

# RAWIW: RAW Image Watermarking Robust to ISP Pipeline

Kang Fu, Xiaohong Liu, Jun Jia, Zicheng Zhang, Yicong Peng,  
Jia Wang, and Guangtao Zhai, *Senior Member, IEEE*

arXiv:2307.15443v1 [eess.IV] 28 Jul 2023

**Abstract**—Invisible image watermarking is essential for image copyright protection. Compared to RGB images, RAW format images use a higher dynamic range to capture the radiometric characteristics of the camera sensor, providing greater flexibility in post-processing and retouching. Similar to the master recording in the music industry, RAW images are considered the original format for distribution and image production, thus requiring copyright protection. Existing watermarking methods typically target RGB images, leaving a gap for RAW images. To address this issue, we propose the first deep learning-based RAW Image Watermarking (RAWIW) framework for copyright protection. Unlike RGB image watermarking, our method achieves cross-domain copyright protection. We directly embed copyright information into RAW images, which can be later extracted from the corresponding RGB images generated by different post-processing methods. To achieve end-to-end training of the framework, we integrate a neural network that simulates the ISP pipeline to handle the RAW-to-RGB conversion process. To further validate the generalization of our framework to traditional ISP pipelines and its robustness to transmission distortion, we adopt a distortion network. This network simulates various types of noises introduced during the traditional ISP pipeline and transmission. Furthermore, we employ a three-stage training strategy to strike a balance between robustness and concealment of watermarking. Our extensive experiments demonstrate that RAWIW successfully achieves cross-domain copyright protection for RAW images while maintaining their visual quality and robustness to ISP pipeline distortions.

**Index Terms**—Robust watermarking, Image signal processing, Camera pipeline, Neural networks

## I. INTRODUCTION

IN the era of information, safeguarding intellectual property is vital, especially for multimedia resources on the Internet, such as recordings, photographs, and videos. Among these, photographs are widespread and require significant copyright protection. Invisible image watermarking plays a crucial role in the field of information security by securing the copyrights of digital images and tracing unauthorized disclosures. This technique involves creating a watermarked image that appears similar to the original image but contains unique copyright information. At the same time, the hidden watermarking must be robust enough to withstand noises and distortions that may occur during transmission, such as Gaussian noise and JPEG compression. A reliable watermarking algorithm should

Kang Fu, Jun Jia, Zicheng Zhang, Yicong Peng, Jia Wang, and Guangtao Zhai are with the Institute of Image Communication and Network Engineering Shanghai Jiao Tong University, 200240 Shanghai, China. E-mail: {fuk20-20, jiajun0302, zzc1998, jack-sparrow, jiawang, zhaiguangtao}@sjtu.edu.cn. Xiaohong Liu is with John Hopcroft Center, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: xiaohongliu@sjtu.edu.cn.

ensure that the hidden information remains detectable even after encountering these challenges. Traditionally, existing watermarking algorithms have focused on manipulating RGB images and hiding information in spatial or transform domains. However, professional photographers nowadays prefer working with RAW images to achieve more satisfactory results, as RAW images offer greater flexibility and ease of manipulation. Unlike RGB images, RAW images store 10-16 bits of unprocessed scene radiance, capturing a higher dynamic range and providing more room for post-processing. The popularity of smartphones like Huawei P20, iPhone 13 Pro, and Samsung Galaxy S22, which support capturing RAW images, indicates a potential surge in the number of RAW images in the near future. Moreover, RAW images are akin to works of art and should not be distributed without proper authorization. Publishers often receive RAW images and use them to generate different RGB images. As a result, applying watermarking algorithms to RAW images is critical for copyright protection but has not received sufficient attention in previous works. In conclusion, with the increasing prevalence of multimedia resources, particularly photographs, on the Internet, the importance of protecting intellectual property through invisible image watermarking cannot be overstated. Paying attention to RAW images and developing effective watermarking techniques for them is crucial in ensuring copyright protection in this digital age.

Unlike RGB images, the copyright protection of RAW images faces significant challenges. Applying watermarking directly to RGB images poses the risk of leaking the original RAW image, and the leaked RAW image can generate unwatermarked RGB images, thereby failing to achieve the goal of copyright protection. On the other hand, attempting to decode the watermarking from RAW images presents difficulties since RAW images are not readily accessible in most cases. The existing invisible image watermarking methods typically hide and extract information within the same image format, specifically in RGB color space. If we were to apply these existing methods directly to RAW images, extracting the watermarking from RAW images would be impractical due to limited accessibility. Additionally, a RAW image can be transformed into different RGB images by various Image Signal Processing (ISP) pipelines. Hence, the process of RAW image watermarking involves encoding the watermarking into a RAW image and subsequently decoding the watermarking from the corresponding RGB images after undergoing various ISP pipelines. This multistage process aims to ensure copyright protection while accounting for the different transformations

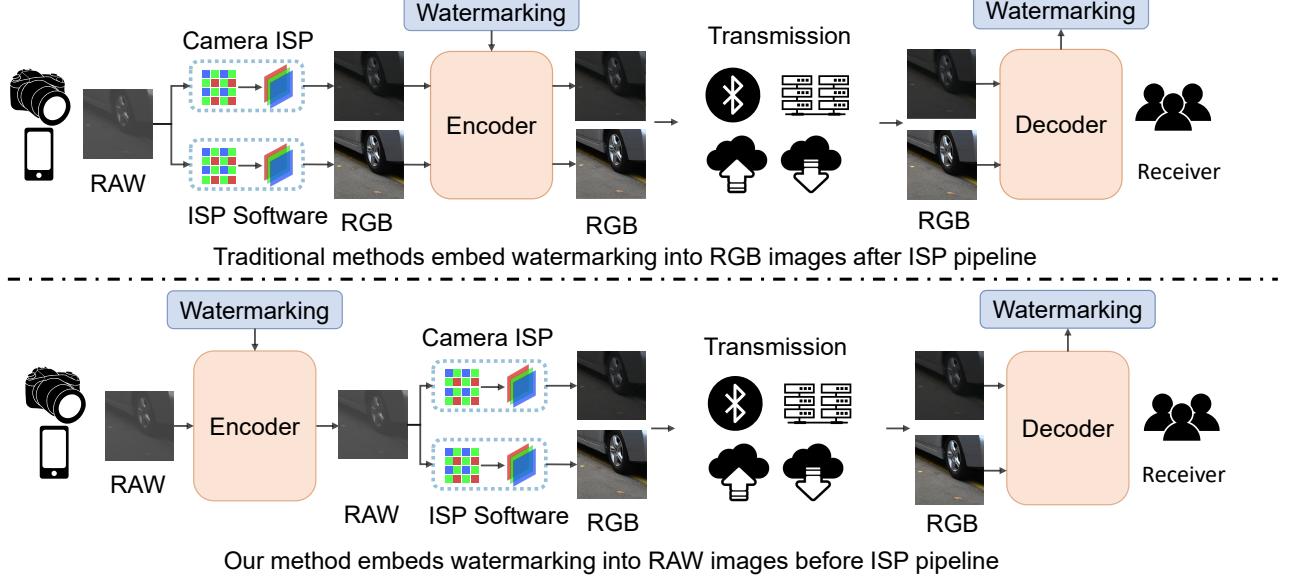


Fig. 1. Schematic diagram of our method compared with previous methods. Our method encodes the watermarking messages into RAW images before the ISP pipeline and decodes the generated RGB images to extract the watermarking. This approach allows the same RAW image to be used with different ISP pipelines, producing varieties of RGB images with different colors and qualities with the extractable watermarking.

that the RAW image undergoes during image processing. In summary, protecting the copyright of RAW images requires overcoming various challenges, such as preventing leakage of the original RAW image and handling the transformation of RAW images to different RGB images by various ISP pipelines. Existing watermarking methods designed for RGB images are not directly applicable to RAW images due to the differences in accessibility and image format, emphasizing the need for specialized techniques for RAW image watermarking.

To address the aforementioned challenges, Meerwald *et al.* [1] proposed frequency domain transform based RAW image watermarking. This paper introduces the pioneering deep learning watermarking-based copyright protection framework for RAW images, called RAW Image Watermarking (RAWIW). RAWIW utilizes Convolutional Neural Networks (CNN) for embedding and detecting the watermarking. The comparison between our RAWIW and the RGB watermarking method is illustrated in Fig 1. As discussed earlier, our proposed RAWIW encodes the watermarking information into the RAW image before applying the Image Signal Processing (ISP) pipeline. This approach enhances encoding efficiency by applying watermarking to the RAW image to undergo different retouching methods (*ISP pipelines*), producing RGB images with diverse color hues and tones. The watermark decoder can then extract the watermarking information from these RGB images, irrespective of the employed retouching. In contrast, RGB watermarking necessitates encoding the watermarking information into each RGB image generated by distinct retouching methods. This process is time-consuming and computationally intensive, as each RGB image must be processed individually. By embedding the watermarking into RAW images before applying ISP pipelines, our framework streamlines the encoding process and enhances encoding efficiency. Furthermore, our method enhances decoding robust-

ness by allowing the decoder to adapt to different retouching methods through flexible changes during the training pipeline. Consequently, our proposed method provides an effective solution for safeguarding the copyright of RAW images.

The RAWIW framework comprises five essential modules: encoder, decoder, discriminator, distortion network, and deep ISP pipeline. Within this framework, the encoder embeds watermarking information into a RAW image, while the decoder retrieves the hidden information from the corresponding RGB image. To ensure the encoded information's robustness during the transfer from RAW to RGB format, a deep differentiable ISP pipeline [2]–[4] is integrated, simulating camera image processing. Additionally, we incorporate a distortion network after the ISP module to simulate the distortions encountered during transmission, further enhancing the robustness of the watermarking against these distortions. The discriminator and encoder modules are trained adversarially to improve the concealment of the watermarking. Moreover, we employ an effective three-stage training strategy to strike a balance between the robustness and concealment of the watermarking. The framework has undergone evaluations using two datasets of RAW images, demonstrating outstanding performance in terms of decoding accuracy and visual quality. Overall, this paper presents a comprehensive approach for adding watermarking to RAW images, applicable in various scenarios, such as digital rights management and copyright protection. This robust and efficient framework can significantly contribute to safeguarding the copyright of RAW images in diverse applications.

The contributions of this paper are threefold:

- To the best of our knowledge, this paper presents the **first deep learning-based cross-domain RAW image watermarking method named RAWIW** which can encode watermarking to RAW images while decoding watermarking from the corresponding retouched RGB images.

This method can achieve the protection of copyright and ownership for RAW images.

- We are **the first to propose a RAW image encoder that considers Bayer patterns and a distortion network that simulates the gap of different ISP pipelines and the distortion of transmission** while we use an effective three-stage training strategy for our method, which can achieve a good trade-off between robustness and concealment of watermarking.
- Extensive experiments demonstrate that the proposed method has good concealment while being robust to different ISP pipelines.

The paper is structured as follows. Section II provides an overview of the background and related work related to the proposed method. In Section III, we elaborate on our proposed watermarking method in detail. Next, Section IV outlines the experimental setup and showcases the results obtained from our experiments. In Section V and Section VI, we discuss the advantages and limitations of the proposed method.

## II. RELATED WORK

### A. Invisible Information Hiding

Invisible information hiding can be broadly classified into two main categories: steganography and digital watermarking. Steganography finds widespread use in the field of information security, with its primary objective being to ensure that information is accessible only to the intended recipient while remaining concealed from unauthorized individuals. Steganography can be further divided into two classes: spatial domain and transfer domain. The classical spatial domain method is Least Significant Bit (LSB), where the hidden message replaces the least significant bits of the cover image. However, this approach alters the statistical properties of the cover image, making it easily detectable by steganalysis. As a result, simple LSB steganography is ineffective in practical applications. On the other hand, steganography in the transfer domain [5]–[11] leverages the statistical characteristics of the image to conceal information. In recent years, numerous steganographic techniques based on CNN have been introduced. The impressive non-linear fitting capability of CNN allows for embedding and extracting information without the need for intricate manual feature extractions. Some well-established methods, such as SSGAN [12] and ASDL-GAN [13], have proposed modifications to the redundant information of the cover image to hide the desired information. [14], [15] use CNN to realize image forgery detection and localization.

Digital watermarking is a crucial aspect of information hiding, involving the insertion of concise messages into images to protect copyright and assert authorship. Unlike steganography, digital watermarking requires a high level of robustness against transmission distortion. Traditional digital watermarking methods can be classified into two categories: spatial [16]–[19] and transfer [20]–[28] domain approaches.

Zhu *et al.* [29] presented an innovative approach for achieving robust image watermarking through adversarial learning. Their pioneering work demonstrated robustness against various distortions, including Gaussian blurring, pixelwise

dropout, cropping, and JPEG compression. Building upon Zhu's framework, Tancik *et al.* [30] introduced Stegastamp, which incorporated shooting noise to achieve robust watermarking for shooting screen and printed images. Jia *et al.* [31] proposed RIHOOP, utilizing differentiable 3-D rendering operations to simulate distortions resulting from camera imaging. However, unlike the aforementioned methods that embed the watermarking in RGB format, our method embeds the watermarking in RAW format and extracts it from RGB format. This distinction enables us to address specific challenges related to RAW images and attain effective copyright protection in diverse scenarios.

### B. Image Signal Processing Pipeline

The Image Signal Processing (ISP) pipeline in a camera is utilized to transform RAW images captured by the camera sensor into RGB images that are perceptually optimized for the Human Visual System (HVS). To achieve exceptional visual quality, the Camera ISP pipeline consists of various modules, such as demosaicing, white balance, color correction, color mapping, gamma correction, image enhancement, noise reduction, and sharpening. However, many of these sub-modules are non-differentiable, meaning that they do not allow for the backpropagation of gradients through them. This non-differentiability poses a challenge when attempting to train the complete ISP pipeline end-to-end in a neural network, hindering the optimization process. As a result, effectively incorporating the ISP pipeline into the neural network architecture requires specialized techniques to overcome these non-differentiable components and ensure smooth training and optimization.

In contrast to the traditional ISP pipeline, where each submodule is treated separately, the deep ISP pipeline operates on RAW images to produce RGB images using a deep neural network. Recent methodologies [32]–[34] based on CNN have demonstrated remarkable advancements in various ISP tasks, showcasing the superiority of CNN in this domain. Consequently, using a CNN instead of the entire ISP pipeline is feasible. Many efforts have been made in recent years to train deep networks to learn the ISP pipeline. Schwartz *et al.* [35] created a dataset containing RAW images and their corresponding RGB images and proposed the DeepISP model. This model establishes a mapping between RAW low-light images and well-lit processed RGB images. CameraNet [36] comprises two distinct CNN modules designed to address two sets of relatively uncorrelated subtasks in an ISP pipeline: restoration and enhancement. Ignatov *et al.* [37] introduced an inverted pyramidal architecture named PyNET, capable of processing images at five distinct levels, to learn a diverse set of features at each level. They also collected a dataset containing paired RAW and RGB images, which was subsequently utilized in two challenges [2], [38]. The top-performing methods in these challenges were MW-ISPNet [2] and AWNet [4], both using a Discrete Wavelet Transform (DWT)-based decomposition to replace upsampling and downsampling operations. MW-ISPNet integrates MWCNN [39] with RCAN citezhang2018rcan models, while

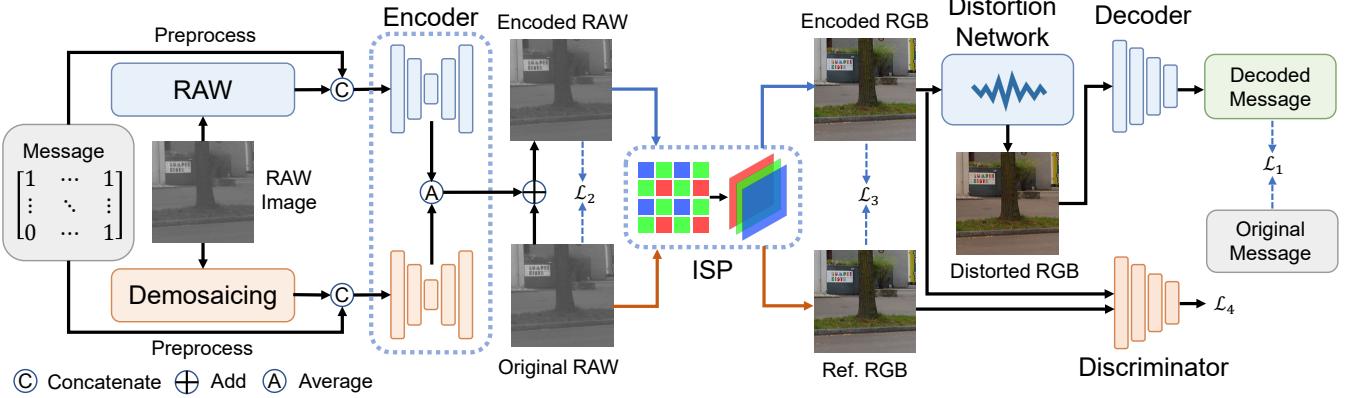


Fig. 2. Overall architecture of the RAWIW framework. RAWIW consists of three major modules: (1) the Encoder, Decoder and Discriminator forms the watermarking module, which embeds the watermarking message into a given RAW and decodes it from the converted RGB image. (2) the deep ISP module simulates the traditional ISP pipeline while keeping the process differentiable. (3) the Distortion Network adds noise and compression to the encoded RGB image, which simulates distortions introduced during the traditional ISP pipeline as well as transmission. Losses  $\mathcal{L}_{1\sim 4}$  are: watermarking message loss, RAW image loss, converted RGB image loss, and discriminator loss respectively.

AWNet employs an attention mechanism. Zhang *et al.* [40] introduced a light ISP network that builds upon the MW-ISPNet architecture and incorporates image alignment during training. This image alignment has led to the current state-of-the-art performance of the light ISP network.

The proposed RAWIW framework incorporates a deep ISP pipeline that represents the traditional ISP pipeline in a differentiable manner. This deep ISP pipeline is constructed using CNN, enabling end-to-end training of the complete framework. By utilizing a deep ISP pipeline, the proposed framework achieves a superior balance between accuracy and computational efficiency. The differentiable nature of the deep ISP pipeline enables efficient backpropagation of gradients throughout the entire framework, facilitating the optimization process and enhancing training effectiveness. In summary, the adoption of a deep ISP pipeline in the RAWIW framework not only enables end-to-end training but also improves the overall performance by efficiently managing computational resources and optimizing the training process.

### III. METHOD

In this section, we introduce the proposed RAW image watermarking method named RAWIW. The schematic overview of the proposed method, as depicted in Fig 2, comprises a watermarking module, a deep ISP module, and a distortion network. The watermarking module aims at embedding and extracting the watermarking, while the deep ISP module converts RAW images to RGB images. The distortion network introduces simulated noises into the framework, thereby enhancing the robustness of the watermarking against a range of adversarial attacks. The following subsections will describe these modules respectively in detail.

#### A. Problem Formulation

Our proposed methods embed a copyright message  $M$  within a RAW image  $\mathbf{R}$  via a RAW image encoder  $E$  with weights  $\theta_E$  to produce an encoded RAW image  $\mathbf{R}_e$ . The ISP

pipeline  $P$  is subsequently utilized to convert the encoded RAW image  $\mathbf{R}_e$  to an RGB image  $\mathbf{I}_e$  and also convert the original RAW image  $\mathbf{R}$  to RGB image  $\mathbf{I}_o$ . Finally, our proposed method utilizes an RGB image decoder  $D$  with weights  $\theta_D$  to extract the embedded message  $M_d$  from the RGB image  $\mathbf{I}_e$ . The above process can be expressed by the following four formulas:

$$\mathbf{R}_e = E_{\theta_E}(\mathbf{R}, M), \mathbf{I}_e = P(\mathbf{R}_e), \mathbf{I}_o = P(\mathbf{R}), M_d = D_{\theta_D}(\mathbf{I}_e). \quad (1)$$

For the purpose of the copyright protection for RAW images, it is imperative to achieve the congruence between  $M_d$  and  $M$ , while simultaneously ensuring that the visual quality of both images remain uncompromised. Therefore, we need to optimize the parameters during the training process:

$$\arg \min_{\theta_E, \theta_D} d_{rgb}(\mathbf{I}_o, \mathbf{I}_e) + d_{raw}(\mathbf{R}, \mathbf{R}_e) + d_{bin}(M, M_d), \quad (2)$$

where  $d_{rgb}$  stands for the RGB image distance, and analogously for  $d_{raw}$ ,  $d_{bin}$ .

#### B. Image Watermarking

The image watermarking module is composed of three parts: encoder, decoder, and discriminator. The encoder is responsible for embedding the watermarking information into the RAW image, while the decoder extracts the watermarking information from the corresponding RGB image. The discriminator and encoder modules are trained adversarially to enhance the concealment of the watermarking information.

**Encoder:** The primary function of the encoder is to integrate watermarking information into a RAW image  $\mathbf{R}$ , producing an encoded RAW image  $\mathbf{R}_e$  that exhibits the minimal perceptual differentiation compared to  $\mathbf{R}$ . Furthermore, the RGB images  $\mathbf{I}_e$  derived from  $\mathbf{R}_e$  exhibit the minimal perceptual differentiation compared to  $\mathbf{I}_o$ . The watermarking message  $M \in \{0, 1\}^L$  is represented as a binary vector of length  $L$ . This binary vector is first processed through a fully connected layer, resulting in a tensor  $\mathbf{T} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 1}$ . Subsequently, the tensor  $\mathbf{T}$  is upsampled to form a tensor  $\mathbf{T}' \in \mathbb{R}^{H \times W \times 1}$ , which is then

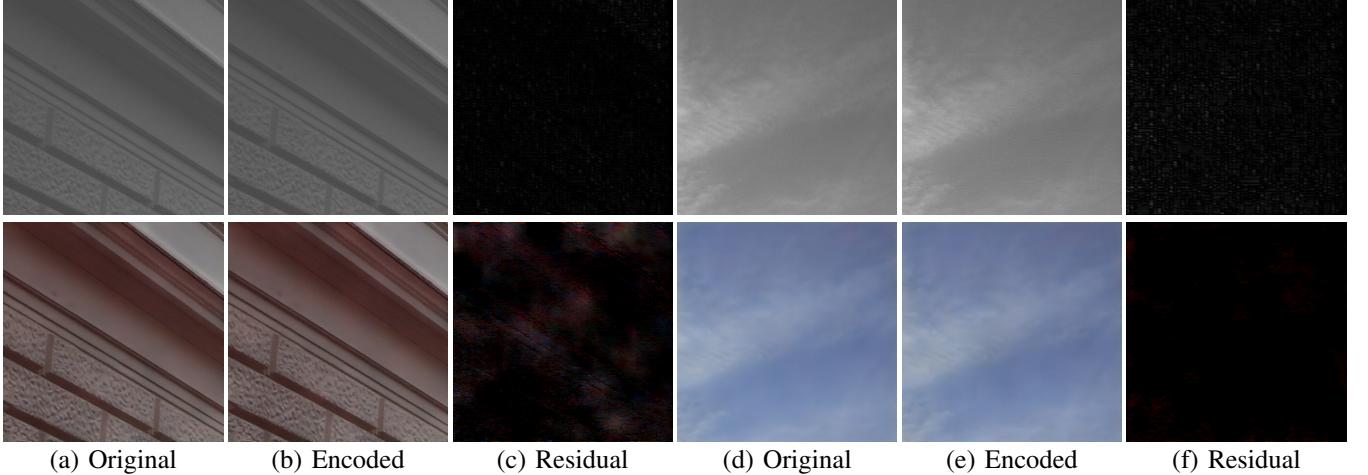


Fig. 3. Examples of encoded and residual images. The first and second columns are RAW and RGB images separately. The residual image is the absolute difference between the encoded image and the original image (To see the difference clearly, we multiply the RAW residuals and the RGB residuals by 10). These examples have 100 bit encoded messages.

concatenated with the RAW image  $\mathbf{R} \in \mathbb{R}^{H \times W \times 1}$ . This pre-process of watermarking message benefits the convergence of encoder and decoder [30]. Considering the properties of bayer pattern in RAW images, we use two U-Net alike networks to compose our encoder, where one is used to process the concatenated tensor  $\mathbf{T}' \otimes \mathbf{R}$  of RAW image and watermarking, where  $\otimes$  denotes the concatenation operation of tensor. The other is used to concatenated tensor  $\mathbf{T}' \otimes \mathbf{R}_d$  of demosaicing RAW image  $\mathbf{R}_d \in \mathbb{R}^{H \times W \times 4}$  and watermarking. Where demosaicing RAW image  $\mathbf{R}_d = U(\mathbf{R}_{re} \otimes \mathbf{R}_b \otimes \mathbf{R}_{gr} \otimes \mathbf{R}_{gb})$ ,  $\mathbf{R}_{re}$ ,  $\mathbf{R}_b$ ,  $\mathbf{R}_{gr}$ ,  $\mathbf{R}_{gb}$  denotes the subimages of red pixels, blue pixels, green pixels in upper left and green pixels in lower right respectively and  $U$  means the upsampling 2 times. Finally, the results from two networks are averaged to get a single channel RAW residual  $\mathbf{R}_r \in \mathbb{R}^{H \times W \times 1}$ , which will add to the original RAW image to get container RAW image  $\mathbf{R}_e \in \mathbb{R}^{H \times W \times 1}$ . We present examples of encoded images and residuals in Fig 3. In subsequent ablation experiments, we also conduct the experiments that only uses RAW images, only demosaiced images and both. The motivation behind the utilization of the two-streamed encoder lies in our aspiration to incorporate supplementary pixel spatial information into the encoder through the implementation of a demosaicing module. It is crucial to emphasize that similar modules have been developed for existing deep ISP pipelines, indicating their effectiveness and relevance. By leveraging these modules, our objective is to enhance the overall performance and capabilities of our framework, resulting in improved image quality and fidelity of the encoded images. The ablation experiment serves as an illustration, demonstrating that the introduction of additional pixel spatial information can indeed enhance image quality and decoding accuracy.

**Decoder:** The function of the decoder is to extract the watermarking from the RGB image produced by the deep ISP pipeline (image will also go through the distortion network if using it in training). We adopt the same decoder structure as Stegastamp [30]. To extract the output message from the RGB images, we resort multiple convolutional layers and

multiple Multilayer Perceptron (MLP), which are designed to extract relevant features and patterns from the image to recover watermarking. In the end, we use a Sigmoid function to map output values to a probability score between 0 and 1. The length of the output message is determined by the number of units in final MLP, which have the same length as the input message. The decoder network is supervised using the cross-entropy loss between copyright watermarking  $\mathbf{M}$  and embeded watermarking  $\mathbf{M}_d$ .

**Discriminator:** Since the adversarial training improves the visual quality of encoded images [29], we introduce a WGAN-style discriminator [41] in our RAWIW to supervise the training of our encoder. The discriminator network comprises a series of convolutional layers followed by a max pooling. It is used to predict whether a watermarking is encoded in the RGB image which is used as a perceptual loss for the whole framework. The discriminator is trained using an input image and its corresponding encoded image, where the Wasserstein loss function is employed as a supervised signal.

### C. Deep ISP Pipeline

RAWIW requires a module to transform RAW images into RGB images. Given the complexity of the traditional ISP pipeline, which includes some non-differentiable operations, we have chosen to utilize the deep ISP pipeline in our investigation. Specifically, we employ MW-ISPNet [2] and AWNet [4], as mentioned above. Additionally, we trained a UNet [3]-style architecture ISP pipeline for conducting experiments. During the training process of the watermarking module, the parameters of the entire deep ISP pipeline are kept fixed. The reason for not modifying the parameters of the ISP pipeline to enhance the concealment of watermarking is that our primary objective is to enable the encoder to learn how to directly encode information into RAW images and the decoder to learn how to decode information from RGB images. Moreover, modifying the ISP pipeline could introduce additional computational complexity, potentially impacting the overall system performance. To compare the impact of

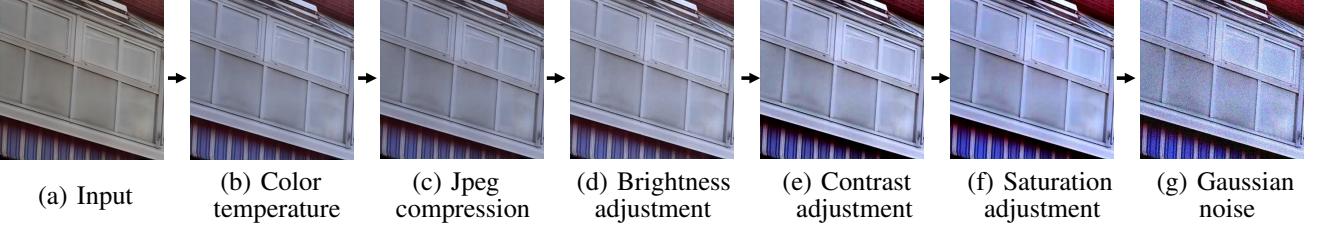


Fig. 4. Image distortion pipeline. During training, in order to make our model robust to distortion during transmission, we insert a distortion network that simulates the distortion between the encoder and decoder. We take the encoder output and apply the random transformations shown here before passing the output through the decoder.

different ISP pipelines on the entire framework, we conducted relevant ablation experiments, and the specific experimental results are presented in Section IV-D.

#### D. Distortion Network

Since the deep ISP pipeline is trained to establish a mapping between RAW images and RGB images, this mapping can be significantly influenced by the ISP pipeline of data collection devices. However, for RAW image copyright protection, the watermarking must remain robust to traditional ISP pipelines as well. To enhance the decoder's robustness to these distortions (we consider the disparity between the deep ISP pipeline and traditional ISP pipelines as distortion), we introduce a distortion network in front of the decoder during training. In our approach, we address the visual gap between RGB images generated by different ISP pipelines, primarily manifested in differences in color temperature and compression. Additionally, to ensure the watermarking's robustness to transmission noises, we incorporate operations such as differentiable color temperature adjustment, JPEG compression [42], Gaussian noise, saturation, contrast, and brightness adjustment into the distortion network. During training, the degrees of these distortions are randomly chosen within a range related to the training epoch, facilitating the decoder's convergence. Fig 4 visually illustrates the image distortion process. The introduction of the distortion network enhances the decoder's capability to handle various distortions, ensuring robustness and maintaining the effectiveness of the watermarking even when faced with the challenges posed by traditional ISP pipelines. The details of the distortions are as follows:

**Color temperature:** In order to adjust the color temperature of images, we use a Kelvin table which comprises of Kelvin values [43] of different color temperatures.

$$(r', g', b') = (r \times \frac{r_t}{255}, g \times \frac{g_t}{255}, b \times \frac{b_t}{255}), \quad (3)$$

where  $r_t$ ,  $g_t$ ,  $b_t$  are the Kelvin values in color temperatures  $t$ . The color temperature  $t$  is randomly selected from:  $[6500 - \frac{5500}{\lambda-\epsilon}, 6500 + \frac{5500}{\lambda-\epsilon}]$ , where  $\epsilon$  is the current epoch number,  $\lambda$  is the total epoch number of third stage. The meaning behind the article remains the same as stated above.

**JPEG compression:** We use the differentiable JPEG compression proposed by [44] to approximate rounding function:

$$\text{rounding}(x) = \begin{cases} x^3 & |x| < 0.5 \\ x & |x| \geq 0.5 \end{cases}. \quad (4)$$

The JPEG quality factor is randomly selected from  $[60 + \frac{40}{\epsilon+1}, 100]$ .

**Brightness, contrast, saturation, and noise:** For the brightness, contrast and saturation, we use the functions in Kornia. The brightness, contrast and saturation factor is randomly selected from  $[0 - \frac{0.3}{\lambda-\epsilon}, 0 + \frac{0.3}{\lambda-\epsilon}]$ ,  $[0 - \frac{0.1}{\lambda-\epsilon}, 0 + \frac{0.1}{\lambda-\epsilon}]$  and  $[0, 0 + \frac{0.1}{\lambda-\epsilon}]$  respectively. At the same time, we use the Gaussian noise and the standard deviation is randomly selected from  $[0, 0 + \frac{0.05}{\lambda-\epsilon}]$ .

#### E. Three-Stage Training and Loss Function

To strike an optimal balance between decoding accuracy, image quality, and robustness to noise, we have designed a three-stage training process. Each stage focuses on optimizing a specific goal: decoding accuracy, image quality, and robustness to noise, respectively. In the first stage, we prioritize achieving successful watermarking decoding, which comes at the cost of noticeable degradation in both RAW and RGB image quality. Moving to the second stage, we observe improvements in image quality, but this might result in increased vulnerability of the watermarking to distortion. In the third stage, we reinforce the resilience of the watermarking by gradually introducing noise augmentation. It is essential to note that our experimental findings demonstrate that attempting to simultaneously optimize all goals from the beginning leads to non-convergence of the loss function. Hence, our step-by-step approach allows us to effectively enhance different aspects of the watermarking process and achieve the desired trade-offs.

**The First Stage:** At this stage, the optimization is solely focused on the RGB image decoder. The loss function for optimizing the decoder is defined as  $L_{S1}$ :

$$L_{S1}(M, M_d) = L_{dec}(M, M_d) = \text{Cross-Entropy}(M, M_d). \quad (5)$$

**The Second Stage:** After the initial stage, our model can retrieve the hidden watermarking message from the encoded RGB images generated by the ISP pipeline. However, it should be noted that the appearance of the encoded RGB images closely resembles Quickly Respond (QR) code images. The observed phenomenon can be attributed to the absence of the visual quality constraints imposed on the model. Therefore, in

this stage, we use  $L_{S2}$  to the second stage as follow:

$$\begin{aligned} L_{S2}(M, M_d, R, R_e, I_o, I_e) &= \lambda_1 L_{dec}(M, M_d) + \lambda_2 L_2(R, R_e) \\ &+ \lambda_3 L_2(I_o, I_e) + \lambda_4 L_P(I_o, I_e) + \lambda_5 L_d(I_o, I_e), \end{aligned} \quad (6)$$

where  $L_2$  is L2 norm loss,  $L_P$  is the LPIPS [45] loss and  $L_d$  is the discriminator loss.  $\lambda_1, \lambda_2, \lambda_3, \lambda_4$  and  $\lambda_5$  are hyperparameter set to 2, 1, 1, 1, and 1, respectively.

**The Third Stage:** Following the aforementioned two stages of training, the model has acquired the ability to encode the information into RAW images and extract the information from RGB images, simultaneously preserving the visual fidelity of both RAW and RGB images. However, the current watermarking is not robust to different distortions (*e.g.*, Gaussian Noise, JPEG compression). To effectively extract the embedded watermarking from the distorted RGB images. We train our encoder and decoder using  $L_{S3}$  as follows:

$$\begin{aligned} L_{S3}(M, M'_d, R, R_e, I_o, I_e) &= \lambda_1 L_{dec}(M, M'_d) + \lambda_2 L_2(R, R_e) \\ &+ \lambda_3 L_2(I_o, I_e) + \lambda_4 L_P(I_o, I_e) + \lambda_5 L_d(I_o, I_e). \end{aligned} \quad (7)$$

The  $L_{S3}$  is very similar to the  $L_{S2}$ , only the second parameter is different, where  $M'_d$  means decoded binary message from distorted RGB images and the hyperparameters setting is same as second stage.

$$M'_d = D_{\theta_D}(\mathbf{N}(I_e)), \quad (8)$$

where  $\mathbf{N}$  is present the distortion network, which can help the encoder to add anti-distortion watermarking into RAW images.

#### IV. EXPERIMENTS

Within this section, we give the details about our experimental settings and conduct extensive experiments.

##### A. Experimental Setting

**Dataset:** In our experiments, we primarily utilized two datasets: the ZRR dataset [37] and the SR-RAW dataset [46]. The ZRR dataset served as the training dataset, comprising 47,863 paired images. This dataset consists of RAW images of size  $448 \times 448 \times 1$ , captured by Huawei p20, and RGB images of size  $448 \times 448 \times 3$ , captured and processed by the Canon 5D Mark IV camera. The test set of the ZRR dataset was used to conduct various experiments, including ablation and robustness testing experiments.

However, since the RAW images in the aforementioned dataset are stored in PNG format, they lack the camera parameters saved during shooting. Consequently, traditional ISP pipelines cannot directly convert them to RGB format. To overcome this limitation, we conducted traditional ISP test experiments using the test set of the SR-RAW dataset. The test set collected 50 scenes of images, with the authors of dataset taking 7 photos at different focal lengths (24, 35, 50, 70, 100, 150, and 240 mm) in each scene. We randomly selected 100 of these images for the traditional ISP test experiment.

**Hyper-Parameter Setting and Implementation Details:** As highlighted in Section III-E, the training phase of our

TABLE I  
TRADITIONAL ISP PIPELINE TEST RESULTS. FOR THE ISP (TRADITIONAL ISP PIPELINES), KEY: R-A: AUTO WHITE BALANCE; R-C: CAMERA WHITE BALANCE; R-D: DAYLIGHT WHITE BALANCE. FOR THE ISPT (DEEP ISP PIPELINES IN TRAINING OF MODELS). FOR THE DISTORTION, ✓ INDICATING THE DISTORTION NETWORK IS USED IN TRAINING OF MODEL AND × INDICATING NOT.

ISP	ISPT	Distortion	PSNR	SSIM	BER	SER
R-A	AWNet	×	32.653	0.811	3.73%	8.33%
	AWNet	✓	30.794	0.857	2.81%	0.02%
	MWNet	×	27.047	0.689	5.63%	0.27%
	MWNet	✓	27.884	0.691	3.10%	0.00%
	UNet	×	31.736	0.891	3.85%	14.58%
	UNet	✓	31.477	0.904	2.50%	0.00%
R-C	AWNet	×	32.704	0.812	3.67%	10.42%
	AWNet	✓	30.866	0.859	3.08%	4.17%
	MWNet	×	27.110	0.698	5.60%	0.25%
	MWNet	✓	28.060	0.693	2.77%	0.00%
	UNet	×	31.474	0.905	3.71%	12.50%
	UNet	✓	31.870	0.892	2.42%	0.00%
R-D	AWNet	×	32.939	0.820	3.40%	4.17%
	AWNet	✓	30.954	0.860	2.92%	2.08%
	MWNet	×	27.222	0.679	5.42%	16.67%
	MWNet	✓	28.348	0.707	2.77%	4.17%
	UNet	×	31.727	0.907	2.56%	0.00%
	UNet	✓	31.947	0.892	4.25%	0.15%

model is segregated into three stages. Specifically, we conducted 5, 5 and 10 epochs for the first stage, second stage, and the third stage respectively.

Two optimizers were used in the model training process. The Adam optimizer was assigned to the task of optimizing the RAW image encoder and the RGB image decoder. The RMSProp was to optimize the discriminator. The learning rate of both optimizers is  $5 \times 10^{-5}$ . The batch size is set to 4.

For our watermarking, we elected to generate the 100-bit binary strings randomly during the training phase. In the testing phase, we restricted the embedded watermarking to 56 bits, subsequently incorporating error-correcting codes using Bose–Chaudhuri–Hocquenghem (BCH) codes [47] to fill the remaining 44 bits, ultimately arriving at a 100-bit watermarking.

It is important to note that we have made certain adjustments to the number of submodules proposed by MW-ISPNet and AWNet in comparison to their respective public models. This reduction in the number of submodules has proven to be beneficial for the convergence of the entire framework, resulting in faster training. We used publicly available code for training purposes but made specific modifications to the models, such as adjusting the number of modules and channel features. Specifically, for AWNet, we set the number of GCRDB submodules to [1, 1, 1, 2, 3]. As for MWNet, we configured the feature channels to [32, 64, 64]. On the other hand, the UNet ISP employs a similar framework to the encoder, and the training process closely resembles that of AWNet.

**Evaluation Metrics:** In the subsequent experiments, we assessed the efficacy of our model in terms of its concealment and robustness capabilities. In regards to concealment, three metrics were employed, namely Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSIM) [48], and Learned Per-

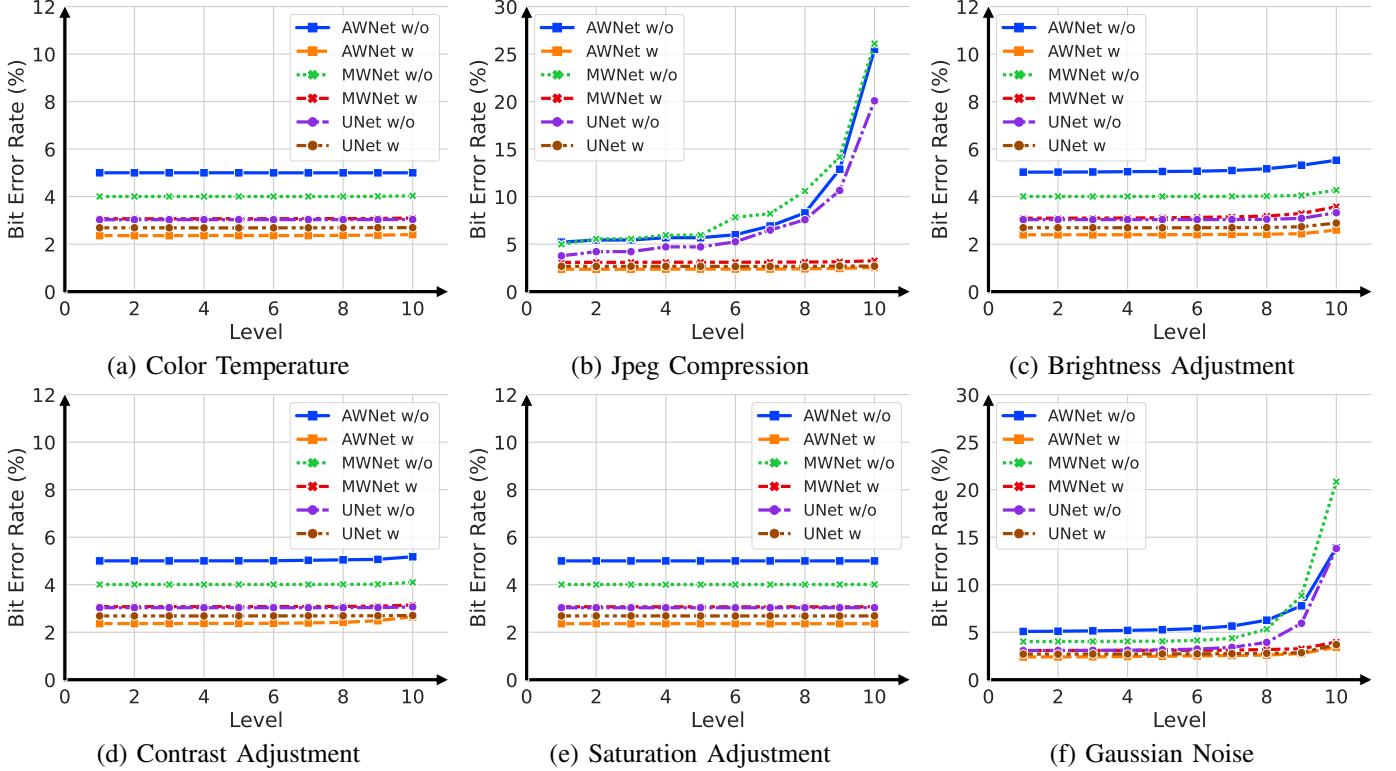


Fig. 5. Bit Error Rate (BER) of the decoding results of different models (The combined encoder is used in the training of the above models.) under diverse levels of various distortions. MW-ISPNET is abbreviated as MWNet.

ceptual Image Patch Similarity (LPIPS) [45]. It is noteworthy that, in this paper, version 0.1 of LPIPS trained on the AlexNet network was utilized.

On the other hand, the evaluation of robustness involved the use of two metrics: Bit Error Rate (BER), and String Error Rate (SER):

$$BER = \frac{E_{bit}}{A_{bit}}, SER = \frac{E_{str}}{A_{str}}, \quad (9)$$

in which  $E_{bit}$  represents the number of bits in error, and  $A_{bit}$  represents the total number of bits to be hidden. The meaning of  $E_{str}$  and  $A_{str}$  is similar. It is noteworthy that, in our experiment, the string was subjected to BCH coding. Consequently, even if the number of erroneous bits is small, the string can still be accurately restored.

### B. Robustness Test

In this section, we test the robustness of our model to different distortions. We test the distortions used in training. For each distortion category, we set 10 different distortion levels (The larger the number, the greater the distortion). 1) Color temperature adjustment: color temperature is uniform sampling in  $T \sim U[6500 - \frac{5500}{10-\varepsilon}, 6500 + \frac{5500}{10-\varepsilon}]$ ; 2) JPEG Compression: JPEG quality is uniform sampling in  $J \sim U[60 + \frac{40}{\varepsilon+1}, 100]$ ; 3) Brightness and contrast adjustment: affine histogram rescaling  $mx + b$  with  $m \sim U[0 - \frac{0.1}{10-\varepsilon}, 0 + \frac{0.1}{10-\varepsilon}]$  and  $b \sim U[0 - \frac{0.3}{10-\varepsilon}, 0 + \frac{0.3}{10-\varepsilon}]$ ; 4) Noise and Saturation adjustment: we use a Gaussian noise (sampling the standard deviation  $\sigma \sim U[0, 0 + \frac{0.05}{10-\varepsilon}]$ ) and saturation factor is uniform sampling in  $S \sim U[0, 0 + \frac{0.1}{10-\varepsilon}]$ . where  $\varepsilon$  is the distortion

level. The specific results of the experiment are shown in Fig 5.

### C. Traditional ISP Pipeline Test

To assess the robustness of the proposed method against the traditional ISP pipeline, a validation experiment was conducted. To achieve the necessary processing from RAW to RGB images, In the experiment, the traditional ISP pipeline we selected is Rawpy library. The Rawpy library, built on the foundation of LibRaw, serves as the underlying implementation for our work. This library offers a range of image processing operations that enable the conversion of RAW images into RGB images. These operations include essential tasks like white balance correction, color space conversion, and brightness adjustment. Given the capabilities and functionality of rawpy, we consider it to effectively represent traditional ISP methodologies. The Rawpy library has three white balance models: R-A (it Automatically calculates the white balance), R-C (it uses the as-shot white balance values of Camera), and R-D (it uses Daylight white balance correction). The encoding and decoding processes of the proposed method remains unchanged throughout the experiment. One hundred unique images were randomly selected from the SR-RAW dataset, with the condition that these images were disjoint from the training set used in the proposed method. The random 56-bit messages, after being subjected to using BCH coding to 100-bit messages, were embedded within each of these selected images. The results of this experiment are presented in Table I. A total of six models were subjected to testing, which are

TABLE II

ABLATION STUDY RESULTS WITH DISTORTION NETWORK. ‘R’, ‘D’ AND ‘C’ STAND FOR RAW ENCODER, DEMOSAICING ENCODER AND COMBINED ENCODER RESPECTIVELY. BEST IN **RED** AND SECOND IN **BLUE**.

ISP	Encoder	PSNR↑			LPIPS↓		SSIM↑		BER↓	SER↓
		RAW	GT	ISP	GT	ISP	GT	ISP		
MW-ISPNet [2]	R	45.389	20.164	28.979	0.154	0.068	0.804	0.975	<b>1.035%</b>	<b>0.083%</b>
	D	37.060	20.487	31.078	0.134	0.051	0.804	0.974	3.006%	<b>0.083%</b>
	C	37.890	20.144	29.169	<b>0.113</b>	<b>0.020</b>	0.804	0.981	1.155%	1.080%
Unet [3]	R	<b>47.581</b>	19.088	32.242	0.174	0.040	0.792	<b>0.988</b>	1.088%	0.332%
	D	44.858	18.486	30.749	0.180	0.038	0.787	0.986	4.002%	0.332%
	C	44.859	19.637	31.693	0.154	0.036	0.797	0.987	2.029%	0.166%
AWNet [4]	R	44.347	20.507	31.407	<b>0.116</b>	<b>0.024</b>	0.812	0.978	2.014%	0.166%
	D	<b>46.399</b>	<b>20.782</b>	<b>32.751</b>	0.143	0.056	<b>0.816</b>	0.984	2.086%	0.498%
	C	40.277	<b>21.046</b>	<b>35.143</b>	0.144	0.057	<b>0.819</b>	<b>0.989</b>	<b>0.880%</b>	<b>0.000%</b>

TABLE III

ABLATION STUDY RESULTS WITHOUT DISTORTION NETWORK. ‘R’, ‘D’ AND ‘C’ STAND FOR RAW ENCODER, DEMOSAICING ENCODER AND COMBINED ENCODER RESPECTIVELY. BEST IN **RED** AND SECOND IN **BLUE**.

ISP	Encoder	PSNR↑			LPIPS↓		SSIM↑		BER↓	SER↓
		RAW	GT	ISP	GT	ISP	GT	ISP		
MW-ISPNet [2]	R	45.240	20.737	36.762	0.111	0.012	0.814	0.983	4.071%	1.495%
	D	43.531	20.728	39.480	0.109	0.008	0.814	0.984	2.354%	2.658%
	C	47.623	20.871	45.018	<b>0.106</b>	<b>0.001</b>	<b>0.816</b>	<b>0.989</b>	<b>2.179%</b>	0.831%
Unet [3]	R	<b>50.110</b>	19.289	38.834	0.163	0.006	0.798	<b>0.988</b>	3.034%	0.664%
	D	48.806	19.449	40.052	0.156	0.006	0.799	0.987	<b>2.252%</b>	1.495%
	C	44.292	19.402	40.814	0.160	0.004	0.798	<b>0.988</b>	3.686%	<b>0.166%</b>
AWNet [4]	R	<b>50.966</b>	<b>21.316</b>	46.145	<b>0.108</b>	<b>0.002</b>	<b>0.826</b>	<b>0.989</b>	3.100%	0.748%
	D	49.489	21.281	<b>46.294</b>	0.109	0.003	<b>0.826</b>	<b>0.988</b>	4.041%	<b>0.498%</b>
	C	46.181	<b>21.311</b>	<b>47.303</b>	<b>0.108</b>	<b>0.002</b>	<b>0.826</b>	<b>0.988</b>	3.359%	2.409%

TABLE IV

COMPARISON RESULTS WITH STATE-OF-THE-ART RGB IMAGE WATERMARKING METHODS. BEST IN **RED** AND SECOND IN **BLUE**.

model	PSNR ↑	SSIM ↑	BER ↓	SER ↓
Stegastamp [30]	29.31	0.927	<b>0.230%</b>	<b>0.000%</b>
RIHOOP [31]	29.42	0.940	<b>0.120%</b>	<b>0.000%</b>
RAWIW(M)	29.169	0.981	1.155%	1.080%
RAWIW(U)	<b>31.693</b>	<b>0.987</b>	2.029%	<b>0.166%</b>
RAWIW(A)	<b>35.143</b>	<b>0.989</b>	0.880%	<b>0.000%</b>

all using combined encoder. Examples of watermarked images generated by different models in this experiment are shown in Fig 6. The experimental results demonstrate the robustness of proposed RAWIW framework against the traditional ISP pipelines, and using the distortion network in training can improve the decoding accuracy of the watermarking for the traditional ISP pipeline.

#### D. Ablation Study

In this section, we conduct ablation experiments to compare watermarking concealment and robustness metrics under different ISP pipeline, encoders and distortions. We use the test data in the ZRR dataset for testing, a total of 1204 paired images. Throughout the ablation experiment, we trained a total of 18 ( $3 \times 3 \times 2$ ) models, using three ISP networks,

UNet, MW-ISPNet, and AWNet, three encoder structures, and whether to use a distortion network. Our model can be trained to store different numbers of bits. We settle on a message length of 100 bits in our experiment as it provides a good compromise between image quality and information transfer. The watermarking is random 100 bits binary string in training and 100 bits binary string encoded by random 56 bits binary through BCH coding. We can recover watermarking in the case of a few bit errors by using BCH coding. The experimental results in Table II and III show that: 1) The ISP network adopts the AWNet structure to make the watermarking concealment better. 2) The combined encoder structure is better than the RAW structure and demosaicing structure. 3) The distortion network will reduce the concealment and improve the robustness of watermarking. Distortion networks enable encoders to add more “obvious” watermarking to RAW images against distortion caused by distortion networks.

#### E. Comparison with SOTA

In this section, we have conducted a comparative experiment between two state-of-the-art RGB image watermarking methods: Stegastamp and RIHOOP. To ensure consistency and accurate evaluation, the experiment settings were derived from the ablation experiment IV-D, and the RGB image watermarking methods encoded watermarking into RGB images generated by ISP. The corresponding results are presented in Table IV. It is evident that both our RAW image watermarking

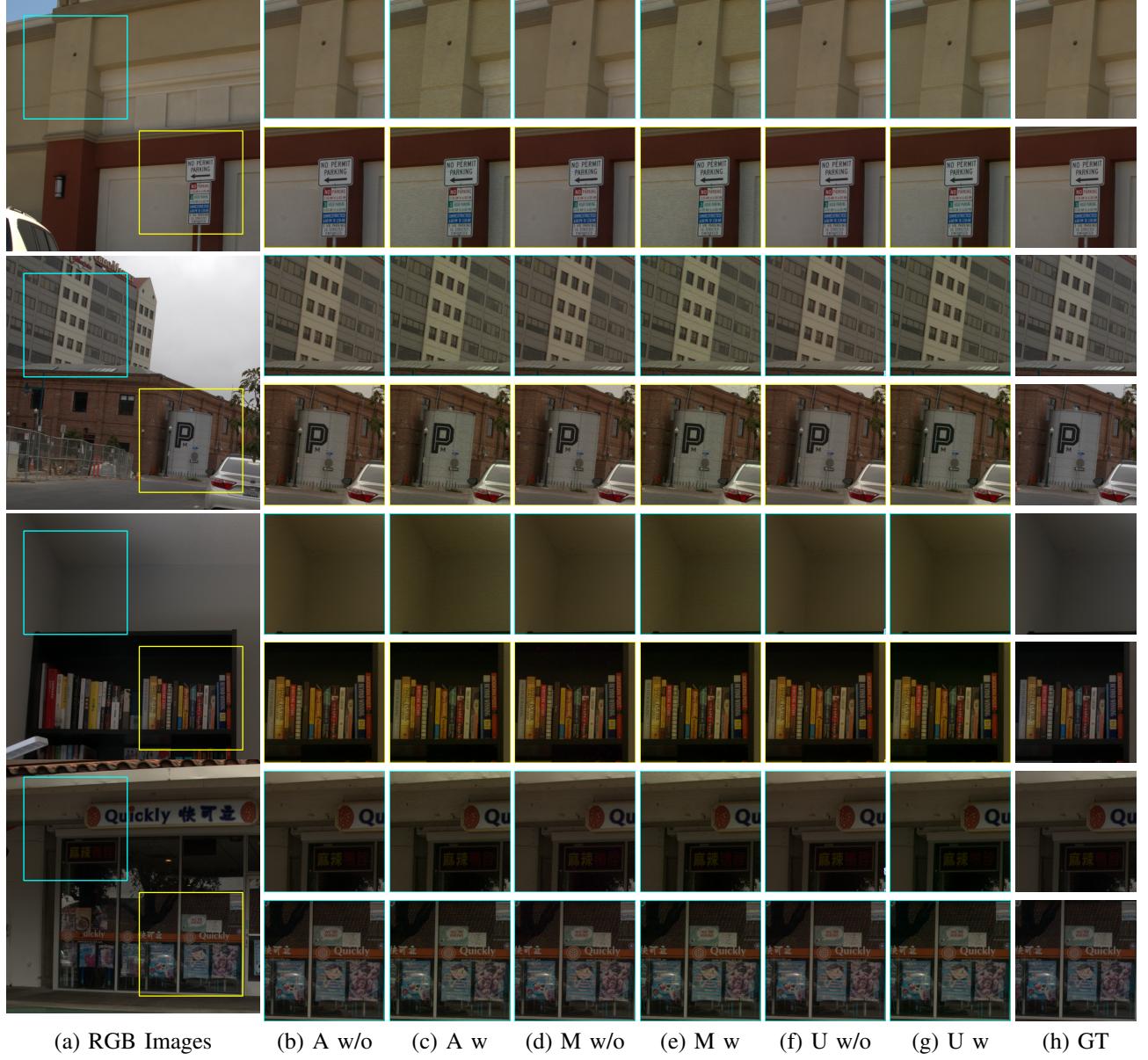


Fig. 6. Visual results on the SR-RAW test set. Training with distortion network will reduce the visual quality of the resulting RGB images but improve the robustness of watermarking. The combined encoder is used in the training of the above models. AWNet, MW-ISPNET, and UNet are abbreviated as A, M, and U.

and RGB watermarking methods can maintain good visual quality while preserving high decoding accuracy. However, it is worth noting that our RAW image watermarking uses RAW images as covers, which have one-third the capacity of RGB images. As a result, the information hiding density of our method is higher compared to the RGB watermarking methods.

## V. LIMITATIONS

Since our method uses the deep ISP pipeline, the training speed will slower than RGB watermarking methods. This is because the gradient of the entire ISP pipeline needs to be calculated. In the future, using a smaller deep ISP pipeline can improve the training speed. At present, our method is limited to encoding and decoding RAW image pixel blocks of a fixed

size. However, due to the considerable size of the RAW image in practical applications, we have opted to encode only the central area pixel block. If we choose other pixel blocks to add watermarking, we need to manually locate the watermarking pixel block from the RGB image. At the same time, since we use pixel blocks to add watermarking, the whole RGB results will have an obvious boundary around watermarked blocks.

## VI. CONCLUSION

This paper introduces an innovative end-to-end framework, referred to as **RAWIW**, with the primary goal of providing copyright protection for RAW images. The framework consists of a watermarking module, a deep ISP module, and a distortion network. Additionally, we employ an effective three-stage training strategy to strike a balance between the

robustness and concealment of watermarking. The proposed method is validated on two RAW image datasets, namely ZRR dataset and SR-Raw dataset. The results demonstrate that the RAWIW framework exhibits robustness to different ISP pipelines and distortions during transmission, all while minimizing the impact on the visual quality of the resulting RGB images. As for future work, we plan to delve further into investigating strategies to enhance the robustness and concealment capabilities of the watermarking process. By continuously refining the framework, we aim to bolster its performance and broaden its applicability in various scenarios related to RAW image copyright protection.

## REFERENCES

- [1] P. Meerwald and A. Uhl, "Watermarking of raw digital images in camera firmware: embedding and detection," in *Advances in Image and Video Technology: Third Pacific Rim Symposium, PSIVT 2009, Tokyo, Japan, January 13–16, 2009. Proceedings 3*. Springer, 2009, pp. 340–348.
- [2] A. Ignatov, R. Timofte, Z. Zhang, M. Liu, H. Wang, W. Zuo, J. Zhang, R. Zhang, Z. Peng, S. Ren et al., "Aim 2020 challenge on learned image signal processing pipeline," in *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 152–170.
- [3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.
- [4] L. Dai, X. Liu, C. Li, and J. Chen, "Awnet: Attentive wavelet network for image isp," *arXiv preprint arXiv:2008.09228*, 2020.
- [5] L. M. Marvel, C. G. Boncelet, and C. T. Retter, "Spread spectrum image steganography," *IEEE Transactions on image processing*, vol. 8, no. 8, pp. 1075–1083, 1999.
- [6] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer*, vol. 31, no. 2, pp. 26–34, 1998.
- [7] J. Wang, J. Ni, X. Zhang, and Y.-Q. Shi, "Rate and distortion optimization for reversible data hiding using multiple histogram shifting," *IEEE transactions on cybernetics*, vol. 47, no. 2, pp. 315–326, 2016.
- [8] W. Lu, J. Zhang, X. Zhao, W. Zhang, and J. Huang, "Secure robust jpeg steganography based on autoencoder with adaptive bch encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 7, pp. 2909–2922, 2020.
- [9] W. Li, K. Chen, W. Zhang, H. Zhou, Y. Wang, and N. Yu, "Jpeg steganography with estimated side-information," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2288–2294, 2019.
- [10] W. Li, W. Zhou, W. Zhang, C. Qin, H. Hu, and N. Yu, "Shortening the cover for fast jpeg steganography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1745–1757, 2019.
- [11] W. Zhang, Z. Zhang, L. Zhang, H. Li, and N. Yu, "Decomposing joint distortion for adaptive steganography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 10, pp. 2274–2280, 2016.
- [12] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, "Ssgan: Secure steganography based on generative adversarial networks," in *Advances in Multimedia Information Processing–PCM 2017: 18th Pacific-Rim Conference on Multimedia, Harbin, China, September 28–29, 2017, Revised Selected Papers, Part I 18*. Springer, 2018, pp. 534–544.
- [13] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1547–1551, 2017.
- [14] X. Guo, X. Liu, Z. Ren, S. Grosz, I. Masi, and X. Liu, "Hierarchical fine-grained image forgery detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3155–3165.
- [15] X. Liu, Y. Liu, J. Chen, and X. Liu, "Pscce-net: Progressive spatio-channel correlation network for image manipulation detection and localization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7505–7517, 2022.
- [16] I. G. Karybali and K. Berberidis, "Efficient spatial image watermarking via new perceptual masking and blind detection schemes," *IEEE Transactions on Information Forensics and security*, vol. 1, no. 2, pp. 256–274, 2006.
- [17] S. Pereira, S. Voloshynoskiy, and T. Pun, "Optimal transform domain watermark embedding via linear programming," *Signal processing*, vol. 81, no. 6, pp. 1251–1260, 2001.
- [18] H. S. Kim and H.-K. Lee, "Invariant image watermark using zernike moments," *IEEE transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 766–775, 2003.
- [19] Y. Yang, T. Zou, G. Huang, and W. Zhang, "A high visual quality color image reversible data hiding scheme based on brg embedding principle and ciede2000 assessment metric," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 1860–1874, 2021.
- [20] H. Tian, Y. Zhao, R. Ni, L. Qin, and X. Li, "Ldft-based watermarking resilient to local desynchronization attacks," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2190–2201, 2013.
- [21] K. A. Birney and T. R. Fischer, "On the modeling of dct and subband image data for compression," *IEEE transactions on Image Processing*, vol. 4, no. 2, pp. 186–193, 1995.
- [22] J. R. Hernandez, M. Amado, and F. Perez-Gonzalez, "Dct-domain watermarking techniques for still images: Detector performance analysis and a new structure," *IEEE transactions on image processing*, vol. 9, no. 1, pp. 55–68, 2000.
- [23] Q. Cheng and T. S. Huang, "Robust optimum detection of transform domain multiplicative watermarks," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 906–924, 2003.
- [24] M. Barni, F. Bartolini, A. De Rosa, and A. Piva, "A new decoder for the optimum recovery of nonadditive watermarks," *IEEE transactions on image processing*, vol. 10, no. 5, pp. 755–766, 2001.
- [25] D. Zheng, J. Zhao, and A. El Saddik, "Rst-invariant digital image watermarking based on log-polar mapping and phase correlation," *IEEE transactions on circuits and systems for video technology*, vol. 13, no. 8, pp. 753–765, 2003.
- [26] S. Xiang, H. J. Kim, and J. Huang, "Invariant image watermarking based on statistical features in the low-frequency domain," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 6, pp. 777–790, 2008.
- [27] X. Wang, Y. Lin, Y. Shen, and P. Niu, "Udtewt-phfms domain statistical image watermarking using vector bw-type r distribution," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
- [28] Y. Huang, H. Guan, J. Liu, S. Zhang, B. Niu, and G. Zhang, "Robust texture-aware local adaptive image watermarking with perceptual guarantee," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
- [29] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [30] M. Tancik, B. Mildenhall, and R. Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2117–2126.
- [31] J. Jia, Z. Gao, K. Chen, M. Hu, X. Min, G. Zhai, and X. Yang, "Rihoop: robust invisible hyperlinks in offline and online photographs," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 7094–7106, 2020.
- [32] M. Afifi, J. T. Barron, C. LeGendre, Y.-T. Tsai, and F. Bleibel, "Cross-camera convolutional color constancy," in *The IEEE International Conference on Computer Vision (ICCV)*, 2021.
- [33] J. Cai, S. Gu, and L. Zhang, "Learning a deep single image contrast enhancer from multi-exposure images," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 2049–2062, 2018.
- [34] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Trans. Graph.*, vol. 35, no. 6, dec 2016. [Online]. Available: <https://doi.org/10.1145/2980179.2982399>
- [35] E. Schwartz, R. Giryes, and A. M. Bronstein, "Deepisp: Toward learning an end-to-end image processing pipeline," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 912–923, 2019.
- [36] Z. Liang, J. Cai, Z. Cao, and L. Zhang, "Cameranet: A two-stage framework for effective camera isp learning," *IEEE Transactions on Image Processing*, vol. 30, pp. 2248–2262, 2021.
- [37] A. Ignatov, L. Van Gool, and R. Timofte, "Replacing mobile camera isp with a single deep learning model," *arXiv preprint arXiv:2002.05509*, 2020.
- [38] A. Ignatov, R. Timofte, S.-J. Ko, S.-W. Kim, K.-H. Uhm, S.-W. Ji, S.-J. Cho, J.-P. Hong, and et.al., "Aim 2019 challenge on raw to rgb mapping: Methods and results," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 3584–3590.
- [39] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-cnn for image restoration," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2018, pp. 886–88609. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/CVPRW.2018.00121>

- [40] Z. Zhang, H. Wang, M. Liu, R. Wang, W. Zuo, and J. Zhang, “Learning raw-to-srgb mappings with inaccurately aligned supervision,” in *The IEEE International Conference on Computer Vision (ICCV)*, 2021.
- [41] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *International conference on machine learning*. PMLR, 2017, pp. 214–223.
- [42] C. Zhang, A. Karjauv, P. Benz, and I. S. Kweon, “Towards robust data hiding against (jpeg) compression: A pseudo-differentiable deep learning approach,” *arXiv preprint arXiv:2101.00973*, 2020.
- [43] J. Hernandez-Andres, R. L. Lee, and J. Romero, “Calculating correlated color temperatures across the entire gamut of daylight and skylight chromaticities,” *Applied optics*, vol. 38, no. 27, pp. 5703–5709, 1999.
- [44] R. Shin and D. Song, “Jpeg-resistant adversarial images,” in *NIPS 2017 Workshop on Machine Learning and Computer Security*, vol. 1, 2017, p. 8.
- [45] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [46] X. Zhang, Q. Chen, R. Ng, and V. Koltun, “Zoom to learn, learn to zoom,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3762–3770.
- [47] R. C. Bose and D. K. Ray-Chaudhuri, “On a class of error correcting binary group codes,” *Information and control*, vol. 3, no. 1, pp. 68–79, 1960.
- [48] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.