DRAW: Defending Camera-shooted RAW against Image Manipulation

Xiaoxiao Hu[1,2*], Qichao Ying[1,2*], Zhenxing Qian[1,2†], Sheng Li[1,2], Xinpeng Zhang[1,2]
[1]School of Computer Science, Fudan University
[2]Key Laboratory of Culture & Tourism Intelligent Computing, Fudan University

Abstract

RAW files are the initial measurement of scene radiance widely used in most cameras, and the ubiquitously-used RGB images are converted from RAW data through Image Signal Processing (ISP) pipelines. Nowadays, digital images are risky of being nefariously manipulated. Inspired by the fact that innate immunity is the first line of body defense, we propose DRAW, a novel scheme of defending images against manipulation by protecting their sources, i.e., camera-shooted RAWs. Specifically, we design a lightweight Multi-frequency Partial Fusion Network (MPF-Net) friendly to devices with limited computing resources by frequency learning and partial feature fusion. It introduces invisible watermarks as protective signal into the RAW data. The protection capability can not only be transferred into the rendered RGB images regardless of the applied ISP pipeline, but also is resilient to post-processing operations such as blurring or compression. Once the image is manipulated, we can accurately identify the forged areas with a localization network. Extensive experiments on several famous RAW datasets, e.g., RAISE, FiveK and SIDD, indicate the effectiveness of our method. We hope that this technique can be used in future cameras as an option for image protection, which could effectively restrict image manipulation at the source.

1. Introduction

In the digital world, the credibility of the famous saying “seeing is believing" is largely at risk since nowadays people can easily manipulate critical content within an image and redistribute the fabricated version via the Internet. Owing to the fact that readers are more susceptible to well-crafted misleading material, fabricated images can be a means for some politicians to sway public opinion. In more severe cases, those fraudulent images can be used to bolster fake news or criminal investigation.

*Xiaoxiao Hu and Qichao Ying contribute equally to this work.
†Corresponding author: Zhenxing Qian (zxqian@fudan.edu.cn)

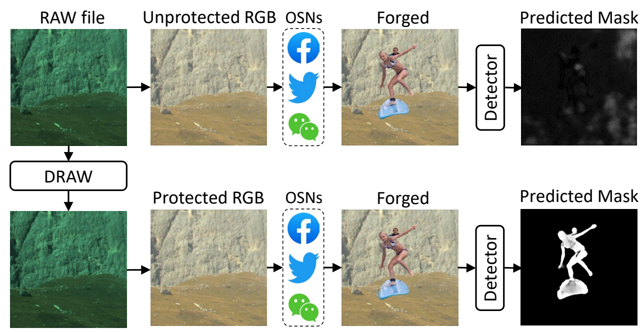


Figure 1. DRAW improves the performance of image manipulation localization against lossy image operations via imperceptible protective signal injection into RAW files.

Image manipulation detection [9, 32] and localization [10, 38] has become a critical area of research for decades, with the goal of distinguishing manipulated images from authentic ones and locating the manipulated areas. While early methods mainly check the integrity of the images from statistical aspects, e.g., the Photo-Response Non-Uniformity (PRNU) noise [9] and the fixed pattern noise (FPN) [23], the uprising of deep networks has greatly strengthened the capability to find traces left by a variety of manipulation [10, 40, 19]. However, the adversary is also continuously evolving both in strength and diversity. For example, recent deep-network-based image editing algorithms [36, 13] are reported to produce highly realistic images with almost no visible artifacts near the edges. Therefore, it remains a big issue whether the learned subtle forensics traces can always be present in the newly forged images. Also, though some works [38, 39] explicitly handle lossy online transmission scenarios, they still face limited performance against well-crafted forgeries, e.g., inpainting, or lossy image operations, e.g., Gaussian blurring.

Inspired by the fact that innate immunity is the first line of body defense and the best weapon to mitigate diseases, safeguarding images against manipulations is an alternative and promising way of deterring malicious attackers. Indeed, the ubiquitous 8-bit RGB images are not the pristine format for reflecting how we perceive the world. They are converted from RAW files via ISP pipelines. Therefore, we propose DRAW, a proactive image protection scheme that defends camera-shooted RAW data against malicious manipulation on the RGB domain. Specifically, we propose to introduce imperceptible protective signal into the RAW data, which can be transferred into the rendered RGB images, even though various types of ISP pipelines are applied. Once these images are manipulated, the localization networks can exactly localize the forged areas regardless of image post-processing operations such as blurring, compression or color jittering. Besides, a novel Multi-frequency Partial Fusion Network (MPF-Net) is proposed to implement RAW protection, which adopts frequency learning and cross-frequency partial feature fusion to significantly decrease the computational complexity. We illustrate the functionality of DRAW in Fig. 1, which promotes accurate manipulation localization without affecting the visual quality.

Extensive experiments on several famous RAW datasets, e.g., RAISE, FiveK and SIDD, prove the imperceptibility, robustness and generalizability of our method. Besides, to compare RAW-domain protection with previous works, we tempt to borrow the success of RGB-domain protection [4, 48] as the baseline method for proactive manipulation localization. The results show that DRAW hosts a noticeable performance gain and a nontrivial benefit of content-related adaptive embedding. In addition, MPF-Net provides superior performance compared to classical U-Net [33] architecture with only 20.9% of its memory cost and 0.95% of its parameters. The novel lightweight architecture makes it possible to be integrated into cameras in the future, thereby changing the current situation where digital images can be freely manipulated.

The contributions of this paper are three-folded, namely:

1. DRAW is the first to propose RAW protection against image manipulation. The corresponding RGB images will carry imperceptible protective signal even though various types of imaging pipelines or lossy image operations are applied.

2. With RAW protection, image manipulation localization networks can better resist lossy image operations such as JPEG compression, blurring and rescaling.

3. A novel lightweight MPF-Net is proposed for integrating RAW protection into cameras in the future, thereby potentially changing the current situation where digital images can be freely manipulated.

2. Related Works

Passive Image Manipulation Localization. Many existing image forensics schemes are designed to detect special kinds of attacks, e.g., splicing detection [38, 34], copy-moving detection [20, 26] and inpainting detection [51, 25]. In addition, some universal tampering detection schemes [10, 40, 19] exploit universal noise artifacts left by manipulation. Mantra-Net [40] uses fully convolutional networks, Z-Pooling and long short-term memory cells for pixel-wise anomaly detection. MVSS-Net [10] jointly exploits the noise view and the boundary artifact using multi-view feature learning and multi-scale supervision. SPAN [19] models the relationship between image patches at multiple scales by constructing a pyramid of local self-attention blocks. RGB-N [49] additionally utilizes auto-generated data augmentation for training. RIML [38] includes adversarial training, where the lossy Online Social Network (OSN) transmission is simulated by modeling noise from different sources. However, these passive schemes are still limited in generalization to well-crafted manipulations or heavy lossy operations.

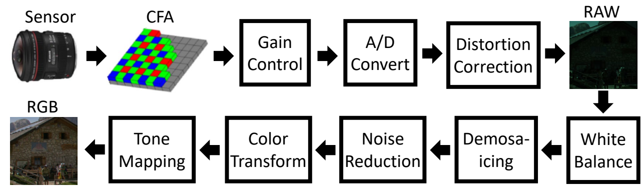


Figure 2. Typical camera imaging pipeline for RAW data acquisition and subsequent RGB image signal processing.

Watermarking for Image Protection. Many image protection schemes based on watermarking [29, 21, 14, 43] have been proposed. Asnani et al. [4] propose to embed templates into images for more accurate manipulation detection. Zhao et al. [48] embed watermarks as anti-Deepfake labels into the facial identity features. FakeTagger [37] embeds the identity information into the whole facial image, which can be recovered after illegal face swapping. Khachaturov et al. [22] and Yin et al [42] respectively propose to attack inpainting or Super-Resolution (SR) models by forcing them to work abnormally on the targeted images. However, these approaches do not tackle the issue of forgery localization, and many of them cannot combat lossy image operations. We alternatively introduce imperceptible protective signal into RAW data and transfer it into RGB images to aid robust manipulation localization.

Models for Limited Computing Resources. Classical network architectures for segmentation-based tasks, e.g., U-Net [33] or FPN [27], usually require non-affordable computing resources for many small devices. MobileNet [17] and ShuffleNet [30] are early works on addressing this issue respectively via Depth-wise Separable Convolution (DSConv) and channel split & shuffle. ENet [31] proposes an asymmetric encoder-decoder architecture with early downsampling. Despite the substantial efforts, these networks are either still computationally demanding or sacrifice performance for model size shrinkage. We propose MPF-Net that contains 20.9% of memory cost and 0.95% of parameters of U-Net yet provides surpassing performance.

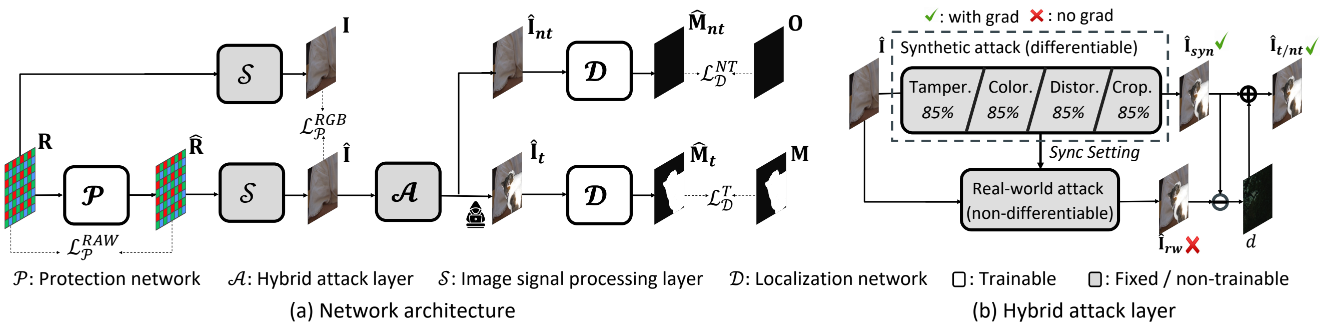


Figure 3. **Pipeline design of DRAW.** We design a lightweight protection network that embeds imperceptible protective signal in the RAW domain and transfers it into the rendered RGB images. On the recipient's side, the localization network identifies the forged areas.

3. Proposed Method

3.1. Approach

Fig. 3 depicts the pipeline design of DRAW. We denote the captured RAW data as $\mathbf{R}$, and use a protection network $\mathcal{P}$ to transform $\mathbf{R}$ into the protected RAW, i.e., $\hat{\mathbf{R}}$. The functionality of $\mathcal{P}$ is to adaptively embed a transferrable protective signal into $\hat{\mathbf{R}}$ for robust and accurate image manipulation localization in the RGB domain. Considering the computational limitation of imaging equipment, we use a novel lightweight MPF-Net specified in Section 3.2 to implement $\mathcal{P}$. Next, we use the ISP layer $\mathcal{S}$ to render $\hat{\mathbf{R}}$ into the protected RGB image $\hat{\mathbf{I}}$. Provided with a number of off-the-shelf deep-network-based ISP algorithms and non-differentiable conventional ISP algorithms, during training, we include a popular conventional method, i.e., LibRaw [1] and two deep-learning methods, i.e., CycleISP [46] and InvISP [41], and leave other ISP algorithms [45, 2] for evaluation. To improve generalizability, interpolation is conducted on one network-rendered RGB $\hat{\mathbf{I}}_{net}$ and one conventional-algorithm-generated RGB $\hat{\mathbf{I}}_{conv}$ to produce $\hat{\mathbf{I}}$, i.e., $\hat{\mathbf{I}} = \omega \cdot \hat{\mathbf{I}}_{conv} + (1-\omega) \cdot \hat{\mathbf{I}}_{net}$, where ω is uniformly within $[0, 1]$.

Afterward, to simulate image redistribution of $\hat{\mathbf{I}}$, we include the hybrid attack layer $\mathcal{A}$ to perform manipulation and lossy operations on $\hat{\mathbf{I}}$. It comprises of modules for tampering, color adjustments, distortions (lossy operations) and cropping. To construct the binary tampering mask $\mathbf{M}$, we apply the free-form mask generation [44] to arbitrarily select areas within $\hat{\mathbf{I}}$. In line with typical forgery detection works [10, 38], we consider inpainting, splicing and copy-moving as the most common three types of tampering, which often alter the underlying meaning of an image. In contrast, color adjustment and distortions are often considered benign yet can potentially erase traces for manipulation localization. During training, these modules can be conditionally performed according to the empirical *activation possibilities* (85%) and in any arbitrary ordering to encourage diversity, e.g., tampering then distorting, cropping then tampering, etc. We respectively denote the attacked images as $\hat{\mathbf{I}}_t$ if the tampering module is activated or $\hat{\mathbf{I}}_{nt}$ if otherwise. The latter is identified as authentic images, whose introduction is to explicitly minimize the false alarm rate of DRAW. Detailed implementations of these modules are specified in the supplement. Besides, to closer the gap between real and simulated lossy operations and color jittering operations, we add the difference between $\hat{\mathbf{I}}_{syn}$ and $\hat{\mathbf{I}}_{rw}$ on to $\hat{\mathbf{I}}_{syn}$, where $\hat{\mathbf{I}}_{syn}$ and $\hat{\mathbf{I}}_{rw}$ respectively denote synthetic and real-world processed image using the same setting. $x = \hat{\mathbf{I}}_{syn} + sg(\hat{\mathbf{I}}_{rw} - \hat{\mathbf{I}}_{syn}), x \in \{\hat{\mathbf{I}}_t, \hat{\mathbf{I}}_{nt}\}$, where sg stands for the stop-gradient operator [7].

On the recipient's side, we use the localization network $\mathcal{D}$ to estimate the manipulated region given a doubted image that could be one of $\hat{\mathbf{I}}_t$ or $\hat{\mathbf{I}}_{nt}$. If it's an manipulated image $\hat{\mathbf{I}}_t$, the predicted mask $\hat{\mathbf{M}}_t$ should be close to the ground-truth $\mathbf{M}$. Otherwise, it should be close to a zero matrix. DRAW is flexible on the selection of $\mathcal{D}$, where many off-the-shelf networks can be applied, e.g., DRAW-HRNet [35], DRAW-MVSS [10] or DRAW-RIML [38].

Objective Loss Functions. We need to include fidelity terms $\mathcal{L}_{\mathcal{P}}^{RAW}$ and $\mathcal{L}_{\mathcal{P}}^{RGB}$ to ensure imperceptible protection using the ℓ_1 distance.

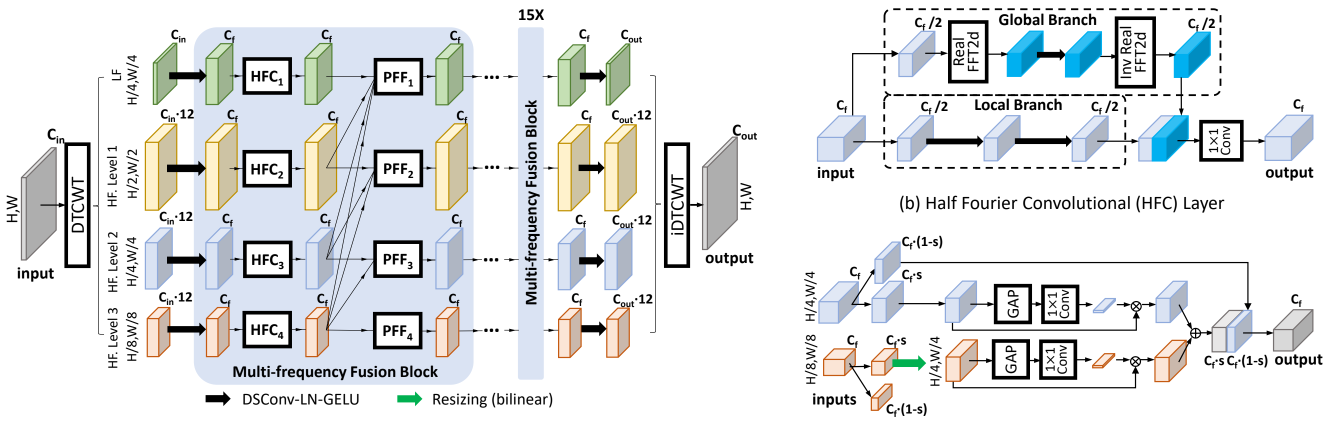
$$\begin{aligned} \mathcal{L}_{\mathcal{P}}^{RAW} &= \mathbb{E}_{\mathbf{R}}\left[\|\mathbf{R} - \mathcal{P}(\mathbf{R})\|_1\right], \\ \mathcal{L}_{\mathcal{P}}^{RGB} &= \mathbb{E}_{\mathbf{R}}\left[\|\mathcal{S}(\mathbf{R}) - \mathcal{S}(\mathcal{P}(\mathbf{R}))\|_1\right]. \end{aligned} \tag{1}$$

Next, we include localization terms to minimize the Binary Cross Entropy (BCE) losses that respectively compare $\hat{\mathbf{M}}_t$ with $\mathbf{M}$, and $\hat{\mathbf{M}}_{nt}$ with a zero matrix.

$$\begin{aligned} L_{\mathcal{D}}^{T} &= -\mathbb{E}_{\hat{\mathbf{I}}_t}\left[\mathbf{M}\log\left(\mathcal{D}(\hat{\mathbf{I}}_t)\right) + (1-\mathbf{M})\log\left(1 - \mathcal{D}(\hat{\mathbf{I}}_t)\right)\right], \\ L_{\mathcal{D}}^{NT} &= -\mathbb{E}_{\hat{\mathbf{I}}_{nt}}\left[\log\left(1 - \mathcal{D}(\hat{\mathbf{I}}_{nt})\right)\right]. \end{aligned} \tag{2}$$

The total loss for DRAW is shown in Eq. (3), where $\alpha, \beta, \gamma, \epsilon$ are empirically-set hyper-parameters.

$$\begin{aligned} \mathcal{L} &= \alpha \cdot \mathcal{L}_{\mathcal{P}}^{RAW} + \beta \cdot \mathcal{L}_{\mathcal{P}}^{RGB} + \gamma \cdot \mathcal{L}_{\mathcal{D}}^{T} + \epsilon \cdot \mathcal{L}_{\mathcal{D}}^{NT}, \\ \alpha &= 10, \beta = 1, \gamma = 0.02, \epsilon = 0.01. \end{aligned} \tag{3}$$

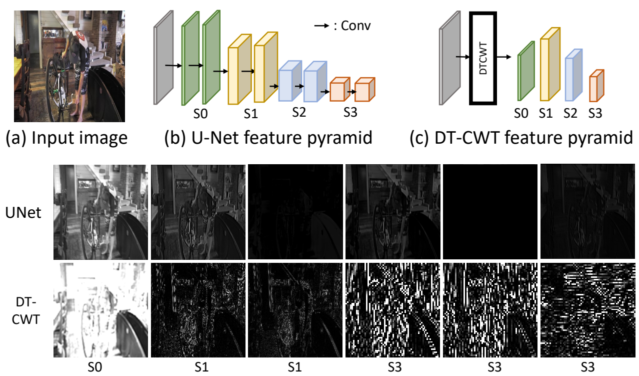


(a) Multi-frequency Partial Fusion Network (MPF-Net)

(b) Half Fourier Convolutional (HFC) Layer

(c) Partial Feature Fusion (PFF) Layer (PFF_3 as example)

Figure 4. **Network design of Multi-frequency Partial fusion Network (MPF-Net).** It decomposes the input into multi-level subbands and during cross-frequency feature fusion, we preserve a proportion of features learned in the current layer. $C_{in} = C_{out} = 3$ and $C_f = 32$.



(d) Comparison of the multi-scale features respectively by U-Net and DTCWT

Figure 5. **Illustration of feature mining respectively using DT-CWT transform and U-Net.** DT-CWT requires fewer *Conv* layers yet the generated features show less redundancy or repetition.

3.2. Multi-frequency Partial Fusion Network

In order to combat sophisticated image manipulation within resource-limited environments such as cellphones and cameras, it is essential to deploy a lightweight architecture yet with rich feature extraction capabilities. Fig. 4 illustrates the network design, where we first use a three-level DT-CWT transform to decompose the input into a low-frequency main component and three levels of higher-frequency subbands. Each level consists of six subbands in complex forms, representing different degrees of wavelet information. The real and imaginary parts of the subbands are then concatenated. In Fig. 5, we compare the feature pyramid of U-Net to that of DT-CWT. Vanilla convolutions can be less efficient due to the restriction of receptive field, feature redundancy, and repetition during training. In contrast, DT-CWT provides a strong prior for mitigating these issues, requiring only one layer of separable convolution and yielding richer patterns within representations.

Following the initial feature extraction, we apply a “DSConv-LN-GELU" layer to further refine the extracted features, which is in short for depth-wise separable convolution [17], Layer Normalization [5] and GELU activation [16]. Next, we cascade sixteen multi-frequency partial fusion blocks in each level as feature refinement and fusion. Each block contains a Half Fourier Convolution (HFC) layer and a Partial Feature Fusion (PFF) layer. Notably, these blocks do not alter either the resolution or channel number of the features. Then we project the features back into the main components and three levels of subbands using another “DSConv-LN-GELU” layer, which are then transformed back into the RGB domain via iDT-CWT.

Half Fourier Convolution Layer (HFC). We observe that features provided by DT-CWT provide a rich local pattern, whereas the global information representation is lacking. Considering that Fast Fourier Transform (FFT) is efficient in giving global information about the frequency components of an image [50, 24], we include both vanilla *Conv* layer and Fast Fourier Transform (FFT) in each HFC to enable simultaneous global and local feature mining. For the HFC layer at level i:

$$\begin{aligned} HFC_i : output &= [GB(input_1), LB(input_2)], \\ input &= [input_1, input_2], \end{aligned} \tag{4}$$

where we evenly split the input tensor by half, send them respectively into the Global Branch (GB) and Local Branch (LB) of the HFC layer, and concatenate the resultant features. GB contains FFT, *Conv* layer and inverse FFT. LB is composed of a cascade of two vanilla *Conv* layers.

Partial Feature Fusion Layer (PFF). On fusing different groups of features, two most commonly-accepted ways are “concatenate-and-reduce" [11, 35] or “attend-to-aggregate" [15, 47]. We propose a novel paradigm of “reserve-attend-and-assemble". Specifically, we split the

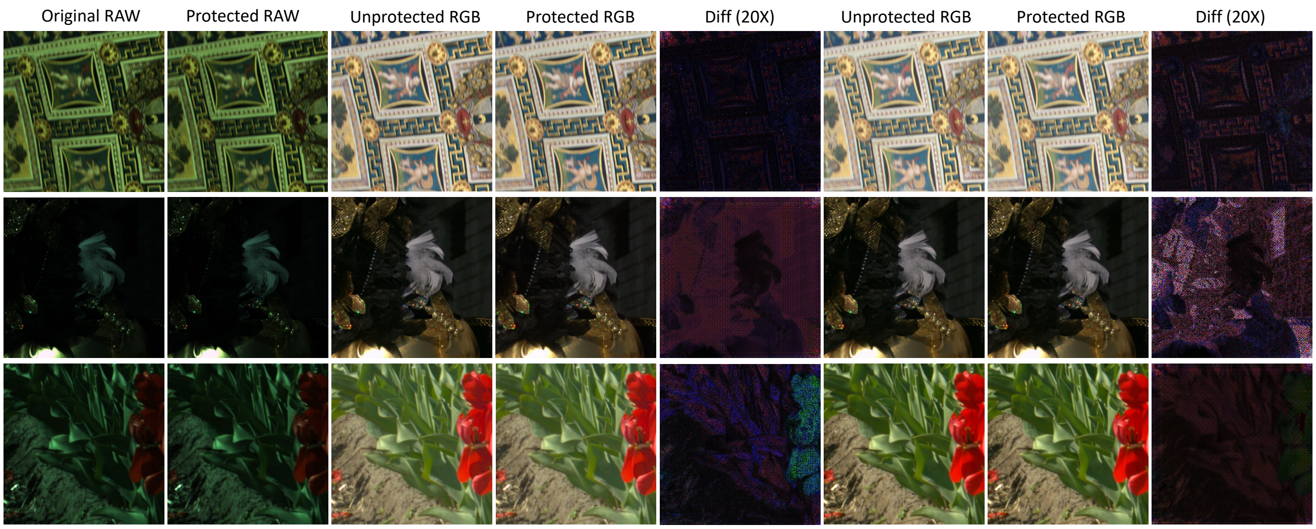


Figure 6. **Examples of protected images under different ISPs.** Dataset: RAISE. In each test, we apply two ISPs for rendering (upper: LibRAW / OpenISP, middle: InvISP / CycleISP, lower: OpenISP / InvISP). The RAW images are visualized through bilinear demosaicing.

Table 1. **Quantitative analysis on the imperceptibility of RAW protection.** $[\mathbf{R}, \hat{\mathbf{R}}]$: RAW file before and after protection. $[\mathbf{I}, \hat{\mathbf{I}}]$: RGB file rendered respectively from $\mathbf{R}$ and $\hat{\mathbf{R}}$ using different ISP pipelines. Dataset: RAISE and Canon.

Process	512×512		256×256		1024×1024	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$[\mathbf{R}, \hat{\mathbf{R}}]$	58.43	-	61.67	-	56.41	-
$[\mathbf{I}, \hat{\mathbf{I}}]$ (InvISP)	45.13	0.977	46.20	0.985	45.60	0.983
$[\mathbf{I}, \hat{\mathbf{I}}]$ (LibRaw)	41.25	0.960	41.97	0.967	41.07	0.957
$[\mathbf{I}, \hat{\mathbf{I}}]$ (Restormer)	45.75	0.980	46.24	0.984	45.03	0.977
$[\mathbf{I}, \hat{\mathbf{I}}]$ (OpenISP)	40.52	0.960	41.95	0.966	40.34	0.955

input features into two halves based on a predetermined ratio s (default 0.25), i.e., $input_i = [input_{i,1}, input_{i,2}]$ for PFF at level i. The first half of the multi-level features $(C_f \cdot\ s)$ are resized into the size of the current level, and then separately reweighed using channel attention (CA) [18]. Next, “assemble" is done by pixel-wisely aggregating all groups of reweighed features and concatenating them with the reserved second half $(C_f \cdot\ (1-s))$. Our paradigm can potentially mitigate the issue of over-attention on certain frequencies or covariance drift of the preserved representation, especially from shallow layers, caused by residual learning. Furthermore, we only pass higher-frequency subbands into lower levels, which also encourages each level to process unique combinations of frequencies which reduces redundancy. The operations in PFF at level i is as follows.

$$PFF_i : output = [input_{i,2}, \sum_{j \leq\ i} CA(Resize(input_{j,1}))] \quad (5)$$

where CA is composed of a global average pooling layer and a 1×1 bottleneck convolution.

4. Experiments

4.1. Experimental Setups

We use RAISE [12] dataset (8156 image pairs) and Canon subset (2997 image pairs) from the FiveK [8] dataset as the training set. Meanwhile, RAISE, Canon subset and Nikon subset (1600 image pairs) from FiveK as well as SIDD dataset [3] are used to evaluate DRAW. We divide them into training sets and test sets at a ratio of 85: 15. We crop each RAW image into non-overlapping sub-images sized $512 \times\ 512$. For quantitative analysis, manually manipulating all protected images requires unaffordable effort. Alternatively, inspired by [49, 38], we borrow the segmentation masks from MS-COCO [28] dataset, crop out the corresponding objects and iteratively add them onto the protected images $\hat{\mathbf{I}}$ until the total manipulation rate exceeds 5%. For *copy-moving* and *inpainting*, we generate the attacked image under the same principle that was used during training. For qualitative analysis, we also manually manipulate over one hundred protected images and show some of the representative examples in the figures.

We train our benchmark model by jointly training $\mathcal{P}$ with HRNet [35] as $\mathcal{D}$. We then fix $\mathcal{P}$ and respectively training MVSS [10] and RIML [38] as $\mathcal{D}$ on top of the protected RGB images. All models are trained with batch size 16 on four distributed NVIDIA RTX 3090 GPUs, and we train the networks for 10 epochs in roughly one day. For gradient descent, we use Adam optimizer with the default hyperparameters. The learning rate is 1×10^{-4}.

4.2. Performances

Image Quality Assessment. Fig. 6 and Table 1 respectively show the qualitative and quantitative results on the

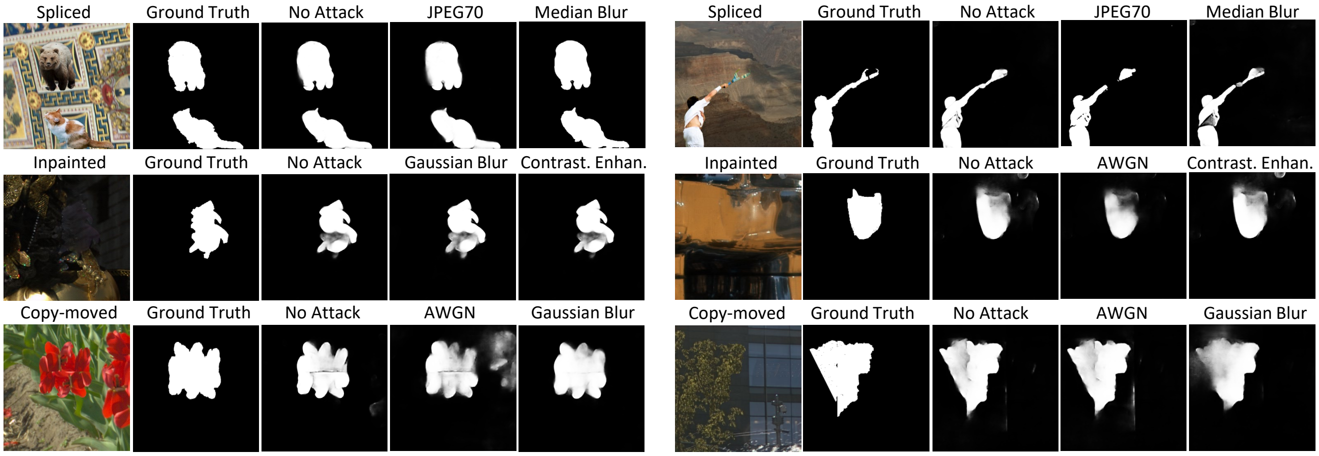


Figure 7. **Exampled forgery localization results of DRAW-HRNet.** Dataset: RAISE and Canon.

Table 2. **Average performance of different methods on forgery localization.** Dataset: RAISE. The best performances are highlighted in bold type. *: open-source pretrained models finetuned on original RAISE images with *copy-moving*, *splicing* and *inpainting*.

	Models	No Attack			Rescaling			AWGN			JPEG90			JPEG70			MBlur			GBlur		
		Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU
splicing	MVSS*	.908	.725	.597	.715	.609	.470	**.954**	.688	.547	**.944**	.627	.481	**.915**	.565	.415	.869	.695	.561	.181	.211	.138
	RIML*	**.941**	**.949**	**.908**	.732	.795	.702	.900	.918	.863	.869	.892	.821	.777	.818	.721	.900	.918	.857	.096	.142	.094
	DRAW-MVSS	.867	.874	.793	.553	.636	.514	.886	.854	.764	.878	.856	.767	.820	.789	.680	.732	.770	.658	.320	.419	.301
	DRAW-RIML	.897	.926	.876	.877	.910	.856	.928	**.946**	**.905**	.913	.932	.884	.889	**.909**	**.849**	.917	.939	**.893**	**.556**	**.639**	**.544**
	DRAW-HRNet	.936	.947	.903	**.922**	**.934**	**.884**	.929	.934	.883	.933	**.935**	**.885**	.902	.861	.776	**.927**	**.940**	.891	.552	.638	.523
copy-moving	MVSS*	.833	.781	.703	.677	.636	.544	.861	.755	.668	.771	.627	.527	.653	.471	.366	.795	.731	.640	.339	.336	.258
	RIML*	.888	.889	.856	.774	.793	.737	.896	.895	.861	.829	.835	.788	.694	.719	.657	.850	.856	.811	.557	.572	.493
	DRAW-MVSS	.901	.893	.857	.839	.836	.780	.915	.890	.850	.862	.842	.793	.804	.767	.706	.871	.851	.803	.631	.657	.582
	DRAW-RIML	.915	.925	.910	.875	.895	.868	.906	.918	.899	.884	.899	.874	.845	.866	.829	.897	.910	.888	.774	.811	.768
	DRAW-HRNet	**.969**	**.970**	**.959**	**.960**	**.956**	**.937**	**.962**	**.957**	**.943**	**.955**	**.951**	**.932**	**.916**	**.884**	**.839**	**.958**	**.955**	**.939**	**.915**	**.920**	**.885**
inpainting	MVSS*	.259	.229	.172	.101	.062	.039	.404	.360	.263	.180	.090	.054	.212	.097	.058	.088	.050	.030	.085	.043	.026
	RIML*	.126	.140	.097	.035	.047	.030	.132	.155	.113	.014	.020	.013	.001	.001	.001	.037	.043	.026	.068	.077	.048
	DRAW-MVSS	.737	.752	.672	.657	.682	.588	.771	.756	.667	.617	.645	.546	**.515**	**.536**	**.434**	.567	.595	.497	.514	.561	.463
	DRAW-RIML	.663	.716	.656	.457	.518	.452	.667	.718	.654	.348	.411	.342	.091	.121	.089	.366	.423	.360	.284	.338	.281
	DRAW-HRNet	**.776**	**.791**	**.735**	**.754**	**.760**	**.685**	**.788**	**.771**	**.697**	**.719**	**.714**	**.625**	.468	.454	.346	**.732**	**.735**	**.647**	**.686**	**.704**	**.618**

imperceptibility of the protection. Besides, we test the overall image quality of protected images using untrained ISP network, namely, Restormer [45], and another conventional ISP, namely, OpenISP [2]. Restormer is originally proposed for image restoration, but we find that the transformer-based architecture also shows excellent performance on RGB image rendering. OpenISP is another popular open-source ISP pipeline apart from LibRaw, and we customize the pipeline by applying the most essential modules. We can observe little artifact from the protected version of RAW data and RGB. From the augmented difference, DRAW imperceptibly introduce content-related local patterns, which function like digital *locks* onto the pixels and forgery localization is conducted by observing the integrity of these *locks*.

Robustness and Accuracy of Manipulation Localization. We conduct comprehensive experiments on RAISE and Canon datasets under different lossy operations. The qualitative and quantitative comparisons in terms of the Recall, F1 and IoU in the pixel domain are reported in Fig. 7, Fig. 8, Table 2 and Table 4. In Table 3, we further conduct color adjustment attacks and hybrid attacks on the protected images and let the networks detect the forged areas. We find that for DRAW-HRNet, although the images are manipulated by diverse lossy operations, we succeed in localizing the tampered areas. If there are no lossy operations, the F1 scores are in most cases above 0.8. Fig. 7 further provides exampled image manipulation localization results of DRAW-HRNet under different lossy operations.

Next, for fair comparison with previous arts, we finetune MVSS and RIML on RAISE and Canon dataset using the mechanisms proposed in the corresponding papers yet additionally considering *splicing*, *copy-moving* and *inpainting*. When heavy image lossy operations are present, MVSS fails to detect the tampered content. While RIML exhibits better robustness due to OSN transmission simulation, its performances under blurring or inpainting attacks are still restricted. However, training these detectors based on the protected images significantly improves their robustness. Besides, while MVSS and RIML are found limited in the generalizability to novel, untrained types of inpainting

Table 3. **Average performance against color adjustment attacks and hybrid attacks on RAISE dataset.** The detector can successfully locate the forged areas in most cases.

Attack	*splicing*			*copy-moving*			*inpainting*		
	Rec.	F1	IoU	Rec.	F1	IoU	Rec.	F1	IoU
Hue Adjust.	.938	.949	.905	.973	.974	.962	.779	.794	.736
Contra. Enhan.	.935	.945	.900	.971	.969	.958	.773	.783	.726
Satur. Adjust.	.937	.948	.904	.969	.968	.958	.784	.795	.738
Bright. Adjust.	.936	.947	.903	.960	.960	.948	.771	.782	.725
JPEG70+Hue.	.900	.855	.769	.906	.872	.824	.489	.508	.396
GBlur+Contra.	.553	.637	.520	.895	.902	.866	.755	.774	.692
MBlur+Satur.	.927	.939	.890	.960	.956	.938	.821	.832	.754
AWGN+Bright.	.930	.935	.885	.952	.944	.925	.842	.842	.778

Table 4. **Average performance of different methods on forgery localization.** Dataset: CANON.

	Models	No Attack		Rescaling		JPEG70		GBlur	
		F1	IoU	F1	IoU	F1	IoU	F1	IoU
splicing	MVSS*	.610	.465	.530	.390	.503	.354	.210	.135
	RIML*	.925	**.872**	.716	.609	.783	.675	.136	.094
	DRAW-MVSS	.841	.738	.875	.789	.842	.739	.829	.731
	DRAW-RIML	.887	.818	.925	.870	.855	.769	.906	.843
	DRAW-HRNet	**.926**	.869	**.939**	**.889**	**.921**	**.861**	**.939**	**.891**
copy-moving	MVSS*	.727	.628	.624	.526	.455	.341	.371	.283
	RIML*	.892	.852	.789	.733	.702	.619	.569	.494
	DRAW-MVSS	.912	.879	.868	.828	.832	.785	.826	.776
	DRAW-RIML	**.969**	**.957**	**.962**	**.947**	.911	.882	.952	.933
	DRAW-HRNet	.968	.956	.960	.945	**.922**	**.895**	**.952**	**.934**
inpainting	MVSS*	.215	.150	.100	.064	.136	.083	.073	.045
	RIML*	.102	.070	.033	.021	.003	.001	.012	.007
	DRAW-MVSS	.829	.753	.676	.574	.115	.076	.513	.407
	DRAW-RIML	**.949**	**.918**	**.889**	**.836**	**.406**	**.326**	**.849**	**.784**
	DRAW-HRNet	.934	.894	.867	.811	.360	.284	.761	.691

Table 5. **Generalizability to untrained ISP pipelines or datasets**. $\mathcal{P}$ and $\mathcal{D}$ are trained on RAISE.

Test Item		Forgery	NoAtk	Rescaling	JPEG70	MBlur	GBlur
ISP	OpenISP	Spli.	.929	.910	.837	.933	.620
		Copy.	.941	.919	.843	.941	.880
		Inpa.	.850	.820	.451	.765	.756
	Restormer	Spli.	.946	.936	.863	.941	.648
		Copy.	.961	.947	.871	.948	.904
		Inpa.	.906	.833	.487	.789	.759
Dataset	Canon	Spli.	.936	.925	.845	.931	.596
		Copy.	.957	.930	.859	.946	.881
		Inpa.	.805	.732	.486	.710	.706
	SIDD	Spli.	.928	.909	.832	.911	.574
		Copy.	.967	.965	.891	.954	.880
		Inpa.	.686	.628	.400	.574	.554

forgeries such as ZITS and LAMA, DRAW can also help improve their accuracy and robustness against such types, therefore ensuring generalizability even without frequently updating the trained parameters.

Generalizability. We conduct additional experiments where $\mathcal{P}$ trained on RAISE dataset is applied on different RAW datasets, i.e., Canon and SIDD, and untrained ISP pipelines, i.e., OpenISP and Restormer. Table 5 shows that raw protection can generalize to untrained cameras and ISP pipelines while preserving promising detection capacity. For instance, given the new ISPs, the F1 scores under JPEG70 attack for *copy-moving* and *splicing* detection are above 0.7, representing successful manipulation localization. Therefore, our method is shown to adapt to untrained ISP pipelines.

Table 6. **Comparison of computational cost** among lightweight image-to-image-translation or segmentation networks.

	SegNet [6]	ShuffleNet [30]	U-Net [33]	ENet [31]	MPF-Net
Params	29.5M	0.94M	26.35M	0.36M	0.25M
FLOPS	0.56T	22.9G	0.22T	2.34G	7.39G
Mem.	465MB	390MB	767MB	46MB	160MB

Table 7. **Comparison with baseline methods on RAISE.** We verify the importance of RAW protection by comparing the results with those of pure robust training using $\mathcal{A}$ and direct RGB protection. $\mathcal{P}^-$: using $\mathcal{P}$ for RGB protection. $\mathcal{D}$: MVSS*

	Used Modules				Rescaling		JPEG70		MBlur		GBlur	
	$\mathcal{P}$	$\mathcal{P}^-$	$\mathcal{A}$	$\mathcal{D}$	F1	IoU	F1	IoU	F1	IoU	F1	IoU
splicing				✓	.609	.470	.565	.415	.695	.561	.211	.138
			✓	✓	**.668**	**.534**	.725	.590	.762	.635	.303	.207
		✓	✓	✓	.358	.253	.438	.317	.487	.361	.149	.097
	✓		✓	✓	.636	.514	**.789**	**.680**	**.770**	**.658**	**.419**	**.301**
copy-moving				✓	.636	.544	.471	.366	.731	.640	.336	.258
			✓	✓	**.859**	**.816**	.648	.582	.782	.728	.528	.456
		✓	✓	✓	.490	.412	.467	.382	.626	.548	.268	.208
	✓		✓	✓	.836	.780	**.767**	**.706**	**.851**	**.803**	**.657**	**.582**
inpainting				✓	.062	.039	.097	.058	.050	.030	.043	.026
			✓	✓	.605	.494	.231	.159	.398	.297	.342	.249
		✓	✓	✓	.387	.291	.480	.371	.381	.288	.374	.279
	✓		✓	✓	**.682**	**.588**	**.536**	**.434**	**.595**	**.497**	**.561**	**.463**

Computational Complexity. We compare the computational requirements of MPF-Net in Table 6 with SegNet [6], ShuffleNet [30], U-Net [33] and ENet [31], which are famous lightweight models for image segmentation. MPF-Net requires lower computing resources, e.g, only 20.9% in memory cost and 0.95% in parameters compared to the classical U-Net.

4.3. Baseline Comparisons

Previous techniques in proactive image forgery detection, e.g., tag retrieval [37] or template matching [4], are not suitable for image manipulation localization. Therefore, we alternatively build two baseline methods that respectively apply pure robust training using our proposed attack layer and apply RGB-domain protection. In the tests, MVSS is employed as localization network. The quantitative comparison results are reported in Table 7. Further details regarding the experimental settings for the two baseline methods are included in the supplement.

RAW Protection vs Pure Robust Training. Our proposed robust training mechanism reflected in the attack layer is different from that proposed in RIML. Specifically, we render the unprotected RAW files $\mathbf{R}$ using $\mathcal{S}$, which are then attacked by $\mathcal{A}$. We see that the introduction of robust training can help boost the performance of MVSS. However, the overall performance is still worse than further applying RAW protection to aid localization. In severe degrading cases such as blurring, the performance gap between RAW

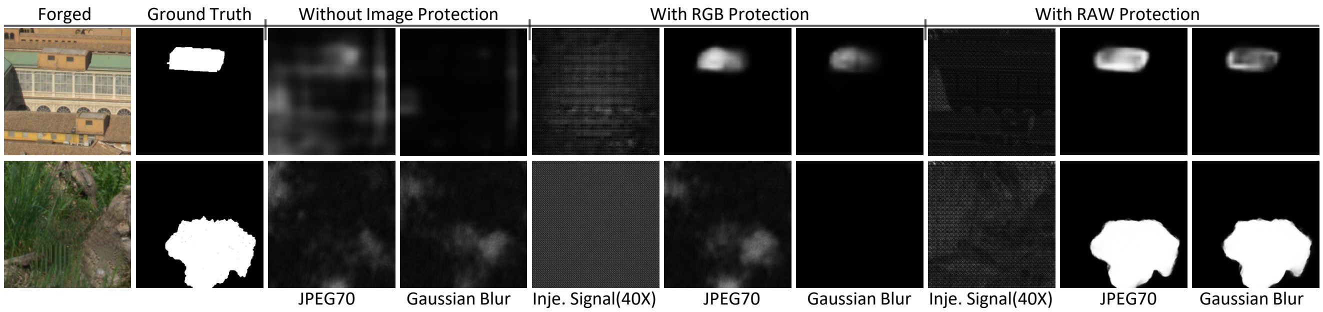


Figure 8. **Baseline analysis on performance between passive localization without image protection, with RGB protection and with RAW protection.** Dataset: RAISE. $\mathcal{D}$: MVSS* (upper), RIML* (lower). Type: copy-moving (upper), inpainting (lower).

protection and robust training without protection regarding F1 score is more than ten percent.

RAW protection vs RGB protection. For fair comparison, we regulate that the overall PSNR on RGB images before and after RGB protection should be above 40 dB, in line with the criterion in Table 1. We conduct qualitative experiment in Fig 8 to evaluate the effectiveness of image protection. According to the experimental results, RGB protection cannot aid robust manipulation localization if the magnitude of RGB modification is restricted. We also grayscale the augmented injected signal for better visualization and found that signal injected by RAW protection is more adaptive in magnitude to the image contents. One possible reason is that the densely-predicting task requires hiding more information than binary image forgery classification task, making it struggle to maintain high fidelity of the original image. In comparison, RAW protection can adaptively introduce protection with the help of content-related procedures, e.g., demosaicing and noise reduction, within the subsequent ISP algorithms that suppress unwanted artifacts and biases. Theoretically, RAW data modification enjoys a much larger search space that allows transformations from the original image into another image with high density upon sampling.

4.4. Ablation Studies

Table 8 and Fig. 9 respectively show the quantitative and qualitative results of ablation studies. In each test, we regulate that the averaged PSNR between $\mathbf{I}$ and $\hat{\mathbf{I}}$, with ISP pipelines evenly applied, should be within the range of 41-43 dB, to ensure imperceptible image protection.

Substituting the architecture of $\mathcal{P}$. We first test if using U-Net with a similar amount of parameters or ENet [31] as $\mathcal{P}$ can achieve similar performance on splicing detection. First, though ENet contains similar amount of parameters compared to MPF-Net, the performance of image manipulation localization using ENet as $\mathcal{P}$ is not satisfactory. Second, though U-Net with *DSConv* provides much better result, because the channel numbers within each layer are restricted within 48 to save computational complexity, the performance is still worse than our benchmark.

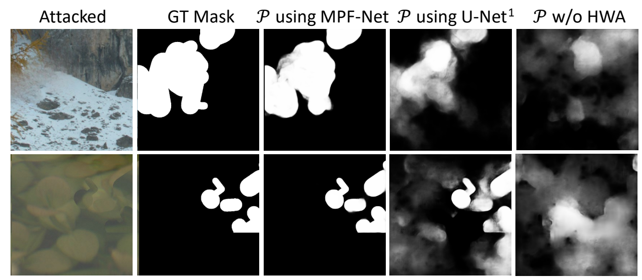


Figure 9. **Examples of ablation studies of DRAW.** We observe that either replacing MPF-Net with U-Net using *DSConv* or removing HFC module results in decreased performance. Upper: inpainting + JPEG80. Lower: copy-moving + median-blur.

Table 8. **Ablation study on DRAW on Nikon using splicing attack.** [1]: replacing *Conv* layers with *DSConv*.

Test	F1 NoAtk	F1 JPEG70	F1 Mblur
$\mathcal{P}$ using U-Net[1]	.877	.769	.535
$\mathcal{P}$ using ENet	.324	.137	.092
MPF-Net w/o HFC	.844	.710	.602
MPF-Net w/o DT-CWT	.852	.751	.626
MPF-Net w/o PFF	.827	.712	.667
w/o diff from real attack	.842	.566	.502
using only one ISP Surrogate	.648	.455	.267
w/o Image Distortion Module	.929	.245	.116
w/o Color Ajustment Module	.814	.759	.641
Full implementation of **DRAW**	.929	**.838**	**.696**

Impact of components in MPF-Net. The most noticeable difference between MPFNet with previous U-shaped networks is that feature disentanglement can be better ensured even with fewer parameters. To verify this, we respectively replace the HFC layer and PFF layer with typical alternatives, i.e., vanilla convolution and channel-wise concatenation. The performances are nearly 5-10 points weaker compared to the MPF-Net setup. First, DT-CWT is a shift-invariant wavelet transform that comes with limited redundancy. Second, partial feature fusion and partial connection are more flexible. The design explicitly keeps some of the features extracted from the current level and directly feeds them into the subsequent block. Therefore, for different levels, the input features will be different, which encourages feature disentanglement.

Impact of pipeline design. We also test the setting of not using the image distortion module or color adjustment module in the pipeline during training. The result is as expected that the scheme will therefore lack generalizability in overall robustness due to the fact that there are not enough random processes that can simulate the real-world situation. Besides, not introducing the difference between the real-world and simulated attacks or using only one ISP surrogate model will also impair the overall performance.

5. Conclusions

We present DRAW which introduces invisible watermarks as protective signal into the RAW data. The protection can not only be transferred into the rendered RGB images regardless of the applied ISP pipeline, but also is resilient to post-processing operations such as blurring or compression. Once the image is manipulated, we can accurately identify the forged areas with a localization network. Extensive experiments on typical RAW datasets prove the effectiveness of DRAW. We also verify that our novel MPF-Net provides superior performance compared to previous lightweight models for our task.

Acknowledgment. This work was supported by National Natural Science Foundation of China under Grant U20B2051, U1936214, U22B2047, 62072114 and U20A20178.

References

[1] Libraw: Library for reading and processing of raw digicam images. *https://github.com/LibRaw/LibRaw*. 3

[2] Openisp. *https://github.com/mushfiqulalam/isp*. 3, 6

[3] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 5

[4] Vishal Asnani, Xi Yin, Tal Hassner, Sijia Liu, and Xiaoming Liu. Proactive image manipulation detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15386–15395, 2022. 2, 7

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[6] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 7

[7] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 3

[8] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR 2011*, pages 97–104. IEEE, 2011. 5

[9] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukás. Determining image origin and integrity using sensor noise. *IEEE Transactions on information forensics and security*, 3(1):74–90, 2008. 1

[10] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14185–14193, 2021. 1, 2, 3, 5

[11] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4641–4650, 2021. 4

[12] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM multimedia systems conference*, pages 219–224, 2015. 5

[13] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Incremental transformer structure enhanced image inpainting with masking positional encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11368, 2022. 1

[14] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on information Forensics and Security*, 7(3):868–882, 2012. 2

[15] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019. 4

[16] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2, 4

[18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 5

[19] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. Span: Spatial pyramid attention network for image manipulation localization. In *European Conference on Computer Vision (ECCV)*, pages 312–328. Springer, 2020. 1, 2

[20] Ashraful Islam, Chengjiang Long, Arslan Basharat, and Anthony Hoogs. Doa-gan: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4676–4685, 2020. 2

[21] Junpeng Jing, Xin Deng, Mai Xu, Jianyi Wang, and Zhenyu Guan. Hinet: Deep image hiding by invertible network. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4733–4742, 2021. 2

[22] David Khachaturov, Ilia Shumailov, Yiren Zhao, Nicolas Papernot, and Ross Anderson. Markpainting: Adversarial machine learning meets inpainting. In *International Conference on Machine Learning (ICML)*, pages 5409–5419, 2021. 2

[23] Kenji Kurosawa, Kenro Kuroki, and Naoki Saitoh. Ccd fingerprint method-identification of a video camera from videotaped images. In *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, volume 3, pages 537–540. IEEE, 1999. 1

[24] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021. 4

[25] Haodong Li and Jiwu Huang. Localization of deep inpainting using high-pass fully convolutional network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8301–8310, 2019. 2

[26] Yuanman Li and Jiantao Zhou. Fast and effective image copy-move forgery detection via hierarchical feature point matching. *IEEE Transactions on Information Forensics and Security (TIFS)*, 14(5):1307–1322, 2018. 2

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[29] Shao-Ping Lu, Rong Wang, Tao Zhong, and Paul L Rosin. Large-capacity image steganography based on invertible neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10816–10825, 2021. 2

[30] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 2, 7

[31] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016. 2, 7, 8

[32] Alin C Popescu and Hany Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, 2005. 1

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015. 2, 7

[34] Ronald Salloum, Yuzhuo Ren, and C-C Jay Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *Journal of Visual Communication and Image Representation*, 51:201–209, 2018. 2

[35] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 3, 4, 5

[36] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2149–2159, 2022. 1

[37] Run Wang, Felix Juefei-Xu, Meng Luo, Yang Liu, and Lina Wang. Faketagger: Robust safeguards against deepfake dissemination via provenance tracking. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3546–3555, 2021. 2, 7

[38] Haiwei Wu, Jiantao Zhou, Jinyu Tian, and Jun Liu. Robust image forgery detection over online social network shared images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13440–13449, 2022. 1, 2, 3, 5

[39] Haiwei Wu, Jiantao Zhou, Jinyu Tian, Jun Liu, and Yu Qiao. Robust image forgery detection against transmission over online social networks. *IEEE Transactions on Information Forensics and Security*, 17:443–456, 2022. 1

[40] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9543–9552, 2019. 1, 2

[41] Yazhou Xing, Zian Qian, and Qifeng Chen. Invertible image signal processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6287–6296, 2021. 3

[42] Minghao Yin, Yongbing Zhang, Xiu Li, and Shiqi Wang. When deep fool meets deep prior: Adversarial attack on super-resolution network. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1930–1938, 2018. 2

[43] Qichao Ying, Jingzhi Lin, Zhenxing Qian, Haisheng Xu, and Xinpeng Zhang. Robust digital watermarking for color images in combined dft and dt-cwt domains. *Mathematical Biosciences and Engineering*, 16(5):4788–4801, 2019. 2

[44] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4471–4480, 2019. 3

[45] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5728–5739, 2022. 3, 6

[46] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling

Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2696–2705, 2020. 3

[47] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 492–511. Springer, 2020. 4

[48] Yuan Zhao, Bo Liu, Ming Ding, Baoping Liu, Tianqing Zhu, and Xin Yu. Proactive deepfake defence via identity watermarking. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4602–4611, 2023. 2

[49] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1053–1061, 2018. 2, 5

[50] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pages 27268–27286. PMLR, 2022. 4

[51] Xinshan Zhu, Yongjun Qian, Xianfeng Zhao, Biao Sun, and Ya Sun. A deep learning approach to patch-based image inpainting forensics. *Signal Processing: Image Communication*, 67:90–99, 2018. 2