow us to compare directly with the authors.

Also, the small set from Zurich RAW to RGB dataset also doesn't allow us to compare with the other available methods because the part we used to create the training and testing is the test set from the original dataset which is much smaller than the full dataset. So for our experiment, we tested the baseline with our dataset and then compared it to the performance of the model after our modifications and used the improvement of the performance as an evaluation for our modifications.

Also to overcome the problem of miss-alignment in the evaluation set that we discussed in the previous chapter 4.5 we aligned the GT in the evaluation stage similar to the training process. We used the GCM and optical flow network similar to the training process to align the GT and we evaluated the performance on the aligned GT. This will give us better quantitative evaluation values from the evaluation metrics that will reflect the visual quality of the evaluated models.

### 5.3.1 Our Model Performance

**Table 5.1:** *The performance of each of our modifications in comparison with the baseline. This experiment was conducted on Zurich RAW to RGB Small*

| Model | PSNR | SSIM | LPIPS | E00 |
|---|---|---|---|---|
| Baseline | 22.18 | 0.8305 | 0.162 | 11.610 |
| Baseline With CDNet | 22.37 | 0.8303 | 0.166 | 11.296 |
| Ours with raw patch guide | 22.54 | 0.8441 | 0.146 | 11.049 |
| Ours with full raw image guide | 24.79 | 0.8593 | 0.135 | 9.486 |

As we see in Tab. 5.1 we evaluated the performance of the model with each modification we applied. We also tested the performance on a list of comprehensive metrics to measure the different aspects of improvements, like texture, color, and so on. We used 4 different metrics to measure the difference to our GT. First PSNR which is our overall metric, is a pixel-wise metric and is affected by all the aspects of the images. Then SSIM Wang et al. (2004) and LPIPS Zhang et al. (2018a) which are mostly used for the perceptual difference of the images and will help us identify the texture and general reconstruction performance of our model. LPIPS works with the same concept as VGG loss. It uses the difference between the feature maps extracted from the compared images as the perceptual difference between them. These metrics also include color information in the difference. But according to the extensive comparison of CD metrics on natural images from Wang et al. (2023) SSIM and LPIPS are not good CD metrics for natural images, especially with non-perfectly aligned images (Which is the case for us), and other CD mercies perform much better. For that case, we used a separate

CD metric for our evacuation because our focus was the color reproduction and our assumption that the color reproduction will be the biggest improvement of our ISP Network. We choose CIEDE2000 Luo et al. (2001) as our CD metric as it shows a very good performance for CD in natural images and also a good robustness with misalignment.

First, we tested the performance of the baseline model with our dataset to compare the performance of our additional improvements. Then, we tested the addition of a specific CD loss to improve the color reproduction of the model. As we see from the results we gained some improvement, especially with the CD metric without much change in the other aspects of the output image. This improvement wasn't that noticeable as the color information is also included in the optimization of the pixel-wise losses like L1 loss and with the alignment loss it can be enough for color optimization. But there are benefits of having a specific CD loss as it adds more weight in the optimization process to the colors.

After we tested our ISP Model with the CMod but first without the full raw image and used the input raw patch as the guide image instead. We did that to see the effect of a specific module to apply color reproduction. We also see an extra improvement by having a separate color module to reconstruct colors separately from the main model. This time we see more improvement in all metrics which shows that having the color reproduction separate help the main model focus more on the other aspects. But the biggest improvement is still with the CD metric. These results showed us the benefit of processing global information like color reproduction separately in addition to having a full view of the input image is crucial for this kind of task. So the guidance branch is very helpful for global tasks.

But even with the full view of the input raw patch, you are still limited to the information from this patch only which miss much more important global information from the rest of the scene. So for our next step, we used the full raw image as the guidance image to have better global information from the whole scene. This addition improved the performance of our model immensely with a performance boost of +2.2 dB from the baseline. We also saw improvement in all measured metrics but by far the biggest improvement was in the CD metric which proves the importance of global information for accurate color reproduction. Training only on image patches is very limiting, especially with the global tasks in the ISP.

Even though the biggest improvement came from the use of the full raw image during training, the other modifications we applied especially the CMod were a crucial part of this improvement. They allowed us to integrate the full raw image in the training accurately and focus it on the part of the ISP that depend on global information the most. Also allowed us to integrate it in an efficient way that will not hinder the model's performance. This allowed us to merge between image


**(a)** *LiteISP Baseline* **(b)** *BaseLine + CDNet* **(c)** *Ours + Patch Guide* **(d)** *Ours + Full Image Guide* **(e)** *DSLR GT*

**Figure 5.4:** *The output of our model in comparison to the baseline.*

patches and full images for better reconstruction globally and locally.

To better investigate the output of our model we investigated the output images. We compared the model output of our model with its different modification stages with the baseline output (same models compared in Tab. 5.1). An example is shown in Fig. 5.4. As we notice for the output, our final model (Ours + Full Image Guide) is the best in all aspects. The output colors are much less washed out and flat compared to the other models and the colors are closer to the GT (the sign in image 3). We can also notice the importance of including the full image for white balancing in the rendering of the sky (images 2 and 3). Our model is much better are rendering the sky as the output looks more natural in comparison to the other models which look more yellow and overexposed. From the samples, we can also notice better color reproduction when we use our CMod without the full image which shows the benefits of having a separate module for global tasks with a

view over the entire input image. Additionally, we can notice some improvement in other aspects including the local features as we see in the third image which is a better rendering of the tree branches. These results confirm the importance of global information and the limitation of depending on image patch training only. This also shows the effectiveness of our proposed model to overcome this issue.

Additionally, We notice from the results even though the colors of our model are very close to the GT there is still some difference. This can be related to the different color processing inside each camera as each camera has its own processing and its own look and the model needs to learn this mapping to better match the GT. So even though having the full raw image helps the model to learn this mapping, this mapping is also very dependent on the GT color variety during training which depends on the data variety and size and our dataset. This can be improved with more variant datasets and better models to learn better mapping, especially since our dataset is very small (148 full images) so the performance will still be limited.

**Table 5.2:** *The model size and inference time comparison between our model and the baseline.*

| Model | Model Size | Inference Time |
|---|---|---|
| Baseline | 9.048 M | 17.629 ms |
| Ours with full raw image guide | 8.915 M | 30.072 ms |

In Tab.5.2 we can see the size and inference time difference between our model and the baseline test on Nvidia GeForce RTX 3090. The inference time of our model is still very efficient. The time increase is mostly coming from having the input as the demosaiced raw instead of the original raw. This modification also decreased the model size because the up-scaling part at the end is not needed. The input size (demosaiced raw) and the output size are the same. A further modification to the architecture can be applied to use the input raw instead of the demosaiced raw and will decrease the time difference immensely. This result shows that our module can improve the model accuracy immensely without affecting the model speed performance much. Also, additional simple modifications can be applied to reduce the difference even more.

### 5.3.2 CMod With other ISP Networks

Our proposed module (CMod) is a separate module from the main architecture and can be integrated with any ISP Network to have better color reproduction and more accurate output. To test that we added our CMod to another ISP Network

and evaluated the accuracy difference. We choose MicroISP Ignatov et al. (2022b) for that test.

**Table 5.3:** *The performance of the addition of our module (CMod) to MicroISP. This experiment was conducted on Zurich RAW to RGB Small*

| Model | PSNR | SSIM | LPIPS | E00 |
|---|---|---|---|---|
| MicroISP | 20.30 | 0.7826 | 0.264 | 13.150 |
| MicroISP with Cmod (full raw image) | 22.04 | 0.8145 | 0.255 | 11.356 |

We followed the same evaluation process as before. For the baseline, we used MicroISP with alignment loss and trained it on our dataset (Zurich RAW to RGB Small). Then we added CMod with full raw image guidance to the MicroISP and measured its performance. As we see from the results in Tab. 5.3, The addition of our module largely improved the model performance. The results are similar to the previous ISP Network with improvement in all tested metrics with the biggest improvement in the color difference. This shows the generalization of our module and the ability to integrate it with any ISP Network with minimal to no modifications required.

## 5.4 Discussion

As we saw from our experiments it is important to include the information from the full image and don't depend on image patch training only. Our proposed module includes the global information from the full image which largely improves the performance of the ISP Network. We also showed that our module can be integrated with any ISP Network and will produce the same improvement.

In our experiments, we were very limited with the dataset we can use because most of the available datasets only include the processed patches without the full images. This limitation can reduce the research innovation as the full images can be integrated during the process as we showed with our method. The authors of the available ISP datasets should make all the components of the ISP dataset available or a comprehensive dataset should be created to push the research forward. Even though we achieved very good results with the limited dataset available the model performance is still limited because of the lack of variety. The color reproduction of our model still struggles in some cases because the color mapping between the DSLR and phone depends on the dataset. The availability of a big dataset with the full raw images will be required to create a reliable and robust color reproduction. Additionally, with the availability of a larger dataset, it will be interesting to

see if the big size of the dataset can decrease the need for the full image global information.

# 6 Discussion

Even though with the current advances and improvements in the learned ISP models it is still far from production. The current state of learned-based ISP lacks a lot of fundamental components compare to the conventional ISP. This approach is very data dependent so in order to obtain a robust model for all the expected scenarios you need to include them in your data which is highly not achievable. This decreases the reliability of the ISP and increases the outliers. Also the use of deep learning to replace the full pipeline limit the ability of the manufacturer to tune the ISP. The conventional isp includes different components that each get tuned to create the output. That is hard to achieve with neural networks as we will be limited with the data we have and can create.

Additionally learned-based ISP currently can only render the automatic ideal settings without any modifications in capturing settings like adjusted exposure or white balance. This is the case because of the nature of the current isp dataset and models. So the current learned-based ISP models can only simulate the automatic settings in the camera. This limits the creative process with the camera which is a crucial part of photography. One of the important advantages of having different modules in the ISP is reusability. Some of the components in the ISP are sensor-specific but others are not and can be used in other different camera systems. This allows the share of modules between different camera systems which reduces the development time of the pipeline. This is not the case with learned-based ISP as the dataset is sensor specific. So to develop another learned-based ISP for a different camera you need to collect a new dataset which is very costly and time-consuming.

These issues are limiting the possibility of starting to use learned-based ISP in production. Even though learned-based ISP will be hard to replace the conventional complex ISP but it can be helpful in other aspects. Leaned-based ISP can be a replacement for the ISP pipelines in budget mobile cameras. Budget mobiles are very limited with hardware to reduce their cost. So manufacturers end up using a simple ISP in these phone cameras. So the use of leaned based ISP, in this case, will be beneficial as they are much cheaper and less complex to create. With the current advances in efficient ISP models, they can run very efficiently on general

hardware without the need for a separate ISP chip. This will be a good balance between performance and cost which will create a better camera on a budget.

We can also utilize leaned-based ISP as a tool to improve the current conventional ISP modules. We can investigate the patterns that the neural network learns to map from raw to RGB. These patterns can help us understand new important aspects of image reconstruction and help us reconfigure the modules in the conventional ISP. Like combining different modules together or having a separate module to address a specific aspect. This can help us discover new clues in the images that help in reconstruction that we didn't know before.

Also, advances in the ISP models can be used to improve ISP modules. As the color module we proposed can be applied in image enhancement models for better performance. So there is value to continue the improvement in this research direction and will help improve the state of camera quality overall.

## 6.1 Future Work

During our work, we noticed some directions that need to be investigated to improve the learned-based ISP. By addressing these issues we can close the gap between learned based ISP and conventional ISP.

**Flexible learned-based ISP**: The ISP models need to be more flexible and be able to process different raw images with different settings. Also, give the option for different tunings like different color mapping for different aesthetics. We can achieve that by creating a model that can accept additional inputs as processing options. A new dataset with more variety in the setting in both input and output will be valuable in this regard (Ex: phone raw images and DSLR images with different exposure values).

**Explainable ISP**: It will be valuable to explain based on what, the ISP model applies its processing. This can be very helpful for debugging and reducing the outliers and making sure of the reliability of the output. This can also help us to discover new patterns in the reconstruction process that can help us reconfigure and improve the conventional ISP structure.

**Global Data Generation Pipeline**: Our proposed data generation pipeline only works with a specific DSLR camera. It is possible to extend this approach to map from any DSLR image to our phone image and create synthetic data from any DSLR images. This can be achieved by an unsupervised generation method like GAN Ignatov et al. (2018) and comprehensive and variant DSLR and phone RGB images for enough generalization. This will open a huge amount of data that can be utilized to train a robust ISP Network.

**Sensor Independent Leaned based ISP**: At the current state if we want to develop a learned-based ISP for a specific camera we need to create a specific

dataset for it. If we can develop a sensor-independent method we can avoid this issue. We can achieve that with a robust model and a big dataset that will allow enough generalization. We can also try to split between sensor-dependent and sensor-independent parts in the raw construction to be able to utilize the datasets available with different cameras.

# 7 | Conclusion

In this work, we investigated the learned-based ISP process as an alternative to the handcrafted conventional ISP process. We investigated the current state of learned-based ISP, especially for mobile phones. We searched for the most common problems in the learned ISP methods and datasets that hinder their performance and limit them in comparison to the conventional ISP. We worked on different aspects of these problems to improve the accuracy and robustness of learned-based ISP methods.

The first problem we worked on was the rendering of night images. We proposed a way that utilizes Knowledge distillation to train an efficient ISP Network that can render night images. Our proposed method utilized a pre-trained teacher model to generate the dataset without needing to annotate our own data. Our proposed ISP Network was one of the top 10 methods in NTIRE 2023 Challenge on Night Photography Rendering Shutova et al. (2023) with a 1000x smaller model than the teacher network and 10x speed. Our model was the only one in the top solution that focused on efficiency. Investigation of other ISP Networks as a student network for knowledge distillation will be required to improve our model performance.

The second problem we worked on is the problem of datasets used for learned-based ISP methods. Because the datasets require the use of 2 different camera systems they suffer from miss-alignment. Additionally, the collection process is complex and time-consuming which limits the size and variability of the datasets. We proposed a novel pipeline that can be used to generate a high-quality fully aligned synthetic ISP dataset from a weakly aligned ISP dataset. Our results showed the quality of the synthetic dataset generated from our pipeline. Our experiments showed that an ISP model trained on our full synthetic data achieve better performance than the one trained on weakly aligned ISP data. Our experiments also showed the ability of our pipeline to generalize for other DSLR images (from the same camera model). After developing the pipeline only DSLR RGB images (from the same camera model) will be required to produce a synthetic dataset. Our pipeline largely decreases the time required to generate the ISP dataset and increases the available ISP data.

For the last problem, we worked on bad color reproduction in Learned-based ISP

methods. We proposed a new module that can be integrated with any ISP Network that combines between the full raw image and the raw patch image for better global and local reconstruction. Our experiments showed the importance of global information from the full image for accurate color reproduction. Our comprehensive evaluation showed that our module improves the ISP Network accuracy immensely in all aspects with the biggest improvement in color reproduction.

Our work investigated different aspects to improve the overall performance, accuracy, and robustness of the learned-based ISP approach. Our work improved the state of different aspects of the Learned-based ISP. Learned based ISP process still in the early research stage and more effort needs to be done and a lot more problems need to be solved before seeing this kind of approach in production.

# Bibliography

(2023). Grammarly: https://www.grammarly.com. [Online; accessed 10-August-2023]. (cited on page 7)

Abdelhamed, A., Afifi, M., Timofte, R., and Brown, M. S. (2020). Ntire 2020 challenge on real image denoising: Dataset, methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 496–497. (cited on page 25)

Afifi, M., Barron, J. T., LeGendre, C., Tsai, Y.-T., and Bleibel, F. (2021a). Cross-camera convolutional color constancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1981–1990. (cited on page 12)

Afifi, M., Brubaker, M. A., and Brown, M. S. (2021b). Histogan: Controlling colors of gan-generated and real images via color histograms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7941–7950. (cited on page 69)

Bychkovsky, V., Paris, S., Chan, E., and Durand, F. (2011). Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 26)

Cheng, D., Prasad, D. K., and Brown, M. S. (2014). Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *JOSA A*, 31(5):1049–1058. (cited on page 25)

Conde, M. V., Timofte, R., Huang, Y., Peng, J., Chen, C., Li, C., Pérez-Pellitero, E., Song, F., Bai, F., Liu, S., et al. (2022). Reversed image signal processing and raw reconstruction. aim 2022 challenge report. In *European Conference on Computer Vision*, pages 3–26. Springer. (cited on pages 50, 53, and 96)

Dai, L., Liu, X., Li, C., and Chen, J. (2020). AWNet: Attentive wavelet network for image ISP. In *Computer Vision – ECCV 2020 Workshops*, pages 185–201. Springer International Publishing. (cited on page 19)

Delbracio, M., Kelly, D., Brown, M. S., and Milanfar, P. (2021). Mobile computational photography: A tour. *Annual Review of Vision Science*, 7:571–604. (cited on pages 9, 10, 11, 12, 13, and 95)

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016). Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*. (cited on page 51)

Ehret, T., Davy, A., Arias, P., and Facciolo, G. (2019). Joint demosaicking and denoising by fine-tuning of bursts of raw images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8868–8877. (cited on page 12)

Ershov, E., Savchik, A., Shepelev, D., Banić, N., Brown, M. S., Timofte, R., Koščević, K., Freeman, M., Tesalin, V., Bocharov, D., Semenkov, I., Subašic, M., Lončarić, S., Terekhin, A., Liu, S., Feng, C., Wang, H., Zhu, R., Li, Y., Lei, L., Li, Z., Yi, S., Han, L.-H., Wu, R., Jin, X., Guo, C., Kinli, F., Menteş, S., Özcan, B., Kıraç, F., Zini, S., Rota, C., Buzzelli, M., Bianco, S., Schettini, R., Li, W., Ma, Y., Wang, T., Xu, R., Song, F., Chen, W.-T., Yang, H.-H., Huang, Z.-K., Chang, H.-E., Kuo, S.-Y., Liang, Z., Zhou, S., Feng, R., Li, C., Chen, X., Song, B., Zhang, S., Liu, L., Wang, Z., Ryu, D., Bae, H., Kwon, T., Desai, C., Akalwadi, N., Joshi, A., Mandi, C., Malagi, S., Uppin, A., Reddy, S. S., Tabib, R. A., Patil, U., and Mudenagudi, U. (2022). Ntire 2022 challenge on night photography rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1287–1300. (cited on pages 24, 34, and 95)

Gehler, P. V., Rother, C., Blake, A., Minka, T., and Sharp, T. (2008). Bayesian color constancy revisited. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE. (cited on page 25)

Gharbi, M., Chaurasia, G., Paris, S., and Durand, F. (2016). Deep joint demosaicking and denoising. *ACM Transactions on Graphics (ToG)*, 35(6):1–12. (cited on pages 3 and 9)

Gijsenij, A., Gevers, T., and Van De Weijer, J. (2011). Computational color constancy: Survey and experiments. *IEEE transactions on image processing*, 20(9):2475–2489. (cited on page 12)

Gou, J., Yu, B., Maybank, S. J., and Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819. (cited on pages 37 and 95)

Guo, C., Li, C., Guo, J., Loy, C. C., Hou, J., Kwong, S., and Cong, R. (2020). Zero-reference deep curve estimation for low-light image enhancement. In *Proceedings*

*of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1780–1789. (cited on page 35)

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. (cited on page 25)

Hirakawa, K. and Parks, T. W. (2006). Joint demosaicing and denoising. *IEEE Transactions on Image Processing*, 15(8):2146–2157. (cited on page 14)

Hsyu, M.-C., Liu, C.-W., Chen, C.-H., Chen, C.-W., and Tsai, W.-C. (2021). CSANet: High speed channel spatial attention network for mobile ISP. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. (cited on pages 22 and 23)

Hu, Y., Wang, B., and Lin, S. (2017). Fc4: Fully convolutional color constancy with confidence-weighted pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4085–4094. (cited on pages 12 and 25)

Ignatov, A., Chiang, C.-M., Kuo, H.-K., Sycheva, A., and Timofte, R. (2021). Learned smartphone isp on mobile npus with deep learning, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2503–2514. (cited on page 22)

Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., and Van Gool, L. (2017). Dslr-quality photos on mobile devices with deep convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 3277–3285. (cited on pages 16 and 38)

Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., and Van Gool, L. (2018). Wespe: weakly supervised photo enhancer for digital cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 691–700. (cited on pages 52, 53, 61, 80, and 96)

Ignatov, A., Malivenko, G., Timofte, R., Tseng, Y., Xu, Y.-S., Yu, P.-H., Chiang, C.-M., Kuo, H.-K., Chen, M.-H., Cheng, C.-M., and Gool, L. V. (2022a). PyNet-v2 mobile: Efficient on-device photo processing with neural networks. In *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE. (cited on page 23)

Ignatov, A., Sycheva, A., Timofte, R., Tseng, Y., Xu, Y.-S., Yu, P.-H., Chiang, C.-M., Kuo, H.-K., Chen, M.-H., Cheng, C.-M., et al. (2022b). Microisp: processing 32mp photos on mobile devices with deep learning. In *European Conference on Computer Vision*, pages 729–746. Springer. (cited on pages 23, 28, 36, 37, 77, and 95)

Ignatov, A., Timofte, R., Ko, S.-J., Kim, S.-W., Uhm, K.-H., Ji, S.-W., Cho, S.-J., Hong, J.-P., Mei, K., Li, J., Zhang, J., Wu, H., Li, J., Huang, R., Haris, M., Shakhnarovich, G., Ukita, N., Zhao, Y., Po, L.-M., Zhang, T., Liao, Z., Shi, X., Zhang, Y., Ou, W., Xian, P., Xiong, J., Zhou, C., Yu, W. Y., Yubin, Y., Hou, B., Park, B., Yu, S., Kim, S., and Jeong, J. (2019a). AIM 2019 challenge on RAW to RGB mapping: Methods and results. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE. (cited on page 18)

Ignatov, A., Timofte, R., Kulik, A., Yang, S., Wang, K., Baum, F., Wu, M., Xu, L., and Van Gool, L. (2019b). Ai benchmark: All about deep learning on smartphones in 2019. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3617–3635. IEEE. (cited on page 21)

Ignatov, A., Timofte, R., Liu, S., Feng, C., Bai, F., Wang, X., Lei, L., Yi, Z., Xiang, Y., Liu, Z., et al. (2022c). Learned smartphone isp on mobile gpus with deep learning, mobile ai & aim 2022 challenge: report. In *European Conference on Computer Vision*, pages 44–70. Springer. (cited on pages 22, 44, and 96)

Ignatov, A., Timofte, R., Zhang, Z., Liu, M., Wang, H., Zuo, W., Zhang, J., Zhang, R., Peng, Z., Ren, S., Dai, L., Liu, X., Li, C., Chen, J., Ito, Y., Vasudeva, B., Deora, P., Pal, U., Guo, Z., Zhu, Y., Liang, T., Li, C., Leng, C., Pan, Z., Li, B., Kim, B.-H., Song, J., Ye, J. C., Baek, J., Zhussip, M., Koishekenov, Y., Ye, H. C., Liu, X., Hu, X., Jiang, J., Gu, J., Li, K., Tan, P., and Hou, B. (2020a). AIM 2020 challenge on learned image signal processing pipeline. In *Computer Vision – ECCV 2020 Workshops*, pages 152–170. Springer International Publishing. (cited on pages 19, 25, 26, and 67)

Ignatov, A., Van Gool, L., and Timofte, R. (2020b). 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. (cited on pages 3, 17, 18, 19, 23, 27, 43, 44, 49, 50, 53, 54, 55, 63, 72, 95, and 96)

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer. (cited on page 38)

Kim, B.-H., Song, J., Ye, J. C., and Baek, J. (2020). PyNET-CA: Enhanced PyNET with channel attention for end-to-end mobile image signal processing. In *Computer Vision – ECCV 2020 Workshops*, pages 202–212. Springer International Publishing. (cited on page 19)

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. (cited on page 38)

Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31. (cited on page 51)

Li, X., Gunturk, B., and Zhang, L. (2008). Image demosaicing: A systematic survey. In *Visual Communications and Image Processing 2008*, volume 6822, pages 489–503. SPIE. (cited on page 11)

Li, Z. and Ma, Z. (2021). Robust white balance estimation using joint attention and angular loss optimization. In *Thirteenth International Conference on Machine Vision*, volume 11605, pages 401–406. SPIE. (cited on page 25)

Li, Z., Yi, S., and Ma, Z. (2022). Rendering nighttime image via cascaded color and brightness compensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 897–905. (cited on page 25)

Liang, Z., Cai, J., Cao, Z., and Zhang, L. (2019). Cameranet: A two-stage framework for effective camera isp learning. *IEEE Transactions on Image Processing*, 30:2248–2262. (cited on page 16)

Lim, S. and Kim, W. (2020). Dslr: Deep stacked laplacian restorer for low-light image enhancement. *IEEE Transactions on Multimedia*, 23:4272–4284. (cited on page 35)

Liu, L., Jia, X., Liu, J., and Tian, Q. (2020). Joint demosaicing and denoising with self guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2240–2249. (cited on page 11)

Liu, P., Zhang, H., Lian, W., and Zuo, W. (2019). Multi-level wavelet convolutional neural networks. *IEEE Access*, 7:74973–74985. (cited on page 54)

Liu, S., Feng, C., Wang, X., Wang, H., Zhu, R., Li, Y., and Lei, L. (2022). Deepflexisp: A three-stage framework for night photography rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1211–1220. (cited on pages 25, 26, 37, and 42)

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110. (cited on pages 27, 43, 47, 53, and 55)

Luo, M. R., Cui, G., and Rigg, B. (2001). The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 26(5):340–350. (cited on pages 31 and 74)

Mei, K., Li, J., Zhang, J., Wu, H., Li, J., and Huang, R. (2019). HighEr-resolution network for image demosaicing and enhancing. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE. (cited on page 18)

Ortiz-Jaramillo, B., Kumcu, A., Platisa, L., and Philips, W. (2019). Evaluation of color differences in natural scene color images. *Signal Processing: Image Communication*, 71:128–137. (cited on page 31)

Qi, Y., Yang, Z., Sun, W., Lou, M., Lian, J., Zhao, W., Deng, X., and Ma, Y. (2021). A comprehensive overview of image enhancement techniques. *Archives of Computational Methods in Engineering*, pages 1–25. (cited on page 2)

Raimundo, D. W., Ignatov, A., and Timofte, R. (2022). Lan: Lightweight attention-based network for raw-to-rgb smartphone image processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 808–816. (cited on page 23)

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer. (cited on pages 16, 54, and 64)

Schwartz, E., Giryes, R., and Bronstein, A. M. (2019). DeepISP: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing*, 28(2):912–923. (cited on pages 15 and 16)

Shekhar Tripathi, A., Danelljan, M., Shukla, S., Timofte, R., and Van Gool, L. (2022). Transform your smartphone into a dslr camera: Learning the isp in the wild. In *European Conference on Computer Vision*, pages 625–641. Springer. (cited on pages 19, 20, 29, 58, 72, and 95)

Shutova, A., Ershov, E., Perevozchikov, G., Ermakov, I., Banić, N., Timofte, R., Collins, R., Efimova, M., Terekhin, A., Zini, S., Rota, C., Buzzelli, M., Bianco, S., Schettini, R., Lei, C., Wang, T., Wang, S., Liu, S., Feng, C., Shao, G., Wang, H., Wang, X., Lei, L., Xu, L., Zhang, C., Wang, Y., Guo, J., Sun, Y., Liu, T.,

Hao, D., Kınlı, F., Özcan, B., Kıraç, F., Chung, H., Lee, N., Kwak, S. K., Conde, M., Seizinger, T., Vasluianu, F., Elezabi, O., Hsieh, C.-H., Chen, W.-T., Yang, H.-H., Huang, Z.-K., Chang, H.-E., Chen, I.-H., Chen, Y.-C., and Chiang, Y.-C. (2023). Ntire 2023 challenge on night photography rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1981–1992. (cited on pages 26, 33, 41, 83, and 99)

Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. (2018). Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 66)

Süsstrunk, S., Buckley, R., and Swen, S. (1999). Standard rgb color spaces. In *Proc. IS&T;/SID 7th Color Imaging Conference*, volume 7, pages 127–134. (cited on page 13)

Truong, P., Danelljan, M., Van Gool, L., and Timofte, R. (2021). Learning accurate dense correspondences and when to trust them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5714–5724. (cited on pages 28 and 43)

Uhm, K.-H., Kim, S.-W., Ji, S.-W., Cho, S.-J., Hong, J.-P., and Ko, S.-J. (2019). W-net: Two-stage u-net with misaligned data for raw-to-RGB mapping. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE. (cited on page 18)

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors (2014). scikit-image: image processing in Python. *PeerJ*, 2:e453. (cited on page 31)

Vinker, Y., Huberman-Spiegelglas, I., and Fattal, R. (2021). Unpaired learning for high dynamic range image tone mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14657–14666. (cited on page 25)

Wang, Y., Huang, H., Xu, Q., Liu, J., Liu, Y., and Wang, J. (2020). Practical deep raw image denoising on mobile devices. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI*, pages 1–16. Springer. (cited on page 15)

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612. (cited on pages 30, 38, 73, and 95)

Wang, Z., Liu, J., Li, G., and Han, H. (2022). Blind2unblind: Self-supervised image denoising with visible blind spots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2027–2036. (cited on page 15)

Wang, Z., Xu, K., Yang, Y., Dong, J., Gu, S., Xu, L., Fang, Y., and Ma, K. (2023). Measuring perceptual color differences of smartphone photographs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. (cited on pages 31, 68, 69, 73, and 97)

Wei, K., Fu, Y., Yang, J., and Huang, H. (2020). A physics-based noise formation model for extreme low-light raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 14)

Xing, Y., Qian, Z., and Chen, Q. (2021). Invertible image signal processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6287–6296. (cited on pages 51, 53, and 96)

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018a). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595. (cited on page 73)

Zhang, Y., Li, D., Law, K. L., Wang, X., Qin, H., and Li, H. (2022). Idr: Self-supervised image denoising via iterative data refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2098–2107. (cited on page 14)

Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y. (2018b). Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301. (cited on page 19)

Zhang, Y., Qin, H., Wang, X., and Li, H. (2021a). Rethinking noise synthesis and modeling in raw denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4593–4601. (cited on page 15)

Zhang, Z., Wang, H., Liu, M., Wang, R., Zhang, J., and Zuo, W. (2021b). Learning raw-to-srgb mappings with inaccurately aligned supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4348–4358. (cited on pages 20, 44, 45, 46, 47, 54, 64, 65, 66, 96, and 97)

Zhao, Y., Po, L.-M., Zhang, T., Liao, Z., Shi, X., Zhang, Y., Ou, W., Xian, P., Xiong, J., Zhou, C., and Yu, W. Y. (2019). Saliency map-aided generative adversarial network for RAW to RGB mapping. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE. (cited on page 19)

Zini, S., Rota, C., Buzzelli, M., Bianco, S., and Schettini, R. (2022). Shallow camera pipeline for night photography rendering. *arXiv preprint arXiv:2204.08972*. (cited on page 25)

Zini, S., Rota, C., Buzzelli, M., Bianco, S., and Schettini, R. (2023). Back to the future: A night photography rendering isp without deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1465–1473. (cited on page 26)

BIBLIOGRAPHY

# List of Figures

LIST OF FIGURES

# List of Tables