Robust Camera Model Identification Over Online Social Network Shared Images via Multi-Scenario Learning

Haiwei Wu, *Student Member, IEEE*, Jiantao Zhou, *Senior Member, IEEE*, Xinyu Zhang, Jinyu Tian, *Member, IEEE*, and Weiwei Sun

***Abstract*— Camera model identification (CMI) can be widely used in image forensics such as authenticity determination, copyright protection, forgery detection, etc. Meanwhile, with the vigorous development of the Internet, online social networks (OSNs) have become the dominant channels for image sharing and transmission. However, the inevitable lossy operations on OSNs, such as compression and post-processing, impose great challenges to the existing CMI schemes, as they severely destroy the camera traces left in the images under investigation. In this work, we propose a novel CMI method that is robust against the lossy operations of various OSN platforms. Specifically, it is observed that a camera trace extractor can be easily trained on a single degradation scenario (e.g., one specific OSN platform); while much more difficult on mixed degradation scenarios (e.g., multiple OSN platforms). Inspired by this observation, we design a new multi-scenario learning (MSL) strategy, enabling us to extract robust camera traces across different OSNs. Furthermore, noticing that image smooth regions incur less distortions by OSN and less interference by image signal itself, we suggest a SmooThness-Aware Trace Extractor (STATE) that can adaptively extract camera traces according to the smoothness of the input image. The superiority of our method is verified by comparative experiments with four state-of-the-art methods, especially under various OSN transmission scenarios. Particularly, for the open-set camera model verification task, we greatly surpass the second-place by 15.30% in AUC on the `FODB` dataset; while for the close-set camera model classification task, we are significantly ahead of the second-place by 34.51% in F1 on the `SIHDR` dataset. The code of our proposed method is available at https://github.com/HighwayWu/CameraTraceOSN.**

Manuscript received 24 November 2022; revised 4 August 2023 and 18 September 2023; accepted 18 September 2023. Date of publication 25 September 2023; date of current version 20 November 2023. This work was supported in part by the Macau Science and Technology Development Fund under Grant SKLIOTSC-2021-2023, Grant 0072/2020/AMJ, Grant 0022/2022/A1, and Grant 0014/2022/AFJ; in part by the Research Committee at University of Macau under Grant MYRG2020-00101-FST and Grant MYRG2022-00152-FST; in part by the Natural Science Foundation of China under Grant 61971476; and in part by the Alibaba Group through the Alibaba Innovative Research Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Benedetta Tondi. *(Corresponding author: Jiantao Zhou.)*

Haiwei Wu, Jiantao Zhou, and Xinyu Zhang are with the State Key Laboratory of Internet of Things for Smart City and the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 999078, China (e-mail: yc07912@umac.mo; jtzhou@umac.mo; mc14958@umac.mo).

Jinyu Tian is with the Faculty of Innovation Engineering, Macau University of Science and Technology, Macau 999078, China (e-mail: jytian@must.edu.mo).

Weiwei Sun is with the Alibaba Group, Hangzhou 311100, China (e-mail: sunweiwei.sww@alibaba-inc.com).

Digital Object Identifier 10.1109/TIFS.2023.3318968

***Index Terms*— Camera model identification, online social networks, deep neural networks, robustness.**

I. Introduction

EACH camera model generates unique pattern noise on the captured images. Such a pattern noise, also known as camera trace, is mainly generated by the different responses of the sensor to photons, and/or the internal image signal processing (ISP) pipeline [1], [2], [3]. The two most commonly-used traditional camera trace extraction methods are the photo-response non-uniformity (PRNU) [1] and the fixed pattern noise (FPN) [4]. Some recent efforts have also attempted to integrate PRNU and convolutional neural network (CNN) to extract camera traces, so as to achieve better discrimination performance [5], [6]. The high discriminability of camera trace makes it widely used in various forensics tasks, such as camera model identification (CMI) [7], forgery detection [5], transmission classification [8], [9].

The state-of-the-art performance of CMI is achieved by learning-based methods [6], [10], [13], which generally consist of two sub-networks, denoted as *extractor* and *classifier*. At the training stage, given a training image and its associated camera label, the extractor aims to extract the camera trace (high-dimensional feature vector), while the classifier supervises the training of the extractor by evaluating whether the extracted trace can be classified as the correct label. At the testing stage, the classifier is usually discarded, and the well-trained extractor can be exploited to extract camera traces for any input images. More discussions regarding the generic learning-based CMI will be given in Fig. 4 and Section III.

In practical situations, the images under investigation are usually not directly from cameras, but rather have been undergone with a series of processing. For instance, nowadays many images are obtained from various online social networks (OSNs), which inevitably apply many types of lossy operations such as JPEG compression, scaling, and post-processing [14], [15], [16]. These operations may interfere with existing camera trace extraction algorithms, causing them to suffer severe performance drop. An illustrating example is given in Fig. 1, which shows the performance on camera model verification task over the `FODB` [12] dataset. As can be observed, the state-of-the-art methods [5], [6], [10], [11] can achieve nearly

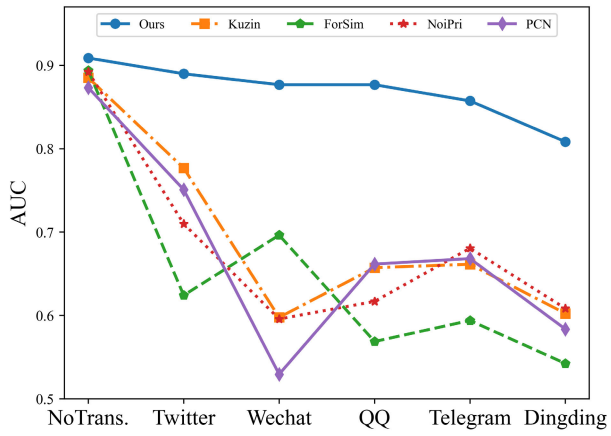

Fig. 1. Performance on open-set verification task of our proposed method, compared with the state-of-the-art schemes: Kuzin [10], NoiPri [5], ForSim [11], and PCN [6]. It should be noted that both the testing dataset FODB [12] and the considered OSN platforms (Twitter, WeChat, QQ, Telegram, and Dingding) are **unknown** at the training stage, mimicking a practical situation. Please refer to Section V for more details.

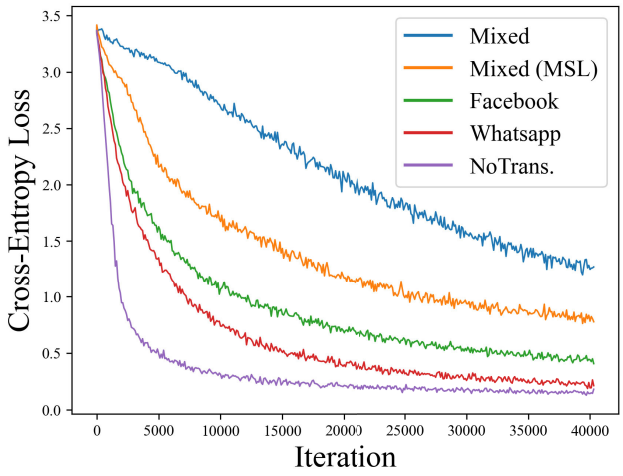

Fig. 2. Observation I: Training losses in the camera classification task on the VISION [17] dataset. The purple line represents the training with NoTrans. (original) images, while the red, green, and blue lines correspond to the single scenarios (Facebook and Whatsapp) and mixed scenario, respectively. Also, our proposed MSL is marked as the orange line.

90% AUC in the no-transmission (NoTrans.) scenario; but the performance of all existing algorithms declines seriously when considering the OSN transmissions, especially for WeChat, QQ, Telegram, and Dingding platforms.

To mitigate the negative impacts of OSNs, in this work, we propose a novel method for extracting camera traces, which are expected to be robust against the transmission of various OSN platforms. For simplicity, we mainly focus on the single round of OSN transmission cases, and later we also show some results on multiple rounds. It should be noted that we are more interested in designing a *unified* extractor, capable of extracting robust camera traces against different types of OSN transmission; rather than preparing a series of extractors, each of which corresponds to one specific type of OSN transmission. Our proposed robust CMI method is mainly inspired by two important observations.

1) Observation I: In Fig. 2, we demonstrate the loss curves of the extractor in the training process for the so-called *single* and *mixed* OSN scenarios. Here, "Single" means that the training data contain only one specific OSN transmission scenario, such as NoTrans., Facebook, or Whatsapp, corresponding to purple, green, and red lines in Fig. 2, respectively. In contrast, "Mixed" refers to the case that the training data include multiple OSN transmission scenarios, such as mixing the above three scenarios, as illustrated by the blue line. It can be clearly seen that the loss curve of the mixed scenario is much higher than that of each single scenario. This implies that the extractor has difficulty in extracting robust traces in mixed scenarios than in a single scenario. We thereby conjecture that the feature space of camera traces for different OSN transmission scenarios is more likely to be non-overlapping, making the training process of seeking local optima much more difficult. In other words, one particular camera trace that is robust for one given OSN transmission scenario may not be suitable for the other scenarios.

Therefore, to enable the camera trace extractor to better learn shared representations from mixed scenarios, we propose the multi-scenario learning (MSL) to strategically train the extractor through a scenario-by-scenario manner. Different from traditional learning-based CMI methods where only one classifier is used, we employ N ($N > 1$) classifiers, corresponding to the N scenarios in the training data.[1] Note that these N classifiers have *unshared* weights, relaxing the constraint that the classifier for each OSN transmission scenario has to be the same in the single classifier case. Specifically, each of the N classifiers supervises the training data of one specific OSN transmission scenario. For instance, the first classifier supervises the data of the NoTrans. scenario, the second classifier supervises the data of the Facebook scenario, etc. Compared with the traditional way of using only one classifier to supervise the training of the extractor [6], [10], the MSL strategy could effectively alleviate the interference of training with data from different OSN transmission scenarios.

Another reason why we adopt the MSL with N classifiers is that the training complexity varies between scenarios. For example, the task of extracting camera traces is clearly easier from the NoTrans. scenario than from the Facebook scenario. Separating these scenarios with different classifiers is of advantageous in learning a shared representation among different OSN transmission scenarios, compared to the traditional methods having only one single classifier. However, the unbalanced training complexity problem may lead to inconsistent training directions for different scenarios during the backward process, resulting in gradient conflicts [18]. Some existing algorithms have been proposed to mitigate gradient conflicts through gradient normalization [19] or gradient dropping [20]. However, these algorithms are designed based on a single backward process, which may not adequately describe the conflicts between different scenarios, especially when the scenarios are similar. Therefore, to thoroughly depict and filter the conflicts between different scenarios, we further propose to

[1]In reality, there is no need to traverse all possible OSNs. Our results show that a relatively small N (e.g., $N = 4$) with appropriately selected training data, could generalize well to all the considered OSN scenarios.

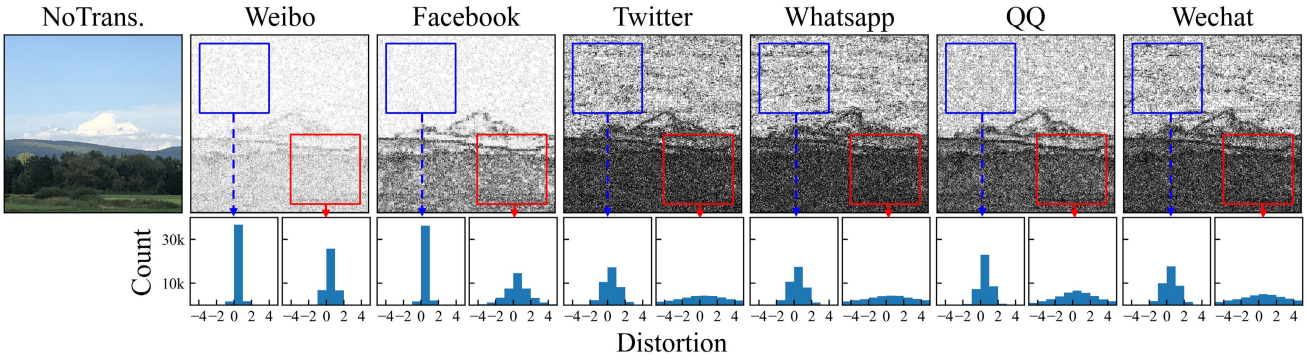

Fig. 3. Observation II: Distortion caused by OSNs in smooth and textured regions. The first row presents the distortion intensity at the pixel-level, which is normalized and multiplied by 10 for better visualization (the darker the color, the greater the distortion). The second row plots the distortions of selected smooth and textured regions, respectively.

introduce a momentum masking operation in the MSL, which generates filter masks by accumulating historical backward processes.

The effect of our proposed MSL is shown in the orange curve in Fig. 2, which is clearly better than the blue curve corresponding to the traditional training with mixed scenario.

2) Observation II: The second observation is that smooth regions in the image typically suffer less OSN processing than textured ones. As shown in Fig. 3, distortion in the sky is relatively lower than that in the trees, indicating that more camera traces in smooth regions may survive the transmission. This phenomenon is reasonable because textured regions contains more high-frequency signals, which are usually discarded by operations such as compression and scaling in OSN [21]. On the other hand, existing works [11], [22] corroborated that camera traces in textured regions are obscured by the complex signal itself. This phenomenon suggests that the trace extractor should pay more attention to the smooth regions of the image. To this end, Guera et al. [22] selected eligible regions for camera traces by estimating a global reliability map through training a CNN, while Mayer and Stamm [11] exploited a fixed entropy threshold to filter suitable image patches. However, estimating the reliability map through a separate CNN like [22] is ineffective, and the patches discarded by [11] could still carry useful information regarding camera traces. In this work, we propose the SmooThness-Aware Trace Extractor (STATE), based on a cross-attention mechanism, so as to extract camera traces from distinct regions more flexibly and effectively.

As expected and will be verified experimentally, our proposed method designed based on the aforementioned observations demonstrates satisfactory robustness and significantly exceeds the state-of-the-art algorithms, especially in various OSN transmission scenarios. As shown in Fig. 1, our method not only surpasses existing algorithms in the NoTrans. scenario, but also achieves much improved performance upon the transmission over various OSN platforms.

In summary, our major contributions are as follows:

- To the best of our knowledge, we are the first to formalize the CMI through the MSL strategy, and demonstrate that this strategy can achieve satisfactory robustness against OSN transmissions.
- We propose the STATE to flexibly and effectively learn the camera traces according to the regional smoothness.
- Our method achieves better robustness performance in comparison with state-of-the-art approaches [5], [6], [10], [11], especially in the scenario of OSN transmissions.
- We build new OSN-transmitted datasets over nine popular OSNs (Twitter, Telegram, Whatsapp, Instagram, Facebook, Weibo, QQ, Dingding, and WeChat) based on existing camera datasets `FODB` [12] and `SIHDR` [23], for not only evaluating the robustness of CMI algorithms, but also benefiting different forensic applications.

The rest of the paper is organized as follows. Section II reviews the related works on CMI and OSN. Section III presents the baseline framework, and Section IV details our proposed robust CMI through MSL and STATE. Experimental results are delivered in Section V and Section VI concludes.

II. Related Works

A. Camera Model Identification (CMI)

Many methods [1], [5], [6], [10], [11], [13], [24], [25], [26], [27], [28], [29], [30] have been proposed to characterize the camera traces contained in digital images. These methods can be roughly divided into traditional and learning-based types. Specifically, several traditional methods extracted camera traces by modeling the operations conducted during the image acquisition. One of the most well-known methods is the PRNU [1], which models noise patterns introduced by sensor imperfections. Different from PRNU with a multiplicative noise model, Thai et al. [24] developed a generalized noise model, involving more image processing operations, such as linear relation between raw pixels, non-linear effect of gamma correction, etc. Similarly, by estimating the parameters of the indispensable processes inside the camera, e.g., color interpolation and color filter array, Swaminathan et al. [25] designed the non-intrustive component forensics. Bonettini et al. [26] later analyzed different JPEG eigen-algorithms and showed that the traces left by JPEG compression can be used for CMI.

In contrast to the tedious process of designing hand-crafted features, many learning-based CMI algorithms have been proposed recently with the rapid development of CNNs. The pioneering deep learning scheme for CMI was designed by

Bondi et al. [13], whose algorithm surpassed the best one exploiting hand-crafted features [27], [28]. By introducing fine-grained labels, Bennabhaktula et al. [30] proposed a camera recognition algorithm that performs the hierarchical classification within homogeneous regions. Inspired by the PRNU, Cozzolino and Verdoliva [5] utilized a siamese network to extract camera noiseprint by simultaneously enhancing camera artifacts and suppressing high-frequency scene contents. By using the PRNU [1] algorithm to pre-extract camera model noise, Mandelli et al. [6] proposed a fast and efficient pair-wise correlation network for better practicality in analyzing large databases. Instead of explicitly characterizing camera traces, Mayer and Stamm [11] introduced a learnable algorithm to measure the similarity of two image patches, so as to determine whether two images come from the same camera model or not.

It should be noted that the existing CMI algorithms [5], [6], [10], [11] still suffer from severe performance degradation when the images under investigation undergo some lossy operations, especially in the practical OSN transmission scenarios. Therefore, it is of paramount importance to develop robust CMI method, facilitating its practical deployment in many multimedia forensic tasks.

B. Online Social Network (OSN)

Online social networks greatly simplify the transmission of multimedia data. Nowadays, there are nearly 3.78 billion daily active users of OSNs [31], and more than 3.2 billion images are shared on OSNs everyday [32]. Facebook, YouTube, Whatsapp, Instagram, and WeChat are the five most popular social platforms at present, with 2.9, 2.2, 2.0, 2.0, and 1.2 billion monthly active users, respectively [33]. However, OSNs are not forensic-friendly platforms because their inevitably conducted lossy operations may seriously affect many types of forensic algorithm [34]. For instance, Castiglione et al. [35] verified that the identification ability of PRNU [1] will deteriorate due to the noise introduced by OSN. Many forensic schemes therefore strive to improve their robustness against the negative effects of OSN [14], [15], [36], [37], [38], [39].

III. Baseline Scheme for Camera Model Identification

Before diving into our robust design of the CMI, we first introduce the architectures of the learning-based baseline scheme, which consists of two networks, namely, *extractor* and *classifier*, as shown in Fig. 4. Similar to the existing schemes [6], [10], [13], the extractor aims to extract camera traces, while the classifier supervises the training of the extractor. In our baseline scheme, the EfficientNet-B0 is adopted for the specific architecture of the extractor. The selection is based on a preliminary experiment comparing the camera trace extraction performance of different candidate architectures including ResNet [40], VGG [41], XceptionNet [42], EfficientNet [43], ViT [44] and SwinTransformer [45], etc. Regarding the classifier architecture, we simply design it as the combination of a linear layer and a SoftMax transformation.

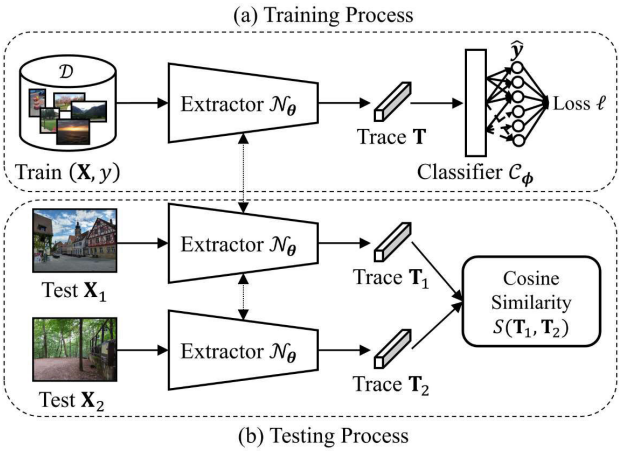

Fig. 4. Training and testing processes of our baseline CMI.

Once determining the architectures for both the extractor and the classifier networks, another crucial issue is how to train and use them at the testing stage. In Fig. 4, we illustrate both the training and testing processes of the baseline CMI scheme. In the training phase, given a dataset $\mathcal{D}$ consisting of images captured by Y different camera models, $(\mathbf{X}, y)$ denotes a pair of training data, where $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ is the input image and $y \in \{1, 2, \cdots, Y\}$ is the label of the camera. The extractor $\mathcal{N}_{\boldsymbol{\theta}}$ with trainable parameters $\boldsymbol{\theta}$ aims to extract high-level features $\mathbf{T}$ that can characterize the camera traces, i.e., $\mathbf{T} = \mathcal{N}_{\boldsymbol{\theta}}(\mathbf{X})$. To supervise the training of $\mathcal{N}_{\boldsymbol{\theta}}$, the classifier $\mathcal{C}_{\boldsymbol{\phi}}$ converts $\mathbf{T}$ into logits $\hat{\mathbf{y}} = \mathcal{C}_{\boldsymbol{\phi}}(\mathbf{T})$ for calculating the loss $\ell(\hat{\mathbf{y}}, y)$, where ℓ is the widely used cross-entropy loss given by:

$$\ell(\hat{\mathbf{y}}, y) = -\sum_{i=1}^{Y} \mathcal{I}[y = i] \cdot \log(\hat{\mathbf{y}}^{<i>}). \tag{1}$$

Here $\hat{\mathbf{y}}^{<i>}$ denotes the ith entry of $\hat{\mathbf{y}}$, and $\mathcal{I}[y = i]$ is a binary indicator function, which takes 1 if $y = i$, and 0 otherwise.

Upon the training, the well trained $\mathcal{N}_{\boldsymbol{\theta}}$ can be used in two testing cases according to [6]: 1) *open-set verification*, aiming at inferring if two testing images are captured by the same camera model or not; and 2) *close-set classification*, aiming at identifying the source camera model of a testing image among a finite pool of camera models. Noted that in both cases, the classifer $\mathcal{C}_{\boldsymbol{\phi}}$ is discarded. Specifically, given two images $\mathbf{X}_1$ and $\mathbf{X}_2$ in the former case, their traces $\mathbf{T}_1$ and $\mathbf{T}_2$ are first extracted by the trained $\mathcal{N}_{\boldsymbol{\theta}}$. Then, the probability of these traces being taken by the same camera is calculated by the cosine similarity:

$$S(\mathbf{T}_1, \mathbf{T}_2) = \cos\Big(\frac{\mathbf{T}_1}{||\mathbf{T}_1||}, \frac{\mathbf{T}_2}{||\mathbf{T}_2||}\Big). \tag{2}$$

As for the testing process of the close-set classification case, given a set of images $\mathbf{X}_i$'s with known labels $y_i \in \{1, 2, \cdots, Y\}$, $\mathcal{N}_{\boldsymbol{\theta}}$ first extracts their traces $\mathbf{T}_i$'s. Then a trace pool $\{\bar{\mathbf{T}}_i\}_{i=1}^{Y}$ can be formed by averaging the traces with the same camera model. In other words, each element of the pool represents the average trace of one specific camera model. When a testing image $\mathbf{X}_t$ comes, its predicted camera type y_t can be obtained by searching for the greatest similarity in the

TABLE I

DISTORTION CAUSED BY DIFFERENT OSNS ON THE FODB [12] DATASET. HERE, "SCALE" AND "SIZE" MEANS THE RESOLUTION REDUCTION AND THE FILE SIZE REDUCTION PERCENTAGES, RESPECTIVELY. ALSO, "JPEG QF" DENOTES THE AVERAGE QF VALUES ADOPTED

OSN	Scale (%)		Size (%)		JPEG QF	
	Mean	Std	Mean	Std	Mean	Std
Facebook	-63.48	22.84	-81.16	10.52	84.94	0.56
Twitter	-62.98	22.70	-81.82	9.55	86.53	2.36
Telegram	-84.83	12.48	-92.79	5.11	80.35	2.10
Whatsapp	-77.40	18.90	-92.06	5.74	84.24	4.03
Instagram	-85.51	11.60	-95.25	3.17	79.24	0.51
Weibo	-57.77	23.90	-60.19	22.12	94.77	1.00
QQ	-67.12	21.45	-86.93	7.78	80.00	0.00
Dingding	-84.60	12.34	-93.93	4.75	69.30	15.88
WeChat	-73.67	24.25	-84.73	17.04	82.64	11.66

trace pool, namely,

$$y_t = \underset{i}{\operatorname{argmax}}\, S(\mathbf{T}_t, \bar{\mathbf{T}}_i) \tag{3}$$

where $\mathbf{T}_t$ is the trace of the testing image extracted via $\mathcal{N}_{\boldsymbol{\theta}}$.

Although our baseline CMI can be used to extract camera traces and eventually be adopted in the aforementioned two testing cases, the performance upon lossy transmissions, e.g., various OSN transmission scenarios, could be seriously degraded. The distortions introduced by these scenarios are likely to destroy camera traces, which are inherently fragile in nature. In Table I, we briefly show the distortions caused by different OSNs on the FODB [12] dataset, in terms of the average resolution and file size reductions, and average adopted JPEG quality factors (QFs). As can be seen, the most severe distortions are caused by Dingding, resulting in 84.60% downsampling in resolution and 93.93% in file size reduction, which can also be observed from the lowest average QF value 69.3. The most friendly OSN platform among the considered ones is Weibo, leading to 57.77% downsampling in resolution and 60.19% in file size reduction. As will be clear soon, these OSN distortions would severely affect the CMI algorithms, and it is therefore of paramount importance to design a robust CMI scheme, capable of reliably extracting camera traces upon OSN transmissions.

IV. ROBUST CAMERA MODEL IDENTIFICATION

Upon having the baseline, we propose a novel method for designing a robust CMI against the transmission over various OSNs, where the key innovations are two-fold: the MSL and the STATE. As expected and will be verified experimentally, the STATE extracts more adaptive traces for distinct inputs, and the MSL strategy better supervises the training of the STATE, jointly contributing to the objective of robust extraction of camera traces.

The training process of the proposed robust CMI is illustrated in Fig. 5. Specifically, given a training image $\mathbf{X}$, we first collect its transmission variants under N scenarios. To achieve the satisfactory robustness, in this work, we define the scenarios consisting of NoTrans. (original), two types of OSN transmission: Facebook and Whatsapp, which are also considered in the VISION dataset [17]. Besides, considering the discrepancy between the training and testing scenarios, we handcraft an augmentation scenario to improve the generalization against unknown (new) OSNs, where the augmentation includes commonly-used post-processing operations such as scaling, compression, blurring, and noise addition. More analyses on the impact of scenarios, e.g., different numbers of scenario and their combinations are deferred to the ablation studies in Sec. V-G.4. For each variant, the STATE $\mathcal{N}_{\boldsymbol{\theta}}$ extracts the corresponding trace $\mathbf{T}$, where the embedded smoothness attention module guides $\mathcal{N}_{\boldsymbol{\theta}}$ to pay more attention to smoother regions in $\mathbf{X}$. Following the Observation I, N classifiers $\{\mathcal{C}_{\boldsymbol{\phi}_n}\}_{n=1}^N$ are employed to supervise the training of $\mathcal{N}_{\boldsymbol{\theta}}$, where each classifier handles the training of each individual scenario. For instance, $\mathcal{C}_{\boldsymbol{\phi}_1}$ deals with the NoTrans. scenario, $\mathcal{C}_{\boldsymbol{\phi}_2}$ deals with the Facebook scenario, etc. Next, the total loss $\sum_{n=1}^N \mathcal{L}_n$ generated on the $\{\mathcal{C}_{\boldsymbol{\phi}_n}\}_{n=1}^N$ will be fed backward to update the learnable parameters $\boldsymbol{\theta}$ associated with STATE $\mathcal{N}_{\boldsymbol{\theta}}$ and the $\{\boldsymbol{\phi}_n\}_{n=1}^N$ associated with N classifiers.

In the testing phase, the procedure is similar to that of the baseline CMI, where only the trained STATE $\mathcal{N}_{\boldsymbol{\theta}}$ is required, while the N classifiers $\{\mathcal{C}_{\boldsymbol{\phi}_n}\}_{n=1}^N$ are discarded.

In the following, we give more details regarding our proposed MSL strategy and the STATE.

A. Multi-Scenario Learning (MSL)

As aforementioned, our MSL strategy motivated by the Observation I employs multiple classifiers with unshared weights to supervise the training of multiple scenarios. An implicit problem arising with the multiple classifiers is that the training process may not be stable. This is because the training complexity varies among different scenarios, leading to the conflicts in gradient directions. To remedy this drawback, we propose to integrate a momentum masking operation in the backward process of the MSL to mitigate gradient conflicts. We now present the details on the forward and backward processes of our proposed MSL.

1) Forward Process: Let $\mathbf{X}$ be the input image, $\{\mathbf{X}_n\}_{n=1}^N$ be its N variants under the N scenarios, and $\{\mathbf{T}_n\}_{n=1}^N$ be their extracted traces by the STATE $\mathcal{N}_{\boldsymbol{\theta}}$. The details regarding $\mathcal{N}_{\boldsymbol{\theta}}$ are deferred to next subsection. Also, each classifier $\mathcal{C}_{\boldsymbol{\phi}_n}$ is employed to supervise the training process based on the training data from the nth scenario, with the following loss:

$$\mathcal{L}_n = \ell(\mathcal{C}_{\boldsymbol{\phi}_n}(\mathbf{T}_n), y), \tag{4}$$

where ℓ is the cross-entropy loss given in (1).

It is worth noting that the key difference between the loss functions (4) and (1) lies in the fact that the former involves multiple unshared classifiers, while the latter uses only one. Also, note that $\mathbf{T}_n$ and $\mathcal{C}_{\boldsymbol{\phi}_n}$ should have the same subscript n, so as to implement the MSL training scenario-by-scenario.

2) Backward Process: In the backward process, the parameters of the classifier $\mathcal{C}_{\boldsymbol{\phi}_n}$ are updated by:

$$\boldsymbol{\phi}_n = \boldsymbol{\phi}_n - r\nabla_{\boldsymbol{\phi}_n}\mathcal{L}_n, \tag{5}$$

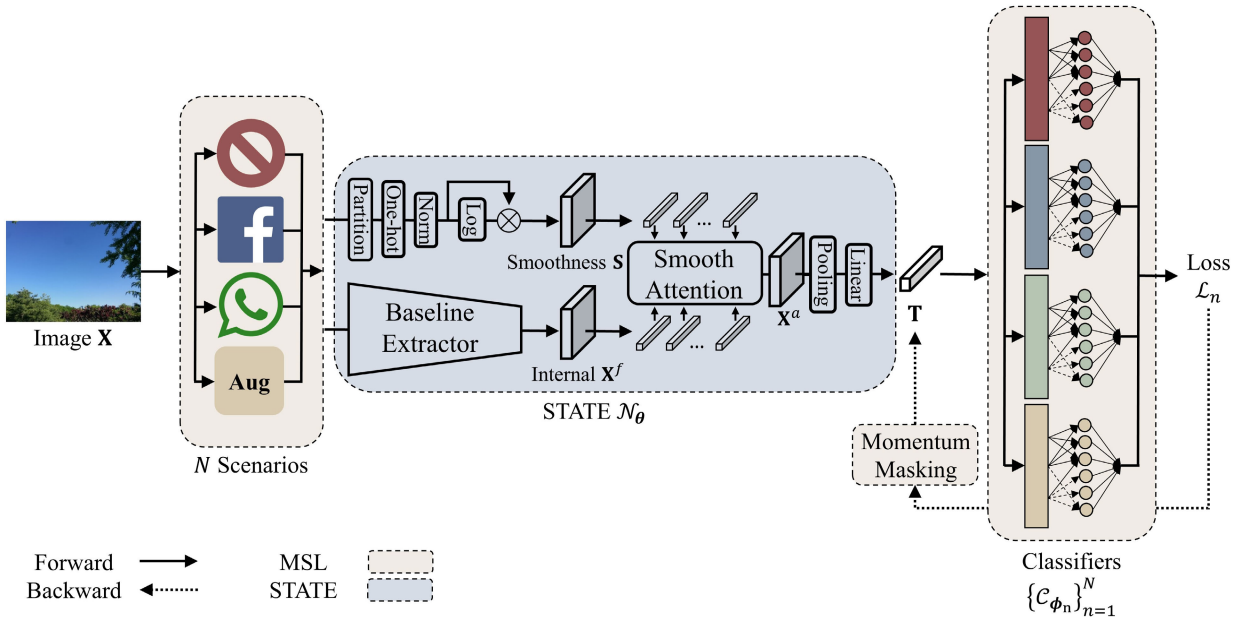

Fig. 5. The training process of our proposed robust CMI. The STATE extracts camera traces from a given image, while the MSL strategy utilizes a set of classifiers to supervise the training of the STATE on a scenario-by-scenario basis.

where r is the learning rate. Similarly, the backward update of the extractor $\mathcal{N}_{\boldsymbol{\theta}}$ can be expressed as:

$$\boldsymbol{\theta} = \boldsymbol{\theta} - r \sum_{n=1}^{N} \nabla_{\boldsymbol{\theta}} \mathcal{L}_n. \tag{6}$$

However, $\sum_{n=1}^{N} \nabla_{\boldsymbol{\theta}} \mathcal{L}_n$ here consists of N terms corresponding to N scenarios, in which inconsistent gradient directions may cause conflicts, leading to the suboptimal training of $\mathcal{N}_{\boldsymbol{\theta}}$ [18]. Also, by noting that $\mathbf{T}_n = \mathcal{N}_{\boldsymbol{\theta}}(\mathbf{X}_n)$ and using the chain rule, (6) can be rewritten as:

$$\boldsymbol{\theta} = \boldsymbol{\theta} - r \sum_{n=1}^{N} \left(\frac{\partial \mathbf{T}_n}{\partial \boldsymbol{\theta}} \right)^T \nabla_{\mathbf{T}_n} \mathcal{L}_n, \tag{7}$$

where $\partial \mathbf{T}_n / \partial \boldsymbol{\theta}$ is the Jacobian matrix of $\mathbf{T}_n$. To mitigate the impact of the gradient conflict on the update of $\mathcal{N}_{\boldsymbol{\theta}}$ in (6) or (7), one solution is to design non-conflicting gradients $\mathbf{L}_n$ as a replacement of $\nabla_{\mathbf{T}_n} \mathcal{L}_n$.

According to [20], the non-conflicting gradients $\mathbf{L}_n$ could be formed by elementwisely masking $\nabla_{\mathbf{T}_n} \mathcal{L}_n$ based on the level of consistency. Specifically,

$$\mathbf{L}_n = \mathbf{M}_n \odot \nabla_{\mathbf{T}_n} \mathcal{L}_n, \tag{8}$$

where $\odot$ denotes elementwise multiplication. Here $\mathbf{M}_n$ is a binary matrix with the same dimension as $\nabla_{\mathbf{T}_n} \mathcal{L}_n$, and is defined as:

$$\begin{aligned} \mathbf{M}_n = {} & \mathcal{I}[\mathbf{P} \succcurlyeq \mathbf{U}] \odot \mathcal{I}[\nabla_{\mathbf{T}_n} \mathcal{L}_n \succcurlyeq \mathbf{0}] \\ & + \mathcal{I}[\mathbf{P} \preccurlyeq \mathbf{U}] \odot \mathcal{I}[\nabla_{\mathbf{T}_n} \mathcal{L}_n \preccurlyeq \mathbf{0}], \end{aligned} \tag{9}$$

where $\mathcal{I}$ is the standard indicator function, $\mathbf{U}$ of the appropriate dimension represents a random matrix sampled from the uniform distribution $U(0, 1)$, and $\succcurlyeq$ ($\preccurlyeq$) is elementwise inequality. Also, $\mathbf{P}$ measures the purity (consistency) of the positive sign contained in the given gradients, which is formulated as:

$$\mathbf{P} = \frac{1}{2}\left(1 + \frac{\sum_n \mathbf{G}_n}{\sum_n |\mathbf{G}_n|}\right), \tag{10}$$

where $\mathbf{G}_n = \text{sign}(\mathbf{T}_n) \odot \nabla_{\mathbf{T}_n} \mathcal{L}_n$ consolidates gradient contributions over the batch dimension [20], and all calculations including the division and absolute value operation are conducted elementwisely.

However, $\mathbf{P}$ calculated by (10) depends on the specific gradient $\nabla_{\mathbf{T}_n} \mathcal{L}_n$ (or the input $\mathbf{X}$) at a single backward, restricting its capability of describing the gradient consistency locally. This may result in unstable training or poor local minima in some cases, such as when conflicts in a batch cancel each other out. We therefore propose to take into account the *historical* gradients through the momentum average [46], rather than only involving the gradients in the current backward. In this way, $\mathbf{P}$ could globally calculate the consistency of different scenarios, stabilizing the training updates in the stochastic gradient descent (SGD).

To apply the idea of momentum to the generation of $\mathbf{P}$, we re-define the purity at the tth backward process as:

$$\mathbf{P}^{(t)} = \frac{1}{2}\left(1 + \frac{\sum_n \mathbf{g}_n^{(t)}}{\sum_n |\mathbf{g}_n^{(t)}|}\right), \tag{11}$$

where

$$\begin{aligned} \mathbf{g}_n^{(t)} &= \mu \cdot \mathbf{g}_n^{(t-1)} + (1-\mu)\mathbf{G}_n^{(t)} \\ &= \mu^{t-1}\mathbf{g}_n^{(1)} + \sum_{i=1}^{t-2} \mu^i (1-\mu)\mathbf{G}^{(t-i)} + (1-\mu)\mathbf{G}_n^{(t)} \end{aligned} \tag{12}$$

accumulates the gradients over the previous $t-1$ historical backward processes, and $\mathbf{g}_n^{(1)}$ is initialized to $\mathbf{G}_n^{(1)}$. Here μ is the decay factor that controls the weight of recent gradients.

In practice, we empirically set $\mu = 0.95$. Clearly, when $\mu = 0$, the momentum $\mathbf{P}^{(t)}$ degenerates to the original $\mathbf{P}$.

After the momentum purity $\mathbf{P}^{(t)}$ is obtained, the mask $\mathbf{M}_n$ in (9) and the non-conflicting gradients $\mathbf{L}_n$ in (8) can be calculated accordingly by replacing $\mathbf{P}$ with $\mathbf{P}^{(t)}$. Eventually, $\mathbf{L}_n$ is used to update the parameters of $\mathcal{N}_{\boldsymbol{\theta}}$ through:

$$\boldsymbol{\theta} = \boldsymbol{\theta} - r \sum_{n=1}^{N} \left(\frac{\partial \mathbf{T}_n}{\partial \boldsymbol{\theta}} \right)^T \mathbf{L}_n. \tag{13}$$

3) MSL Training Algorithm: We summarize the whole MSL training process in Algorithm 1. More specifically, the forward process is described in lines 5∼7, while the rest lines are devoted to the backward process. In line 5, we collect the variants of inputs in N scenarios, which could also be conducted offline in advance. Then, the STATE $\mathcal{N}_{\boldsymbol{\theta}}$ extracts camera traces in line 6, and classifiers $\mathcal{C}_{\boldsymbol{\phi}_n}$'s are utilized to calculate the loss in line 7. To mitigate gradient conflicts in the loss, lines 8∼20 are mainly for generating momentum masks $\mathbf{M}_n$, which are then used in line 21 for updating $\boldsymbol{\theta}$. Eventually, the trained $\mathcal{N}_{\boldsymbol{\theta}}$ is produced in line 24.

Remark: A naive and alternative training strategy is to re-label the data in different scenarios and calculate the prediction through a single classifier. For instance, a dataset with 29 cameras and 4 scenarios can be represented as a single classification task with $29 \times 4 = 116$ categories. A potentially crucial problem is that this alternative assumes the sufficient variability among different scenarios; otherwise some categories are, to some extent, indistinguishable. However, this assumption is not always true, e.g., for the NoTrans. scenario and cropping scenario, the camera traces are more or less the same. In other words, one trace may actually correspond to multiple labels, which could lead to the instability in the training process. In our proposed MSL strategy, this dilemma can be naturally avoided through N classifiers.

We now present the details on the architecture of the extractor STATE.

Algorithm 1 MSL Training Process

Input: Number of scenarios N; training data $\mathcal{D}$; learning rate r; network $\mathcal{N}_{\boldsymbol{\theta}}$; classifiers $\{\mathcal{C}_{\boldsymbol{\phi}_n}\}_{n=1}^{N}$
Output: Trained network $\mathcal{N}_{\boldsymbol{\theta}}$

1 Initialize $t = 1$
2 Randomly initialize $\boldsymbol{\theta}$, $\{\boldsymbol{\phi}_n\}_{n=1}^{N}$
3 **for** $(\mathbf{X}, y) \subset \mathcal{D}$ **do**
4 **for** $n \in \{1, \cdots, N\}$ **do**
5 Collect $\mathbf{X}_n$ ▷ Transmission
6 $\mathbf{T}_n = \mathcal{N}_{\boldsymbol{\theta}}(\mathbf{X}_n)$
7 Calculate $\mathcal{L}_n$ ▷ Eq. (4)
8 $\mathbf{G}_n = \text{sign}(\mathbf{T}_n) \odot \nabla_{\mathbf{T}_n} \mathcal{L}_n$
9 **if** $t > 1$ **then**
10 $\mathbf{g}_n^{(t)} = \mu \cdot \mathbf{g}_n^{(t-1)} + (1-\mu)\mathbf{G}_n$ ▷ Accumulation
11 **else**
12 $\mathbf{g}_n^{(t)} = \mathbf{G}_n$
13 **end**
14 $\boldsymbol{\phi}_n = \boldsymbol{\phi}_n - r \nabla_{\boldsymbol{\phi}_n} \mathcal{L}_n$ ▷ Update $\mathcal{C}_{\boldsymbol{\phi}_n}$
15 **end**
16 $\mathbf{P}^{(t)} = \frac{1}{2}\left(1 + \frac{\sum_n \mathbf{g}_n^{(t)}}{\sum_n |\mathbf{g}_n^{(t)}|}\right)$
17 $\mathbf{U} \sim \text{Uniform}(0, 1)$
18 **for** $n \in \{1, \cdots, N\}$ **do**
19 $\mathbf{M}_n = \mathcal{I}[\mathbf{P} \succcurlyeq \mathbf{U}] \odot \mathcal{I}[\nabla_{\mathbf{T}_n} \mathcal{L}_n \succcurlyeq \mathbf{0}] + \mathcal{I}[\mathbf{P} \preccurlyeq \mathbf{U}] \odot \mathcal{I}[\nabla_{\mathbf{T}_n} \mathcal{L}_n \preccurlyeq \mathbf{0}]$
  $\mathbf{L}_n = \mathbf{M}_n \odot \nabla_{\mathbf{T}_n} \mathcal{L}_n$
20 **end**
21 $\boldsymbol{\theta} = \boldsymbol{\theta} - r \sum_{n=1}^{N} (\partial \mathbf{T}_n / \partial \boldsymbol{\theta})^T \mathbf{L}_n$ ▷ Update $\mathcal{N}_{\boldsymbol{\theta}}$
22 $t = t + 1$
23 **end**
24 Output $\mathcal{N}_{\boldsymbol{\theta}}$

B. SmooTh-Aware Trace Extractor (STATE)

STATE aims to perform attentive camera trace extraction from a given image according to its local smoothness. In this work, we use the well-known Shannon entropy [47] to represent the smoothness of an image block. Obviously, a smaller entropy value indicates a smoother region with less activities, and vice versa. In our proposed STATE, we utilize the cross-attention [48] layer to implement the attentive extraction, where the smoothness matrix is transformed into an attention map and serves as a guidance. We would like to emphasize that our proposed STATE explicitly applies the smoothness prior according to the Observation II and thus significantly differs from the trivial adoption of the cross-attention layer.

The procedure of STATE is illustrated in Fig. 5. Concretely, we first partition the input image of size $H \times W$ into non-overlapping blocks $\{\mathbf{R}_i\}$ every $\frac{H}{\hat{H}}$ rows and $\frac{W}{\hat{W}}$ columns, resulting in $\hat{H} \times \hat{W}$ blocks in total. For the ith block $\mathbf{R}_i$, we calculate the associated smoothness indicator, i.e., the Shannon entropy E_i, as:

$$E_i = -\sum_{v=0}^{255} p_v \log(p_v), \tag{14}$$

where

$$p_v = \frac{\hat{H}\hat{W}}{HW} \sum_{h=0}^{H/\hat{H}} \sum_{w=0}^{W/\hat{W}} \mathcal{I}[\mathbf{R}_i^{<h,w>} = v] \tag{15}$$

is the probability of the pixel value within $\mathbf{R}_i$ being v. The smoothness matrix $\mathbf{S} \in \mathbb{R}^{\hat{H} \times \hat{W}}$ can then be obtained by grouping and reshaping $\{E_i\}$ into the dimension $\hat{H} \times \hat{W}$. In the actual implementation, we grayscale the input image to reduce the complexity and use one-hot encoding for the efficient batch processing.

Upon having the smoothness matrix $\mathbf{S}$, we first extract internal feature $\mathbf{X}^f$ by the baseline extractor, and then perform cross-attention on the $\mathbf{X}^f$ according to $\mathbf{S}$. Particularly, $\mathbf{X}^f$ and $\mathbf{S}$ are tokenized by respectively flattening into $\hat{\mathbf{X}}^f$ and $\hat{\mathbf{S}}$ with size $\hat{H}\hat{W} \times \hat{C}$, where $\hat{C}$ is the channel number after the tokenization. Then K-head attention is performed on the

flattened $\hat{\mathbf{X}}^f$ and $\hat{\mathbf{S}}$ by calculating the attention on every $\hat{C}/K$ channels, resulting in the K-head features $\{\hat{\mathbf{X}}_k^a\}_{k=1}^K$, where

$$\hat{\mathbf{X}}_k^a = \text{Attention}(\hat{\mathbf{S}}\mathbf{Q}_k, \hat{\mathbf{X}}^f\mathbf{K}_k, \hat{\mathbf{X}}^f\mathbf{V}_k), k = 1, \ldots, K, \quad (16)$$

and $\mathbf{Q}_k, \mathbf{K}_k, \mathbf{V}_k \in \mathbb{R}^{\hat{C}^2/K}$ are *query*, *key*, and *value* matrices of the kth projection for the cross-attention function [48]. Here, the cross-attention is performed by:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{SoftMax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{\hat{C}/K}})\mathbf{V}. \quad (17)$$

Next, the concatenated outputs of all heads $\{\hat{\mathbf{X}}_k^a\}_{k=1}^K$ are linearly projected to generate the flattened attention $\hat{\mathbf{X}}^a$ via:

$$\mathbf{X}^a = \text{MLP}(\text{Concat}(\hat{\mathbf{X}}_1^a, \hat{\mathbf{X}}_2^a, \ldots, \hat{\mathbf{X}}_K^a)), \quad (18)$$

in which MLP$(\cdot)$ represents a MLP with GELU activation [49]. Finally, the refined attention feature $\mathbf{X}^a$ is obtained by reshaping it back into $\hat{H} \times \hat{W} \times \hat{C}$ resolution.

Considering the dimensional redundancy in $\mathbf{X}^a$, we follow the tradition [6], [10] and employ a global average pooling and a linear mapping to encode $\mathbf{X}^a$ into a low-dimensional space $\mathbb{R}^d$, so as to obtain the final camera traces $\mathbf{T}$.

V. Experimental Results

In this section, we comprehensively evaluate the performance of the proposed CMI method in terms of open-set verification, close-set classification, robustness against OSN transmissions, post-processing operations, and re-transmissions. Further, systematic ablation studies are given and analyzed. Before presenting the detailed results, let us first introduce the experimental settings.

A. Experimental Settings

1) Training Datasets: For the training of the proposed method, similar to [5], [6], we use the VISION [17] dataset that includes 29 different camera models. Following [5], [6], [11], and [10], we merge the images from different devices but of the same model to avoid the ambiguity. It should be noted that this dataset contains Facebook and Whatsapp transmitted variants, which can be conveniently used as the training data of the N scenarios (see Fig. 5).

2) Testing Datasets: To better simulate the practical situation and evaluate the generalization, we adopt the FODB [12] and SIHDR [23] as cross testing datasets, having **NO** overlap with the training data. The FODB dataset comprises 25 models and 27 devices, whereas the SIHDR dataset consists of 21 models and 23 devices, respectively. Within these two datasets, there are two models in each that have an additional device. Unless otherwise specified, we label the images of different devices but of the same model in FODB (and SIHDR) dataset based on their model category, similar to the process conducted in VISION.

3) Online Social Networks: Although the FODB dataset itself contains five OSN transmitted versions (Twitter, Telegram, Whatsapp, Instagram, and Facebook), we further extend both the FODB and SIHDR datasets to nine popular OSN transmission scenarios, including Twitter, Telegram, Whatsapp, Instagram, Facebook, Weibo, QQ, Dingding, and WeChat. This enable us to evaluate the robustness of CMI algorithms more extensively against the practical OSN transmissions nowadays. The extended datasets and the details regarding the operating system and OSN platform versions are made available at **https://github.com/HighwayWu/CameraTraceOSN**.

4) Comparative Methods: To show the superior performance of our proposed CMI method, we adopt four state-of-the-art algorithms as competitors, namely, Kuzin [10], ForSim [11], NoiPri [5], and PCN [6].

5) Implementation Details: During the training, the number of scenarios adopted in the MSL strategy is set to 4, including the NoTrans., Facebook and Whatsapp (provided by the VISION dataset itself), and an additional handcrafted augmentation scenario. The reason for introducing the augmentation scenario is to further improve the generalization ability of the network to unseen scenarios. More specifically, the augmentation is formed by randomly mixing 0∼50% downsampling, JPEG compression with QFs 70∼100, Gaussian blurring with kernels 3∼5, Gaussian noise addition with variances 3∼10.

We implement our method using the PyTorch deep learning framework, where the Adam [50] with default parameters is adopted as the optimizer. The learning rate is initialized to 1e-4 and halved if the validation loss fails to decrease for 5 epochs until the convergence. In the training process, all the input images are randomly cropped into 512×512 patches. The internal feature $\mathbf{X}^f$, the smoothness matrix $\mathbf{S}$, and the cross-attention feature $\mathbf{X}^a$ share the same feature dimensions of $32 \times 32 \times 320$. The dimension d (see Section IV-B) of the extracted trace $\mathbf{T}$ is set to 256. Based on the observation that increasing the resolutions of the testing images would improve the performance [6], [12], we set the testing size as 1536×1536. To facilitate the reproducibility of our results, our code is available at **https://github.com/HighwayWu/CameraTraceOSN**.

B. Evaluation of Open-Set Verification Task

Open-set verification task aims to infer whether two given images are captured by the same camera model. To this end, we randomly choose 25 images from each camera model, and then form 5,000 positive and 5,000 negative pairs for each testing dataset. As the verification task is essentially a binary classification problem, we adopt the widely used Area Under the receiver operating characteristic Curve (AUC) as the criterion (higher the better) for evaluating the performance.

As can be observed from Table II, when the images are not transmitted through an OSN, all the existing methods perform similarly well, with the AUC ranging from 87.32% to 89.38%, while our method is slightly better with a 1.50% AUC improvement over the best existing scheme. However, when images are transmitted through OSNs, the verification performance of all existing methods degrades significantly. Taking Twitter as an example, the AUC values of Kuzin

TABLE II

OPEN-SET VERIFICATION TASK ON THE FODB AND SIHDR DATASETS BY USING AUC AS THE CRITERION. FOR EACH COLUMN WITH THE SAME OSN TRANSMISSION SCENARIO, THE BEST VALUE IS **BOLD**, AND THE SECOND-BEST IS UNDERLINED

Methods	FODB [12]										Mean
	NoTrans.	Twitter	Telegram	Whatsapp	Instagram	Facebook	Weibo	QQ	Dingding	WeChat	
Kuzin [10]	.8851	.7768	.6693	.6614	.6514	.6927	.7709	.6573	.6024	.5977	.6965
ForSim [11]	.8938	.6241	.5616	.5940	.5628	.6168	.7375	.5688	.5424	.6964	.6398
NoiPri [5]	.8923	.7099	.6825	.6806	.6305	.6629	.6696	.6169	.6085	.5958	.6750
PCN [6]	.8732	.7509	.5852	.6682	.5975	.7062	.8259	.6617	.5836	.5295	.6782
Ours	**.9088**	**.8899**	**.8574**	**.8229**	**.7836**	**.8712**	**.8677**	**.8768**	**.8084**	**.8081**	**.8495**
Methods	SIHDR [23]										Mean
	NoTrans.	Twitter	Telegram	Whatsapp	Instagram	Facebook	Weibo	QQ	Dingding	WeChat	
Kuzin [10]	.8545	.8451	.7320	.7656	.7517	.7859	.8290	.7535	.7283	.7046	.7750
ForSim [11]	.9592	.8685	.6457	.6511	.6100	.6668	.8374	.6230	.6448	.7056	.7212
NoiPri [5]	.9482	.8703	.8423	.8493	.6517	.7692	.9223	.8188	.8233	.8798	.8375
PCN [6]	.9576	.9267	.7389	.9148	.7649	.8809	.9512	.8270	.7307	.6378	.8331
Ours	**.9804**	**.9750**	**.9394**	**.9500**	**.9308**	**.9719**	**.9594**	**.9660**	**.9359**	**.9145**	**.9523**

TABLE III

CLOSE-SET CLASSIFICATION TASK ON THE SIHDR DATASET BY USING PRECISION (PRC), RECALL (RCL) AND F1 AS CRITERIA (HIGHER THE BETTER)

Methods	Metric	NoTrans.	Twitter	Telegram	Whatsapp	Instagram	Facebook	Weibo	QQ	Dingding	WeChat	Mean
Kuzin [10]	PRC	.6446	.5667	.4207	.5082	.5202	.4852	.5231	.5101	.4623	.5069	.5148
NoiPri [5]		.8604	.7497	.7133	.6890	.5221	.6798	.7030	.5938	.6111	.6759	.6798
PCN [6]		.9294	.8937	.5299	.7617	.6665	.8256	.8987	.7898	.5913	.6248	.7511
Ours		**.9708**	**.9435**	**.8758**	**.9141**	**.8488**	**.9401**	**.9317**	**.9349**	**.8120**	**.7998**	**.8971**
Kuzin [10]	RCL	.6079	.5475	.4250	.4588	.4421	.4947	.4800	.4816	.4500	.4313	.4819
NoiPri [5]		.8225	.7176	.4833	.6206	.3853	.5538	.6687	.4969	.5031	.7750	.6027
PCN [6]		.8150	.8111	.3280	.4876	.3726	.6381	.8350	.5300	.4133	.2722	.5503
Ours		**.9643**	**.9190**	**.8476**	**.8929**	**.8381**	**.9333**	**.9167**	**.9214**	**.7762**	**.7775**	**.8787**
Kuzin [10]	F1	.5232	.5012	.3240	.3563	.3999	.4365	.4327	.4276	.3174	.3283	.4047
NoiPri [5]		.7587	.6111	.3104	.5505	.3058	.4724	.5470	.4080	.4197	.6137	.4997
PCN [6]		.8216	.7660	.2922	.5071	.3282	.6446	.8372	.4933	.3750	.1986	.5264
Ours		**.9638**	**.9133**	**.8427**	**.8902**	**.8306**	**.9306**	**.9097**	**.9188**	**.7639**	**.7512**	**.8715**

[10], ForSim [11], NoiPri [5], and PCN [6] drop by 10.83%, 26.97%, 18.24%, and 12.23%, respectively, compared to the NoTrans. scenario. Such severe performance degradation is probably because the lossy operations performed by OSNs greatly eliminate the original camera traces, especially those that are not robust. Conversely, by utilizing our proposed MSL strategy and smoothness attention, our proposed method can explore highly robust camera traces, resulting in only 1.89% AUC reduction in the Twitter scenario. For other OSN transmission scenarios in Table II, our method still exhibits satisfactory robustness, outperforming the second-best method by 15.30% in AUC on average.

As for the results on the SIHDR dataset in the lower part of Table II, we can observe similar phenomenon as that of the case in the FODB. For instance, regarding ForSim, the most severe performance drop is found in QQ platform with 33.62% AUC degradation, and the most friendly platform is Weibo, resulting in 12.18% AUC reduction. In contrast, our method leads to a very desirable robustness, surpassing the AUC of the second place by 11.48% on average. It should also be noted that all the methods considered here perform better on SIHDR than FODB, primarily due to the fact that SIHDR has fewer camera categories and greater variability among categories. It should be noted that, in each column of Table II, all the selected pairs have both images passing through the same respective OSN. A more challenging experiment where images within a pair are transmitted through *different* OSNs are deferred to Section V-H for further elaboration.

C. Evaluation of Close-Set Classification Task

Close-set classification task targets to predict the source of a given query image among a finite pool of camera models, where the "ground-truth" trace representing each camera model is extracted from a set of predefined images (i.e., anchor set) [6]. Typically, the "ground-truth" traces can be obtained by utilizing the PRNU [1] algorithm, or simply by averaging the deep features. Similar to the open-set evaluation, we randomly sample 45 images from each camera category, among which 25 images are used as the anchor set, and the remaining ones serve as the query images. For each query image, the predicted camera label will be granted to the one in the pool with the highest similarity to its extracted trace.

To evaluate the classification task, we display the confusion matrix in Fig. 6, along with the corresponding precision (PRC), recall (RCL), and F1 scores in Table III. Formally, PRC and RCL are defined by:

$$\mathrm{PRC} = \frac{1}{Y}\sum_{y=1}^{Y}\frac{\mathrm{TP}_y}{\mathrm{TP_y} + \mathrm{FP_y}}, \tag{19}$$

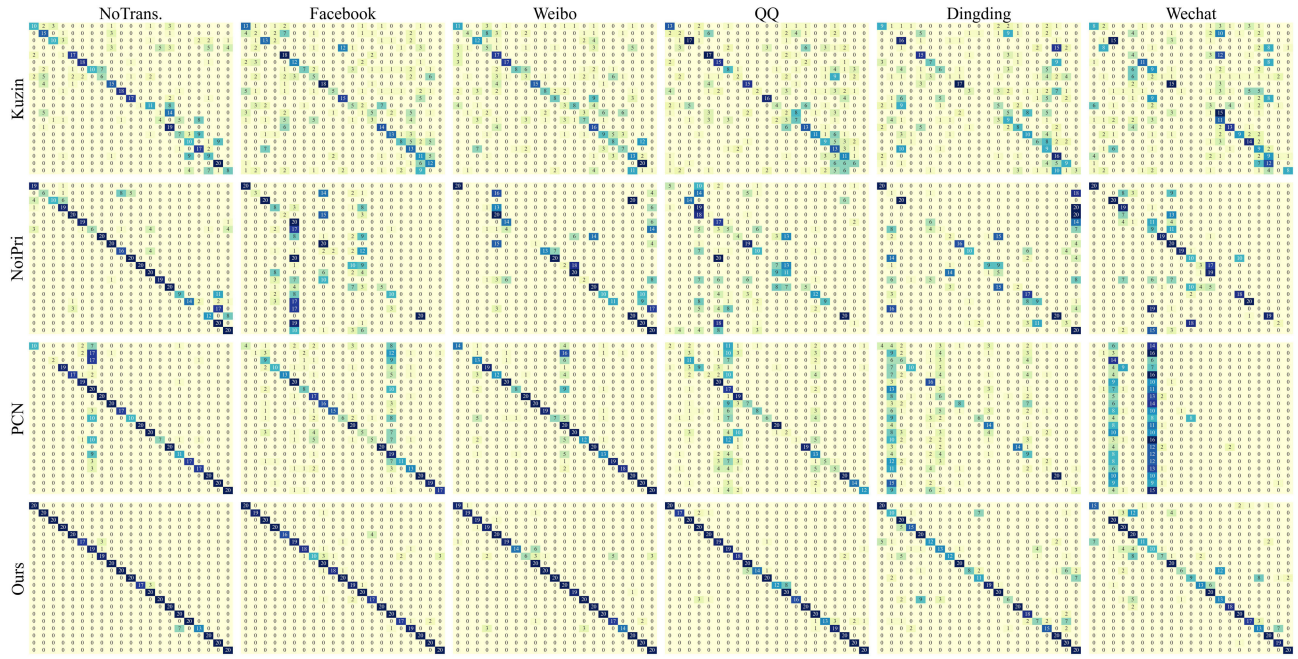

Fig. 6. Confusion matrix of the close-set classification task on the SIHDR dataset. For each matrix, the vertical and horizontal axes represent the actual label and the predicted one, respectively. Here, each row (column) stands for a unique camera model, e.g., the first, second, and last rows correspond to GioneeS55, Huaiwei-P8, and iPhone 5S, respectively. Please refer to our code website for the specific information of the camera models.

$$\mathrm{RCL} = \frac{1}{Y}\sum_{y=1}^{Y}\frac{\mathrm{TP}_y}{\mathrm{TP}_y + \mathrm{FN}_y}, \tag{20}$$

where TP_y, FP_y, and FN_y represent True Positive, False Positive, and False Negative for a given class y, respectively. Then the macro-averaged F1 score can be calculated as follows:

$$\mathrm{F1} = \frac{1}{y}\sum_{y=1}^{Y}\frac{2\times\mathrm{TP}_y}{2\times\mathrm{TP}_y + \mathrm{FP}_y + \mathrm{FN}_y}. \tag{21}$$

As can be seen from Table III, in the case of NoTrans., our method achieves rather satisfactory result (96.38% F1), which is significantly better than the two strong competitors PCN [6] and NoiPri [5] algorithms by big margins (the gaps are 14.22% and 20.51% respectively in F1). When the images are transmitted through OSNs, all existing methods are seriously affected. For instance, the F1 reductions of PCN [6] are respectively 32.83% and 62.30% on QQ and WeChat, which are deemed to be huge. This implies that it is difficult for PCN to extract discriminative camera traces under OSN disturbances. This phenomenon can also be easily observed from the confusion matrix given in Fig. 6 (see the last column of the third row). In contrast, our method is generally robust to *all* OSN transmissions, resulting in an average F1 score of 87.15%. Compared with the competing algorithms Kuzin [10], NoiPri [5], and PCN [6], we achieve overwhelming advantages in terms of F1 by 46.68%, 37.18%, and 34.51%, respectively. Here both the query and anchor images in each column of Table III have passed through the same respective OSN.

It should be noted that we here omit the results of ForSim [11]. This is because the similarity network used in the ForSim can only extract features that measure the similarity of a pair given inputs, and cannot capture features specific to a type of camera trace. Consequently, ForSim is unable to extract corresponding features for the anchor set.

D. Evaluation of Open-Set Classification Task

In addition to the previous open-set verification and closed-set classification tasks, we now consider the open-set classification task, which not only involves the detection of whether a given image is known, but also includes the further classification to its specific label. Specifically, assume that the cameras in SIHDR dataset are known, and 25 images per camera are employed as the anchor set. We select 20 new images per camera model in SIHDR and determine whether they can be correctly classified as known and to the correct camera models. Additionally, we select 20 images per camera model in FODB and assess whether they can be classified as unknown, i.e., none of the known (suspected) models in SIHDR. To avoid the ambiguity, we exclude the camera models that are common to both the SIHDR and FODB datasets. Obviously, the determination process requires the establishment of a threshold for accepting or rejecting a testing example. In particular, we directly reject an example when its maximum similarity with known camera models, $\mathbf{S}_{max}$, is below a given threshold δ, and the acceptance occurs only when $\mathbf{S}_{max} > \delta$ and the example is predicted correctly.

Experimental results regarding a range of thresholds from 0 to 1 are presented in Fig. 7, using accuracy and F1 as criteria. Due to the differences in the methods used to extract camera traces, competing methods achieve their respective optimal performance at different threshold values. Specifically, Kuzin [10] gets the optimal accuracy of 79.26% and F1 score of 68.87%, while NoiPri [5] attains the best accuracy of 72.30% and F1 score of 65.58%. Due to PCN's

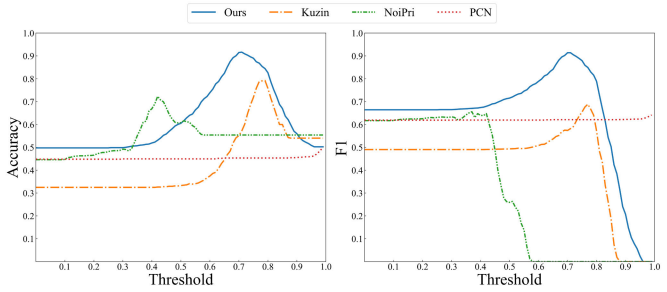

Fig. 7. Open-set classification performance at different threshold values.

inclination towards generating relatively high similarity scores for both positive and negative pairs, it becomes less sensitive to threshold variations, resulting in the absence of peaks in Fig. 7. Specifically, the highest accuracy and F1 score achieved by PCN are only 50.07% and 64.15%, respectively. This suggests that PCN is not well-suited for the open-set classification task. In comparison, our proposed algorithm leads to a remarkable accuracy of 91.63% and F1 score of 91.40%, surpassing the second-best by a substantial margin of +12.37% in accuracy and +22.53% in F1.

E. Evaluation of Post-Processing

Although the focus of this paper is on the robust design of CMI scheme against various OSN transmission scenarios, we now show that our design also naturally leads to satisfactory robustness to commonly-used post-processing as well. Specifically, the post-processing operations considered include JPEG compression with QFs ranging from 70 to 95, linear resizing of factors from 10% to 50%, Gaussian blurring with kernel sizes [3, 5, 7], and Gaussian noise addition with variances [3, 5, 7, 9]. There are a total of 18 different post-processing operations, with JPEG compression, resizing, blurring, and noise addition having 6, 5, 3, and 4 variants respectively. These operations are applied to the SIHDR dataset, which contains 929 images. Therefore, a total of $18 \times 929 = 16,722$ images are generated. For each image, only one type of post-processing attack is applied. Given that the OSN already encompasses compound attack scenarios, our focus here primarily lies in countering a single post-processing attack. The comparison results are presented in Fig. 8. As can be seen, although ForSim [11] and NoiPri [5] perform well in NoTrans. scenario, they are struggling to maintain stable performance when encountering strong post-processing like Gaussian blurring and Gaussian noise addition. Kuzin [10] is relatively stable under JPEG compression and resizing; but the performance drops quickly in the cases of Gaussian blurring and noise additions. In contrast, PCN [6] and ours present much better robustness against these post-processing operations; our method is still superior to PCN for all the considered cases, especially in Gaussian blurring with large kernel sizes.

F. Evaluation of Re- and Cross-Transmissions

In practice, re-transmission and/or cross-transmission over multiple OSN platforms (i.e., images are downloaded and re-uploaded to the same or different OSNs) are very common. We now briefly discuss the problem of evaluating the robustness of different CMI algorithms under the re-transmission and cross-transmission cases. Specifically, we consider the transmissions via Facebook followed by Facebook/QQ, and Whatsapp followed by Whatsapp/QQ. The results are tabulated in Table IV. It can be observed that the second round of transmission, either re-transmission or cross-transmission, affects much less than the first round of transmission. Our method exhibits rather strong robustness against both rounds of OSN transmissions, while the competing algorithms give much inferior performance. For instance, in the transmission via Whatsapp followed by Whatsapp/QQ, our method suffers only 2.21%/1.99% AUC drops, while PCN has a more serious performance degradation, reaching 10.26%/9.36% AUC loss.

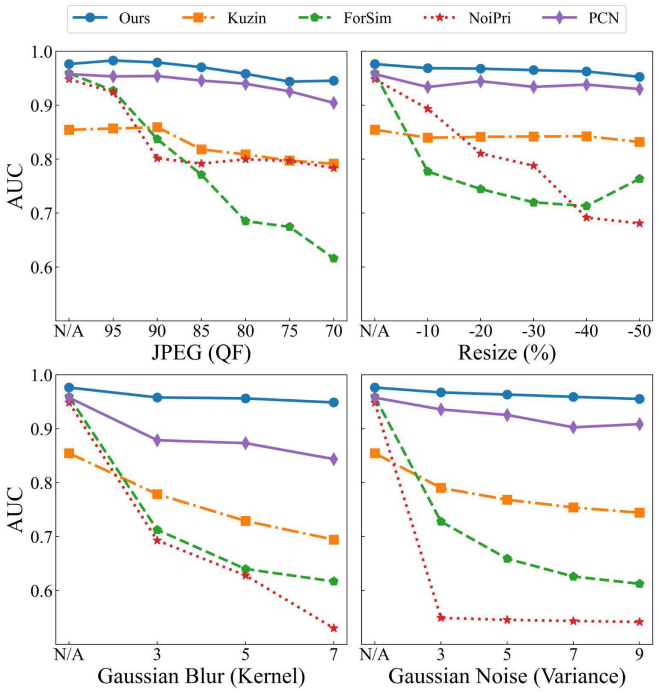

Fig. 8. Robustness evaluations against JPEG compression, linear resizing, Gaussian blurring, and Gaussian noise addition.

G. Ablation Studies

In this subsection, we conduct the ablation studies of our proposed method by analyzing how each component contributes to the extraction of robust camera traces. Specifically, we first prohibit the utilization of both the smoothness attention and the MSL strategy, resulting in the baseline performance. We then incorporate different variants of the attention module and the MSL strategy into the baseline, evaluating the additional performance gains brought by them. The comparative results are given in Table V, which, due to page limit, only includes the open-set verification results regarding the NoTrans., Facebook, and Whatsapp scenarios over the SIHDR [23] dataset.

1) Adoption of Smoothness Attention: In the second row of Table V, we give the results with the attention mechanism. Here, in addition to our smoothness attention, we also evaluate

TABLE IV

OPEN-SET VERIFICATION TASK AGAINST RE- AND CROSS- TRANSMISSIONS ON THE SIHDR DATASET. CRITERION IS AUC

Transmission 1st	Transmission 2nd	Kuzin	ForSim	NoiPri	PCN	Ours
-	-	.8545	.9592	.9482	.9576	**.9804**
Facebook	-	.7859	.6668	.7692	.8809	**.9719**
Facebook	Facebook	.7850	.6656	.7697	.8806	**.9716**
Facebook	QQ	.7488	.6138	.8175	.7791	**.9627**
Whatsapp	-	.7656	.6511	.8493	.9148	**.9500**
Whatsapp	Whatsapp	.7355	.6205	.8456	.8122	**.9279**
Whatsapp	QQ	.7349	.6182	.8389	.8212	**.9301**
Mean		.7730	.6834	.8274	.8749	**.9598**

TABLE V

ABLATION STUDIES REGARDING THE SMOOTHNESS ATTENTION, MSL, AND THE STRUCTURE OF BASELINE EXTRACTOR. CRITERION IS AUC

Brand	Detail	Transmission of SIHDR [23] NoTrans.	Facebook	Whatsapp	Mean
Baseline (EffNet-B0)		.8524	.7975	.8017	.8172
+ Attention	Reliability [22]	.8544	.7991	.8023	.8186
	Filter [11]	.8621	.8060	.8080	.8254
	Smoothness (Ours)	.8929	.8268	.8157	.8451
+ Learning	Unweighted	.9463	.8759	.8584	.8935
	GradNorm [19]	.9491	.9046	.8990	.9176
	GradDrop [20]	.9431	.9091	.9225	.9249
	MSL (Ours)	.9506	.9371	.9354	.9410
+ Smooth + MSL		.9804	.9719	.9500	.9674
Baseline (MobileFormer-24M)		.8151	.7942	.8131	.8075
+ Smooth + MSL		.9638	.9457	.9286	.9460

the performance of variant attentions including the entropy filter [11] or reliability map [22]. As can be seen, all three attention modules do improve the performance, by guiding the baseline extractor to pay more attention to smooth regions in the image. Specifically, the improvement brought by the reliability map [22] is limited (only 0.14% gains), mainly because of two factors: 1) it estimates the reliability of a given image patch-by-patch (locally), which lacks interactivity from a global perspective; and 2) it requires pre-training and is therefore difficult to be trained end-to-end, which limits the optimization of the extractor. As for the entropy filter [11] approach, it needs to manually adjust a threshold to filter regions with high entropy, which inevitably discards some valuable information and is cumbersome in practice. As a result, the increment brought by the entropy filter is minor, being only 0.82%. In contrast, our cross-attention based smoothness attention module is not only capable of customizing its smoothness automatically for different inputs, but can also be readily incorporated into an end-to-end training, resulting in a 2.79% performance gain. In Fig. 9, we also visualize the extractor attention with or without the proposed smoothness attention, for a more intuitive understanding. Clearly, our smoothness attention module is effective in guiding the feature extraction to pay more attention to smooth regions.

2) Adoption of MSL: To analyze the contribution of different learning strategies, we present the corresponding ablation results in the third row of Table V. Noted that the baseline extractor in the first row utilizes a single classifier for the optimization, while all results in the third row adopt multiple classifiers, belonging to the category of MSL. In addition to our MSL strategy, we also include variants by using GradNorm [19] or GradDrop [20] for obtaining the non-conflicting gradients. It can be concluded that the MSL with multiple classifier does greatly improve the overall performance; even a simple unweighted strategy can bring a gain of 7.63%. The incorporation of GradNorm [19] or GradDrop [20] is shown to achieve larger (10.04% and 10.77%, respectively,) improvements. Owing to momentum masking operation that explicitly utilizes the gradients produced by the historical backward processes, our proposed MSL achieves an even better performance gain to 12.38%, compared with the baseline. Finally, by jointly using the smoothness attention and the MSL strategy, our robust CMI can greatly outperform the baseline, leading to a total performance improvement of 15.02%.

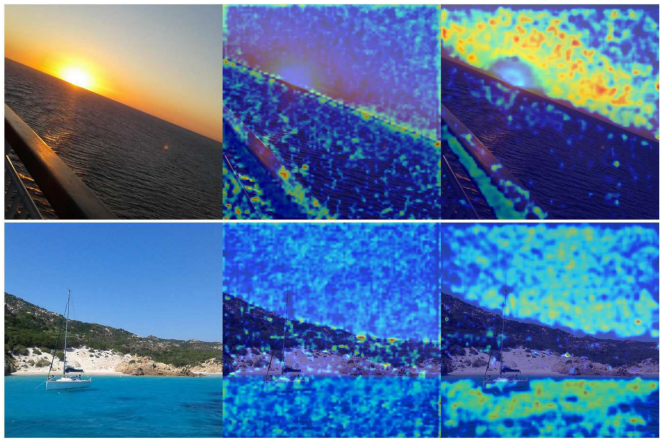

Fig. 9. Visualization results by using GradCam [52] for the cases of without or with our smoothness attention module. Warmer colors indicate more attention.

TABLE VI

EFFECT OF THE N SCENARIOS IN THE MSL STRATEGY. NOTRANS., FB, WA, AUG REPRESENT THE TRANSMISSION SCENARIO OF ORIGINAL, FACEBOOK, WHATSAPP, AND AUGMENTATION, RESPECTIVELY

N Scenarios	Transmission of SIHDR [23] NoTrans.	Facebook	Whatsapp	Twitter	Instagram	QQ	Mean
NoTrans.	.9122	.8244	.7986	.9025	.8322	.8477	.8608
FB	.8223	.8770	.8489	.9092	.8570	.8492	.8718
WA	.8342	.8630	.8724	.8844	.8539	.8413	.8599
NoTrans. + FB	.9616	.9479	.8967	.9597	.9201	.9341	.9380
NoTrans. + WA	.9706	.9204	.9175	.9585	.9008	.9236	.9276
NoTrans. + Aug	.9788	.9464	.9084	.9637	.9151	.9350	.9379
NoTrans. + FB + WA	.9604	.9530	.9306	.9602	.9271	.9484	.9452
NoTrans. + FB + WA + Aug	.9804	.9719	.9500	.9750	.9308	.9660	.9624

3) Selection of Baseline Extractor: As mentioned in Section III, our proposed robust CMI is versatile, where the baseline extractor (EfficientNet-B0) can be flexibly replaced by other networks. To this end, we adopt another state-of-the-art network, MobileFormer [51], as the baseline to demonstrate that the robustness of the extractor can also be greatly improved by applying our robust designs. As shown in the last row of Table V, the robustness of the MobileFormer baseline has been well strengthened, e.g., the average AUC is increased by 13.85%.

4) Effect of N Scenarios in MSL: In the design of the MSL strategy, an interesting question that may arise is how many scenarios are needed to achieve the desired robustness. We therefore conduct additional experiments to analyze the

TABLE VII

OPEN-SET VERIFICATION TASK ON THE SIHDR DATASET BY USING AUC AS THE CRITERION. ON EACH COLUMN, MULTIPLE OSNS ARE INVOLVED. NT, FB, WA, IN, TE, TW, WC, WB, DD ARE SHORT NAMES FOR NOTRANS., FACEBOOK, WHATSAPP, INSTAGRAM, TELEGRAM, TWITTER, WECHAT, WEIBO, AND DINGDING, RESPECTIVELY

Methods	SIHDR [23] with Multiple OSNs					Mean
	NT, FB	NT, WA	NT, FB, WA	NT, FB, WA IN, TE, TW	NT, FB, WA, TW, TE IN, WC, WB, QQ, DD	
Kuzin [10]	.7333	.6976	.6866	.6791	.6562	.6906
ForSim [11]	.6436	.5326	.5317	.5563	.5220	.5572
NoiPri [5]	.6971	.6377	.6142	.5379	.5287	.6031
PCN [6]	.7168	.5827	.5447	.5203	.5146	.5758
Ours	**.8027**	**.7858**	**.7729**	**.7548**	**.7488**	**.7730**

TABLE VIII

DEVICE IDENTIFICATION EVALUATION. CRITERION IS AUC

Methods	FODB [12]		SIHDR [23]		Mean
	Same	Diff.	Same	Diff.	
Kuzin [10]	.8513	.8478	.9404	.9033	.8857
ForSim [11]	.9519	.9261	.9393	.8931	.9276
NoiPri [5]	.8666	.8628	.8907	.8315	.8629
PCN [6]	.9371	.9109	.9503	.9115	.9275
Ours	**.9947**	**.9825**	**.9858**	**.9491**	**.9780**

effect of the N scenarios in the MSL, and the results are reported in Table VI. As can be seen, when $N = 1$, the robustness of the extractor is far from satisfactory. The underlying reason is two-fold: 1) the MSL contains only one single classifier when $N = 1$, thus degenerating into a normal training process; and 2) the singularity of the scenario could lead to severe overfitting to the specific scenario, and result in poor generalization performance. When the number of considered scenarios increases, the aforementioned dilemmas can be well alleviated. Even when we only use both original and Facebook scenarios ($N = 2$), we observe substantial performance enhancement, achieving 7.72% AUC improvement on average. In addition, considering the discrepancy between the training and testing process, we introduce a data augmentation scenario to further enhance the generalization of the extractor against unknown (new) OSNs. According to the results from the combined scenario "NoTrans. + Aug", it can be concluded that this augmentation scenario is effective, possibly because these augmented operations overlap, to some extent, with the ones employed by other OSNs. Finally, the results of the last two rows show that further increasing the number of scenarios to $N = 3$ and $N = 4$ could continuously improve the performance. We also have tried more combinations with a larger N; but the additional performance gains are very marginal, at the cost of significantly increased complexity. Therefore, in our scheme, we adopt the combination of four scenarios, namely, "NoTrans. + FB + WA + Aug".

H. Evaluation of Challenging Cross-OSN Scenarios

Recall that in the aforementioned experiments, images within a pair are transmitted through the same OSN (consistent OSN scenario). In this subsection, we conduct more challenging experiments where the images are sourced from different OSNs (inconsistent OSN scenario), such as one from NoTrans. and another from Facebook (marked as "NT, FB" below). In these scenarios, the performance outcomes of each algorithm are depicted in Table VII. For the columns containing more than two OSNs, the images in a pair come from two randomly selected OSNs. It is evident that our proposed algorithm still outperforms existing methods by big margins, with an average AUC improvement of 8.24% over the second-ranked Kuzin [10]. However, unsurprisingly, the performance of all algorithms exhibits severe degradation, compared to the consistent OSN scenario. Due to the mixture of the camera traces and OSN degradation, it is rather challenging to separate the camera traces out and subsequently perform the camera model identification. In fact, in some extreme cases, it is possible that camera-A, affected by OSN-A interference, and camera-B, affected by OSN-B interference, have identical accumulated effect, thereby misleading camera model identification algorithms. One potential solution is to perform the blind separation to distinguish the camera traces from the OSN interference. However, the blind separation itself is an extremely challenging task. In this work, we focus on the robust extraction of camera traces in the presence of consistent OSN interference. In the future, we will continue to explore and investigate robust camera model identification algorithms in more challenging inconsistent OSN scenarios.

I. Evaluation of Device Identification

While the main focus of this paper is camera model identification, we also attempt to involve some experiments about camera *device* identification, i.e., evaluating the algorithms' ability to identify images of the same model but different devices. Experimental results are shown in Table VIII, where the columns specified by "Same" refer to the AUC results where all images in positive pairs originate from the same device. Similarly, the columns with "Diff." correspond to the cases that the images in positive pairs come from the same model but different devices. As can be observed, our proposed method is still capable of accurately classifying images from different devices but of the same model, surpassing the second-ranked ForSim [11] by 5.04% in AUC metric. Additionally, the overall performance of the algorithms tends to be slightly better in the "Same" condition, compared to the cases of "Diff." condition. These results suggest that

different devices of the same camera model still possess certain distinctive camera traces.

VI. Conclusion

In this paper, we investigate the problem of designing robust CMI over OSN shared images. Based on two key observations, we propose a robust CMI scheme by explicitly exploiting smoothness-based cross-attention and MSL strategy. Extensive comparative experiments with several state-of-the-art methods demonstrate the superiority of our method, especially in the scenarios of various OSN transmissions. Our robust design could also shed some lights on other forensic problems such as OSN-resistant watermarking, robust forgery detection, etc.

References

[1] J. Luka, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, Jun. 2006.

[2] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim, "A holistic approach to cross-channel image noise modeling and its application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1683–1691.

[3] C. Chen, Z. Xiong, X. Liu, and F. Wu, "Camera trace erasing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2947–2956.

[4] G. Holst, "CCD arrays, cameras, and displays," *Int. Soc. Opt. Eng.*, vol. 1, no. 1, pp. 113–115, 1996.

[5] D. Cozzolino and L. Verdoliva, "Noiseprint: A CNN-based camera model fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 144–159, 2020.

[6] S. Mandelli, D. Cozzolino, P. Bestagini, L. Verdoliva, and S. Tubaro, "CNN-based fast source device identification," *IEEE Signal Process. Lett.*, vol. 27, pp. 1285–1289, 2020.

[7] M. Kharrazi, H. T. Sencar, and N. Memon, "Blind source camera identification," in *Proc. Int. Conf. Image Process. (ICIP)*, Oct. 2004, pp. 709–712.

[8] R. Caldelli, I. Amerini, and C. T. Li, "PRNU-based image classification of origin social network with CNN," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 1357–1361.

[9] J. You, Y. Li, J. Zhou, Z. Hua, W. Sun, and X. Li, "A transformer based approach for image manipulation chain detection," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 3510–3517.

[10] A. Kuzin, A. Fattakhov, I. Kibardin, V. I. Iglovikov, and R. Dautov, "Camera model identification using convolutional neural networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 3107–3110.

[11] O. Mayer and M. C. Stamm, "Forensic similarity for digital images," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1331–1346, 2020.

[12] B. Hadwiger and C. Riess, "The Forchheim image database for camera identification in the wild," in *Proc. Int. Conf. Pattern Recognit.*, Jan. 2021, pp. 500–515.

[13] L. Bondi, L. Baroffio, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps toward camera model identification with convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 259–263, Mar. 2017.

[14] W. Sun, J. Zhou, R. Lyu, and S. Zhu, "Processing-aware privacy-preserving photo sharing over online social networks," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 581–585.

[15] W. Sun, J. Zhou, Y. Li, M. Cheung, and J. She, "Robust high-capacity watermarking over online social network shared images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 1208–1221, Mar. 2021.

[16] W. Sun, J. Zhou, L. Dong, J. Tian, and J. Liu, "Optimal pre-filtering for improving Facebook shared images," *IEEE Trans. Image Process.*, vol. 30, pp. 6292–6306, 2021.

[17] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, and A. Piva, "VISION: A video and image dataset for source identification," *EURASIP J. Inf. Secur.*, vol. 2017, no. 1, pp. 1–16, Dec. 2017.

[18] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-task learning for dense prediction tasks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3614–3633, Jul. 2022.

[19] Z. Chen, V. Badrinarayanan, C. Lee, and A. Rabinovich, "GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2018, pp. 794–803.

[20] Z. Chen et al., "Just pick a sign: Optimizing deep multitask models with gradient sign dropout," in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 2039–2050.

[21] W. Luo, J. Huang, and G. Qiu, "JPEG error analysis and its applications to digital image forensics," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 480–491, Sep. 2010.

[22] D. Güera, F. Zhu, S. K. Yarlagadda, S. Tubaro, P. Bestagini, and E. J. Delp, "Reliability map estimation for CNN-based camera model attribution," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 964–973.

[23] O. Shaya, P. Yang, R. Ni, Y. Zhao, and A. Piva, "A new dataset for source identification of high dynamic range images," *Sensors*, vol. 18, no. 11, p. 3801, Nov. 2018.

[24] T. H. Thai, F. Retraint, and R. Cogranne, "Camera model identification based on the generalized noise model in natural images," *Digit. Signal Process.*, vol. 48, pp. 285–297, Jan. 2016.

[25] A. Swaminathan, M. Wu, and K. J. R. Liu, "Nonintrusive component forensics of visual sensors using output images," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 1, pp. 91–106, Mar. 2007.

[26] N. Bonettini, L. Bondi, P. Bestagini, and S. Tubaro, "JPEG implementation forensics based on eigen-algorithms," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2018, pp. 1–7.

[27] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Evaluation of residual-based local features for camera model identification," in *Proc. Int. Conf. Image Anal. Process.*, 2015, pp. 11–18.

[28] C. Chen and M. C. Stamm, "Camera model identification framework using an ensemble of demosaicing features," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Nov. 2015, pp. 1–6.

[29] X. Liao, J. Chen, and J. Chen, "Image source identification with known post-processed based on convolutional neural network," *Signal Process. Image Commun.*, vol. 99, Nov. 2021, Art. no. 116438.

[30] G. S. Bennabhaktula, E. Alegre, D. Karastoyanova, and G. Azzopardi, "Camera model identification based on forensic traces extracted from homogeneous patches," *Expert Syst. Appl.*, vol. 206, Nov. 2022, Art. no. 117769.

[31] (2021). *10 Social Media Statistics You Need to Know in 2021*. [Online]. Available: https://www.oberlo.com/blog/social-media-marketing-statistics

[32] *126 Amazing Social Media Statistics and Facts*. Accessed: Sep. 28, 2023. [Online]. Available: https://www.brandwatch.com/blog/amazing-social-media-statistics-and-facts/

[33] *20 Top Social Media Sites to Consider for Your Brand in 2022*. Accessed: Sep. 28, 2023. [Online]. Available: https://buffer.com/library/social-media-sites/

[34] A. Castiglione, G. Cattaneo, and A. De Santis, "A forensic analysis of images on online social networks," in *Proc. 3rd Int. Conf. Intell. Netw. Collaborative Syst.*, Nov. 2011, pp. 679–684.

[35] A. Castiglione, G. Cattaneo, M. Cembalo, and U. Ferraro Petrillo, "Experimentations with source camera identification and online social networks," *J. Ambient Intell. Humanized Comput.*, vol. 4, no. 2, pp. 265–274, Apr. 2013.

[36] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Blind PRNU-based image clustering for source identification," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 9, pp. 2197–2211, Sep. 2017.

[37] R. Rouhi, F. Bertini, D. Montesi, X. Lin, Y. Quan, and C.-T. Li, "Hybrid clustering of shared images on social networks for digital forensics," *IEEE Access*, vol. 7, pp. 87288–87302, 2019.

[38] F. Bertini, R. Sharma, and D. Montesi, "Are social networks watermarking us or are we (unawarely) watermarking ourself?" *J. Imag.*, vol. 8, no. 5, p. 132, May 2022.

[39] H. Wu, J. Zhou, J. Tian, J. Liu, and Y. Qiao, "Robust image forgery detection against transmission over online social networks," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 443–456, 2022.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[42] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[43] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
[44] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representat.*, 2021, pp. 1–21.
[45] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.
[46] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Comput. Math. Math. Phys.*, vol. 4, no. 5, pp. 1–17, Jan. 1964.
[47] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.
[48] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.
[49] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.
[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[51] Y. Chen et al., "Mobile-former: Bridging MobileNet and transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5260–5269.
[52] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

Haiwei Wu (Student Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from the University of Macau, Macau, China, in 2018, 2020, and 2023, respectively. His research interests include multimedia security, image processing, and machine learning.

Jiantao Zhou (Senior Member, IEEE) received the B.Eng. degree from the Department of Electronic Engineering, Dalian University of Technology, in 2002, the M.Phil. degree from the Department of Radio Engineering, Southeast University, in 2005, and the Ph.D. degree from the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, in 2009. He held various research positions with the University of Illinois at Urbana–Champaign, The Hong Kong University of Science and Technology, and McMaster University. He is currently a Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. His research interests include multimedia security and forensics, multimedia signal processing, artificial intelligence, and big data. He holds four granted U.S. patents and two granted Chinese patents. He has coauthored two papers that received the Best Paper Award at the IEEE Pacific-Rim Conference on Multimedia in 2007 and the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo in 2016. He is serving as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, and IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING.

Xinyu Zhang received the B.S. degree in computer science from Zhejiang University, Zhejiang, China, in 2021. He is currently pursuing the M.S. degree with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China. His research interests include multimodal machine learning, deep learning, and low-level computer vision.

Jinyu Tian (Member, IEEE) received the B.S. and M.S. degrees in mathematics from the College of Mathematics and Statistics, Chongqing University, Chongqing, China, in 2014 and 2017, respectively, and the Ph.D. degree from the Faculty of Science and Technology, University of Macau, Macau, China, in 2022. He is currently an Assistant Professor with the School of Computer Science and Engineering, Macau University of Science and Technology. His research interests include deep learning, subspace learning, and adversarial machine learning.

Weiwei Sun received the B.S. degree from the College of Electronics and Information Engineering, Shenyang University, Shenyang, China, in 2012, the M.S. degree from the Department of Computer and Information Science, University of Macau, Macau, in 2015, and the Ph.D. degree in computer science from the State Key Laboratory of Internet of Things for Smart City, University of Macau, in 2020. He is currently a Researcher with the Alibaba Group. His research interests include the multimedia signal processing, online social networks, and multimedia security and forensics.