Rethinking Image Forgery Detection via Contrastive Learning and Unsupervised Clustering

Haiwei Wu   Yiming Chen   Jiantao Zhou[*]

State Key Laboratory of Internet of Things for Smart City
Department of Computer and Information Science
University of Macau, Macau, People's Republic of China
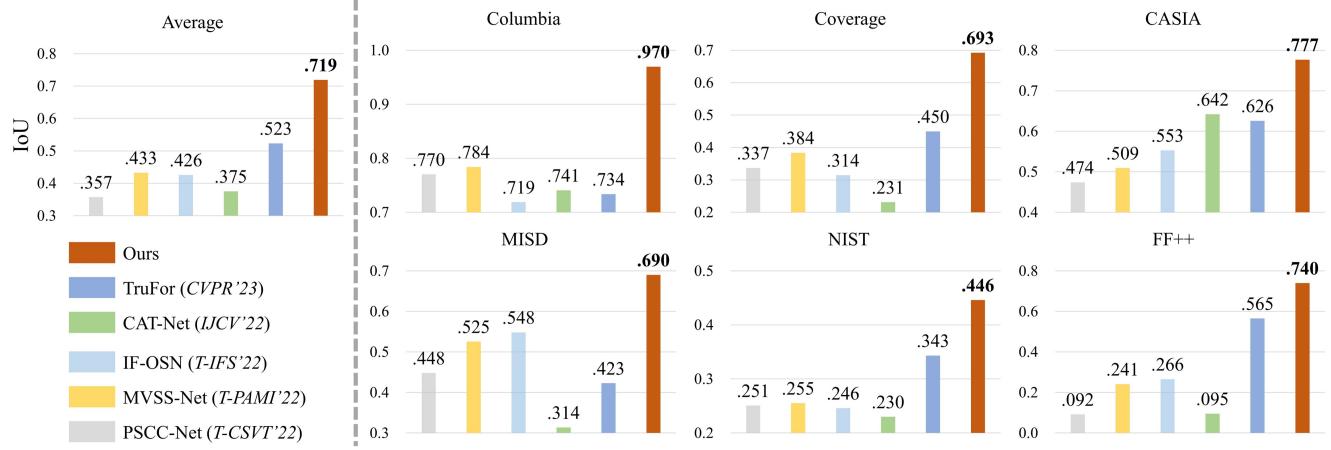
{yc07912, yc17486, jtzhou}@um.edu.mo

Figure 1: Our method significantly outperforms several state-of-the-art competing algorithms over six *cross-testing* datasets.

Abstract

*Image forgery detection aims to detect and locate forged regions in an image. Most existing forgery detection algorithms formulate classification problems to classify pixels into forged or pristine. However, the definition of forged and pristine pixels is only relative within one single image, e.g., a forged region in image A is actually a pristine one in its source image B (splicing forgery). Such a relative definition has been severely overlooked by existing methods, which unnecessarily mix forged (pristine) regions across different images into the same category. To resolve this dilemma, we propose the FOrensic ContrAstive cLustering (FOCAL) method, a novel, simple yet very effective paradigm based on contrastive learning and unsupervised clustering for the image forgery detection. Specifically, FOCAL 1) utilizes pixel-level contrastive learning to supervise the high-level forensic feature extraction in an image-by-image manner, explicitly reflecting the above relative definition; 2) employs an on-the-fly unsupervised clustering algorithm (instead of a trained one) to cluster the learned features into forged/pristine categories, further suppressing the cross-image influence from training data; and 3) allows to further boost the detection performance via simple feature-level concatenation without the need of retraining. Extensive experimental results over six public testing datasets demonstrate that our proposed FOCAL **significantly** outperforms the state-of-the-art competing algorithms by big margins: +24.3% on Coverage, +18.6% on Columbia, +17.5% on FF++, +14.2% on MISD, +13.5% on CASIA and +10.3% on NIST in terms of IoU. The paradigm of FOCAL could bring fresh insights and serve as a novel benchmark for the image forgery detection task. The code is available at https://github.com/HighwayWu/FOCAL.*

1. Introduction

The continuous advancement and widespread availability of image editing tools such as Photoshop and Meitu have

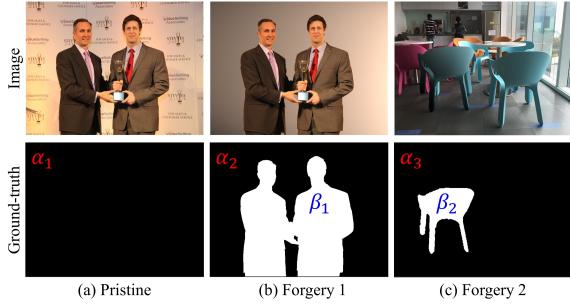Figure 2: First row: pristine and forged images. Second row: forgery masks, where pristine (α_1, α_2 and α_3) and forged (β_1 and β_2) regions are labeled black and white.

led to very convenient manipulations of digital images without much domain knowledge. The authenticity of images has thus attracted great attention recently, as maliciously manipulated (forged) images could bring serious negative effects in various fields such as rumor spreading, economic fraud, acquisition of illegal economic benefits, etc.

Numerous forensic methods [6, 8, 11, 15, 17, 22, 25, 28, 31, 32, 45, 50, 42, 2] (and references therein) have been developed to detect and localize forged regions in images, among which the deep learning based schemes offer better performance than the ones relying on hand-crafted features. Some forensic methods are dedicated to detecting specific types of forgery, such as splicing [19], copy-move [26], and inpainting [49], while more powerful and practical solutions are for detecting complex and mixed types of forgery, even accompanied with transmission degradation and various post-processing operations [8, 25, 28, 50].

In general, these existing learning-based image forgery detection methods formulate two-class classification problems to classify pixels into forged or pristine. It should be pointed out that the definition of forged and pristine pixels is only *relative* within one single image. For instance, pixels associated with the two persons in Fig. 2 (a) are pristine, while the same pixels are forged in Fig. 2 (b). Unfortunately, such a relative definition has been severely overlooked by existing classification-based forgery detection methods, which unnecessarily mix forged (pristine) regions across different images into the same category. In fact, the regions α_1, α_2, and α_3 in Fig. 2 do not necessarily have similar forensic features, though they belong to the same pristine category (similarly for β_1 and β_2). As a result, a classifier could be misled when seeing the same set of pixels are labeled as forged and pristine unfavorably, leading to unstable training and inferior detection performance.

Rethinking the relative definition of forged and pristine pixels inspires us to re-formulate the previously prevailing classification problem, into a new paradigm with contrastive

learning and unsupervised clustering. Specifically, we in this work propose the FOrensic ContrAstive cLustering (FOCAL) method, a novel, simple yet effective paradigm for image forgery detection. FOCAL utilizes pixel-level contrastive learning to supervise the high-level forensic feature extraction in an image-by-image manner, explicitly exploiting the above relative definition. The ground-truth forgery mask naturally offers the pixel-level discrimination of positive and negative categories, enabling our pixel-level contrastive learning. In addition, another unique characteristic of our contrastive learning is the image-by-image supervision, which could effectively avoid the mutual influence of features across different images in a batch. Further, FOCAL employs an on-the-fly unsupervised clustering algorithm to cluster the learned features into forged/pristine categories, further avoiding the cross-image interference from the training data. Note that here the adopted clustering module does not involve any trainable parameters and hence does not participate in the training process. It is also shown that further performance improvement can be achieved via direct feature-level fusion without the need of retraining.

Extensive experimental results over six public testing datasets demonstrate that our proposed FOCAL *significantly* outperforms the state-of-the-art competing algorithms [8, 25, 28, 50, 15] by big margins: +24.3% on `Coverage`, +18.6% on `Columbia`, +17.5% on `FF++`, +14.2% on `MISD`, +13.5% on `CASIA` and +10.3% on `NIST` in terms of IoU. The paradigm of FOCAL could bring fresh insights and serve as a novel benchmark for the image forgery detection task. Our major contributions can be summarized as follows:

- We rethink the inherent limitations of classification-based image forgery detection paradigm, from the perspective of relative definition of forged/pristine pixels.

- We design FOCAL, a novel, simple yet effective paradigm based on contrastive learning and unsupervised clustering for image forgery detection.

- The proposed FOCAL significantly outperforms several state-of-the-art image forgery detection methods over six (cross-domain) datasets with average gains being 19.6% in IoU and 10.4% in F1.

2. Related Works on Image Forgery Detection

Classification-based image forgery detection with deep learning has achieved the state-of-the-art performance [8, 25, 28, 50]. CAT-Net [25] localizes forged regions through classifying DCT coefficients. PSCC-Net [28] utilizes multi-scale features for forgery detection. Dong *et al*. (2023) introduced MVSS-Net to jointly extract forged features by multi-view learning. Wu *et al*. (2022) designed a robust
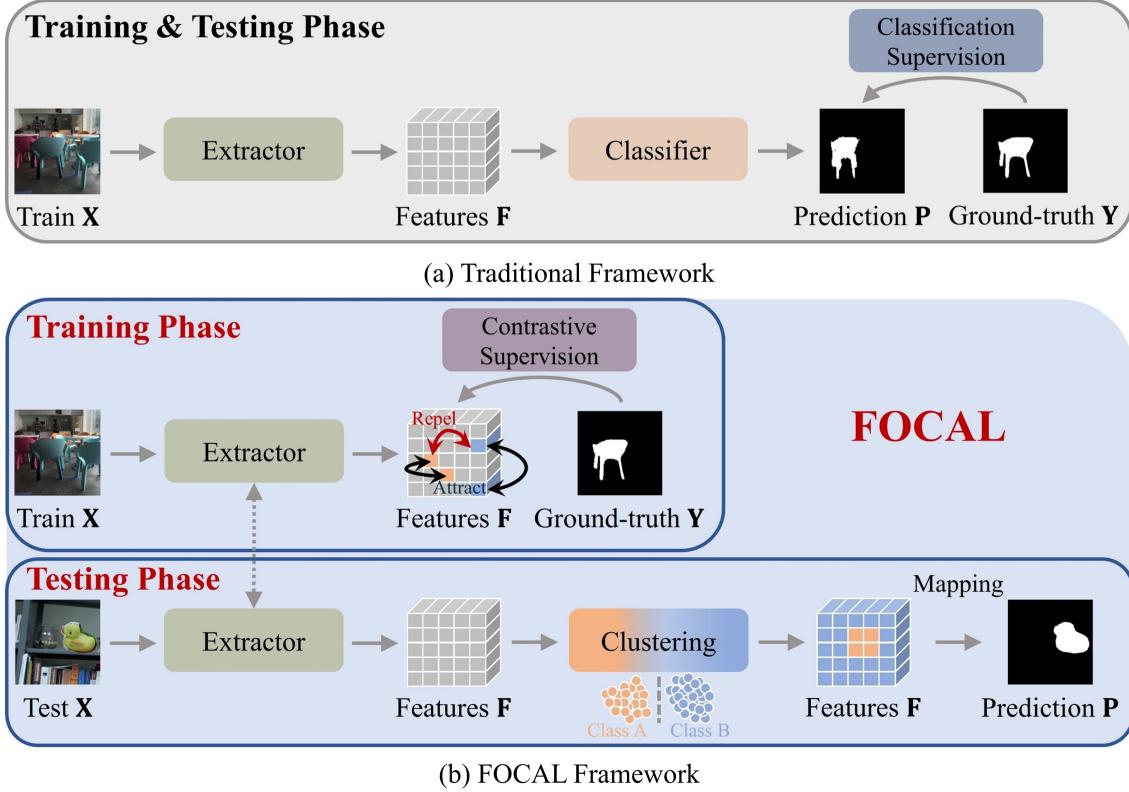
(a) Traditional Framework

(b) FOCAL Framework

Figure 3: (a) Traditional classification-based forgery detection framework; (b) Our proposed FOCAL framework, which utilizes contrastive learning to supervise the training phase, while employing an unsupervised clustering algorithm in the testing phase.

training framework based on adversarial noise modeling for image forgery detection over online social networks. Recently, Guillaro *et al*. (2023) presented TruFor which combines both RGB image and a learned noise-sensitive fingerprint to extract forensic clues. Noticing the limitations of the widely-used cross-entropy loss, some recent works also involve contrastive loss to assist the network training for image forgery detection [6, 15, 32, 47, 54, 56].

There are only a few methods trying to detect forgery from the perspective of clustering [4, 31, 33, 36, 39, 55, 59, 60], though the performance is much inferior to that of the classification-based ones. This type of method mainly uses a simple clustering algorithm to categorize the image blocks (pixels) into forged and pristine, where various noise features, *e.g*., image noise level [55], camera noise [4], JPEG quantization noise [33], were adopted.

Although the aforementioned image forgery detection methods have achieved reasonably good results, their design principles are completely different from our proposed FOCAL in the following aspects: 1) classification-based approaches ignore the relative definition of forged and pristine pixels, and thereby do not take advantage of the unsupervised clustering; 2) those approaches involving clustering

almost all work with hand-crafted features, which cannot well represent the forensic traces and are difficult to be generalized to unseen forgery types.

3. FOCAL for Image Forgery Detection

Before diving into the details of our FOCAL, we introduce the general framework of the traditional classification-based image forgery detection method, which consists of two neural networks, namely, *extractor* and *classifier*, as shown in Fig. 3 (a). Given an input $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$, the extractor first extracts discriminative feature $\mathbf{F} \in \mathbb{R}^{\hat{H} \times \hat{W} \times \hat{C}}$, based on which the classifier generates a predicted binary forgery mask $\mathbf{P} \in \mathbb{R}^{\hat{H} \times \hat{W}}$. To optimize the network, the cross-entropy loss $\mathcal{L}_c(\mathbf{P}, \mathbf{Y})$ is usually employed, where $\mathbf{Y} \in \{0, 1\}^{\hat{H} \times \hat{W}}$ is the ground-truth forgery mask (1's and 0's for forged and pristine pixels, respectively). Instead of using the classification-based approach, we build a *contrastive-clustering* framework FOCAL (see Fig. 3 (b)) for image forgery detection, explicitly exploiting the relative definition of forged/pristine pixels within an image. We now give the details of the FOCAL training via contrastive learning and FOCAL testing via unsupervised clustering.

3.1. FOCAL Training via Contrastive Learning

The training procedure of FOCAL is illustrated in the upper part of Fig. 3 (b). Once we extract high-level features $\mathbf{F}$ from a given input $\mathbf{X}$, we directly supervise $\mathbf{F}$ through pixel-level contrastive learning. The ground-truth forgery mask $\mathbf{Y}$ naturally offers us the indexes of the positive and negative categories, enabling an effective pixel-level contrastive learning. As will be clearer soon, FOCAL's contrastive learning is supervised in an image-by-image manner, which is highly different from the existing algorithms [19, 6, 15, 54, 56] that perform supervision on the entire forward mini-batch.

Specifically, we adopt an improved InfoNCE loss [16, 35] to implement the contrastive learning in FOCAL. We first construct a dictionary by performing a flattening operation $f(\cdot) : \mathbb{R}^{\hat{H} \times \hat{W} \times \hat{C}} \to \mathbb{R}^{\hat{H}\hat{W} \times \hat{C}}$ on features $\mathbf{F}$, namely,

$$f(\mathbf{F}) \to \{q, k_1^+, k_2^+, \cdots, k_J^+, k_1^-, k_2^-, \cdots, k_K^-\}. \quad (1)$$

where $\{q, k_1^+, k_2^+, \cdots, k_J^+, k_1^-, k_2^-, \cdots, k_K^-\}$ is defined as the dictionary and q is an encoded query. We let $\{q, k_1^+, k_2^+, \cdots, k_J^+\}$ represent the features belonging to pristine regions (indexed by 0's in $\mathbf{Y}$), while $\{k_1^-, k_2^-, \cdots, k_K^-\}$ stands for those of forged regions (indexed by 1's in $\mathbf{Y}$). In the image forgery detection task, the forged or pristine regions usually cover an area with more than 1 pixel (feature), which means that the number of positive keys J in the dictionary is much larger than 1 as well. Then the improved InfoNCE loss tailored to the image forgery task can be computed as

$$\mathcal{L}_{InfoNCE++} = -\log \frac{\frac{1}{J} \sum_{j \in [\![1,J]\!]} \exp(q \cdot k_j^+/\tau)}{\sum_{i \in [\![1,K]\!]} \exp(q \cdot k_i^-/\tau)}, \quad (2)$$

where τ is a temperature hyper-parameter [51]. Note that in the original InfoNCE loss [16, 35], there is only a single positive key in the dictionary that q matches. In our improved InfoNCE loss (2), we involve all the positive keys in each loss calculation by taking the expectation of the dot product of q with a set of $\{k_j^+\}$'s. This would facilitate the optimization process.

It should be emphasized that the supervision in the training phase is directly conducted between the ground-truth forgery mask $\mathbf{Y}$ and the extracted feature $\mathbf{F}$, while no predicted forgery mask is generated. Furthermore, for each image in the forward mini-batch, $\mathcal{L}_{InfoNCE++}$ is calculated in an image-by-image manner (one-by-one), rather than over the entire batch, and is then summed up to calculate the overall loss. To be more concrete, given a mini-batch features $\{\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_B\}$, the overall contrastive loss $\mathcal{L}_{ct}$ is calculated by

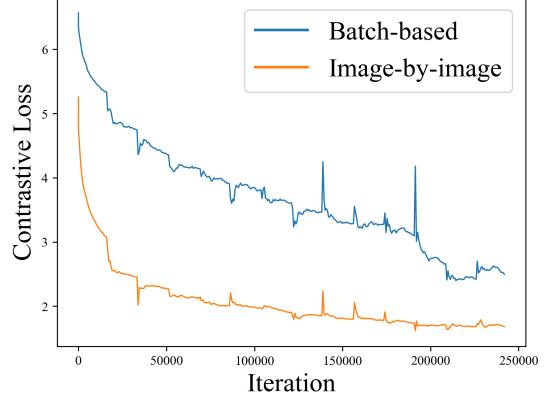$$\mathcal{L}_{ct} = \frac{1}{B} \sum_{b=1}^{B} (\mathcal{L}_{InfoNCE++}(\mathbf{F}_b)). \quad (3)$$

Figure 4: Overall contrastive loss curves of the traditional batch-based and our image-by-image one given in (3).

Note that in the above (3), the mini-batch features are *not* merged to compute an overall $\mathcal{L}_{InfoNCE++}$, avoiding the cross-image influence from the training data. This total loss designed under the guidance of relative definition of forged/pristine pixels is vastly different from those in [5, 15, 16, 32], where loss computation is conducted at the batch-level. To further justify the rationality of (3), we plot the contrastive loss curves of the traditional batch-based and our image-by-image one in Fig. 4. It can be clearly seen that the image-by-image design of the loss function (orange line) not only leads to much faster convergence, but also makes the optimization much more stable. Particularly, the high-amplitude impulses detected in the blue line indicate that there might be serious conflicts in the associated batch of images, *e.g.*, a situation similar to the case of Fig. 2 (a) and (b), where conflicting labels are presented.

Eventually, the well-trained extractor will be used in the FOCAL testing phase. As expected and will be verified experimentally, our pixel-level contrastive learning with image-by-image overall loss design significantly improves the image forgery detection performance.

3.2. FOCAL Testing via Unsupervised Clustering

We now are ready to present the details on the FOCAL testing phase. The crucial issue is how to map the extracted features into a predicted forgery mask. Compared to traditional frameworks using trained classifiers (see Fig. 3 (a)), we propose to employ an unsupervised online-learning algorithm (see the bottom half of Fig. 3 (b)). As aforementioned, the definition of forged and pristine pixels is only relative within one single image, and can be hardly generalized across different images. This explains why the previous classification-based approaches do not offer satisfactory detection results, as the classifier trained from training data may not be able to infer the unseen testing data.
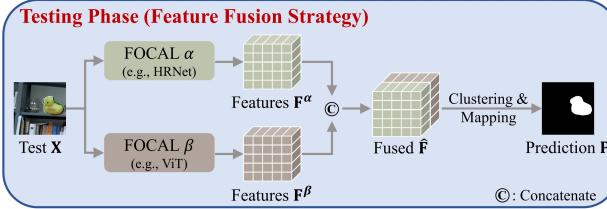
Figure 5: Feature-level fusion for boosting the detection performance of FOCAL. Retraining is not needed.

Therefore, it would be a wiser solution to map the features of different images to the final forgery mask *separately*. To this end, we adopt an on-the-fly clustering algorithm. Specifically, we employ HDBSCAN [10] to cluster **F**, and label the cluster with the most elements as pristine (otherwise forged), implicitly assuming that forged pixels only occupy a relatively smaller portion. Features **F** extracted by our proposed contrastive learning and image-by-image overall loss may be already very discriminative, making an unsupervised algorithm sufficient to handle the clustering task. The performance comparison using different clustering algorithms is deferred to experimental results.

3.3. Feature Fusion Strategy

We now show that the performance of the standalone FOCAL can be further improved through simple yet effective feature fusion strategy. Fig. 5 gives an example of fusing two FOCAL α and FOCAL β with distinct backbones (*e.g.*, HRNet [46] or ViT [24]). The fused feature can be readily obtained from direct concatenation, namely,

$$\hat{\mathbf{F}} = \text{Concat}(\mathbf{F}^{\alpha}, \mathbf{F}^{\beta}), \qquad (4)$$

where $\mathbf{F}^{\alpha}$ and $\mathbf{F}^{\beta}$ are extracted features by FOCAL α and FOCAL β, respectively, and need to be scaled to the same resolutions. Prediction results can then be generated by the subsequent clustering and mapping accordingly. As will be validated through experiments, the above feature-level fusion significantly outperforms the naive result-level fusion [58]. Also, such a feature fusion strategy can be easily extended to cases with more than two FOCAL networks, and there is no retraining involved.

4. Experimental Results

In this section, we first present the detailed experimental settings. Then, image forgery detection/localization results on six public testing datasets are reported and compared with those of several state-of-the-art algorithms. Finally, extensive ablation studies and further analysis are conducted.

4.1. Settings

Training Datasets: We train the FOCAL using the *same* training dataset as [25, 15]. This training dataset contains over 800K forged images, where the forgery type is diverse, including splicing, copy-move, Photoshop and various post-processing operations.

Testing Datasets: Six commonly-used datasets are adopted for testing, namely, `Columbia` [18], `Coverage` [48], `CASIA` [9], `NIST` [14], `MISD` [21], and `FF++` [40]. These testing datasets encompass a plethora of highly sophisticated forgeries, *e.g.*, `MISD` comprising multi-source forgeries, and `FF++` harboring faces synthesized via GANs [13]. Note that **NO** overlap exists between the training and testing datasets, aiming to simulate the practical situation and evaluate the generalization of the forgery detection algorithms.

Competitors: The following state-of-the-art learning-based image forgery detection algorithms PSCC-Net [28], MVSS-Net [8], IF-OSN [50], CAT-Net [25], and TruFor [15] are selected as comparative methods. Their released codes can be found in their official links [1] [2] [3] [4] [5]. To ensure the fair comparison, we also *retrain* PSCC-Net, MVSS-Net, and IF-OSN on the training dataset of CAT-Net, in addition to directly using their released versions. We also involve two well-known clustering-based algorithms Lyu-NOI [31] and PCA-NOI [55] [6].

Evaluation Metrics: Follow the convention [8, 25, 50, 15], we utilize the pixel-level F1 and Intersection over Union (IoU) scores as the evaluation criteria (higher the better), where the threshold is set to 0.5 by default.

Implementation Details: We implement FOCAL by using PyTorch deep learning framework. HRNet [46] and ViT [24] are adopted for the specific backbones of the FOCAL extractor. The Adam [23] with default parameters is selected as the optimizer, and the learning rate is initialized to 1e-4. The batch size is set to 4 and the training is performed on 4 NVIDIA A100 GPU 40GB. All the input images are resized to 1024×1024, and the corresponding feature space of $\mathbf{F}$ is $\mathbb{R}^{256 \times 256 \times 256}$ for HRNet and $\mathbb{R}^{128 \times 128 \times 512}$ for ViT.

4.2. Quantitative Comparisons

Table 1 lists quantitative comparisons of different image forgery detection methods, in terms of pixel-level F1 and IoU scores. Here we additionally report the results of PSCC-Net [28], MVSS-Net [8], and IF-OSN [50] retrained with the training set of CAT-Net [25]. Generally, the retrained MVSS-Net and IF-OSN achieve comparable

---

[1]https://github.com/proteus1991/PSCC-Net
[2]https://github.com/dong03/MVSS-Net
[3]https://github.com/HighwayWu/ImageForensicsOSN
[4]https://github.com/mjkwon2021/CAT-Net
[5]https://github.com/grip-unina/TruFor
[6]NOI2&5 in https://github.com/MKLab-ITI/image-forensics

Methods	Columbia		Coverage		CASIA		MISD		NIST		FF++		Mean	
	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU
Lyu-NOI [31]	.522	.150	.481	.125	.356	.095	.507	.199	.478	.026	.496	.071	.473	.111
PCA-NOI [55]	.539	.168	.529	.125	.472	.093	.517	.150	.460	.046	.523	.108	.507	.115
PSCC-Net [28]	.577	.480	.655	.337	.716	.409	.746	.448	.300	.078	.509	.092	.584	.307
PSCC-Net[†] [28]	.850	.770	.584	.179	.753	.474	.735	.403	.632	.251	.518	.068	.679	.357
MVSS-Net [8]	.766	.591	.700	.384	.707	.396	.803	.525	.621	.243	.553	.127	.691	.378
MVSS-Net[†] [8]	<u>.888</u>	<u>.784</u>	.690	.356	.770	.509	.765	.450	.635	.255	.633	.241	.730	.433
IF-OSN [50]	.766	.612	.561	.178	.741	.465	<u>.811</u>	<u>.548</u>	.639	.246	.628	.266	.691	.386
IF-OSN[†] [50]	.846	.719	.651	.314	.828	.553	.765	.521	.608	.226	.607	.222	.717	.426
CAT-Net [25]	.864	.741	.614	.231	<u>.846</u>	<u>.642</u>	.665	.314	.620	.230	.534	.095	.690	.375
TruFor [15]	.821	.734	<u>.741</u>	<u>.450</u>	.835	.626	.746	.423	<u>.688</u>	<u>.343</u>	<u>.817</u>	<u>.565</u>	<u>.774</u>	<u>.523</u>
FOCAL (HRNet)	.962	.929	.769	.524	.864	.706	.857	.639	.710	.403	.837	.605	.833	.634
FOCAL (ViT)	.980	.969	.835	.647	.897	.766	.874	.666	.724	.433	.846	.630	.859	.685
FOCAL (Fusion)	**.981**	**.970**	**.863**	**.693**	**.898**	**.777**	**.886**	**.690**	**.737**	**.446**	**.904**	**.740**	**.878**	**.719**

Table 1: Quantitative comparison of detection results using F1 and IoU as criteria. [†]: retrained versions with CAT-Net datasets. The best results are in **bold** and the second best results (excluding FOCAL variants) are in <u>underlined</u>.
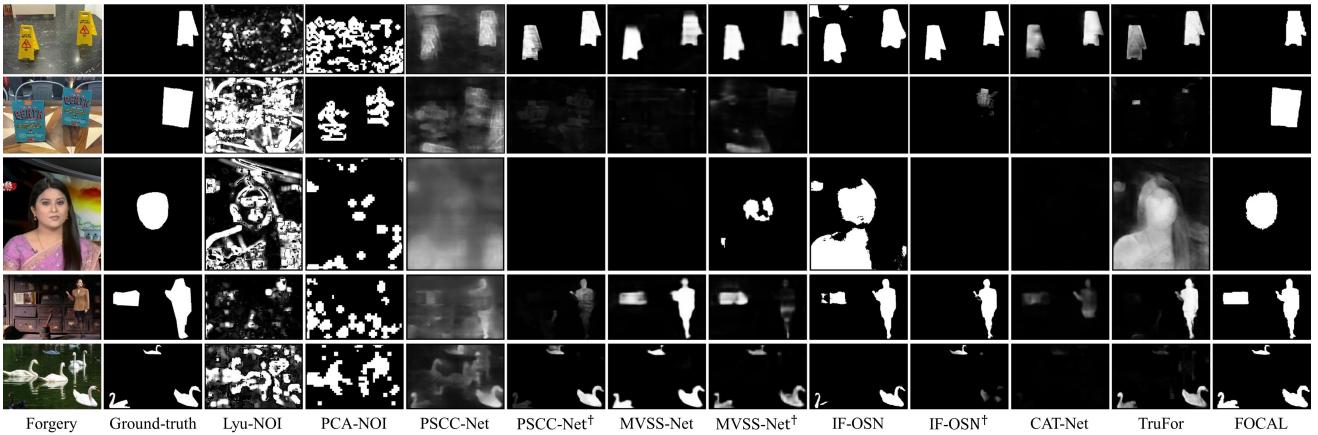

Figure 6: Qualitative comparison of forgery detection on some representative testing images. For each row, the images from left to right are forgery, ground-truth forgery mask, detection results generated by Lyu-NOI, PCA-NOI, PSCC-Net, MVSS-Net, IF-OSN, CAT-Net, TruFor and our FOCAL (Fusion), respectively. [†]: retrained versions with CAT-Net datasets.

performance to their officially released versions, while the retrained PSCC-Net leads to much better performance, *i.e.*, +9.5% in F1 and +5.0% in IoU. This phenomenon indicates that different training datasets may have a huge impact on the eventual performance. For the benefits of the competing methods, we take the higher performance of the original and retrained versions in the following analysis.

As can be observed from Table 1, the traditional clustering-based algorithms Lyu-NOI [31] and PCA-NOI [55] achieve unsatisfactory performance of ∼50% in F1 and ∼11% in IoU. This is mainly due to the fact that their hand-crafted noise features are heavily corrupted by post-processing operations commonly observed in the testing datasets. In contrast, the latest classification-based

competitors with deep learning offer much better detection results. Among them, MVSS-Net and IF-OSN achieve slightly better results on Columbia and MISD datasets, with IoU scores being 78.4% and 54.8%, respectively; while on CASIA dataset, CAT-Net exhibits the better performance of IoU scores 64.2%. The recently published TruFor [15] achieves good results on the remaining three datasets. Thanks to the contrastive learning with image-by-image overall loss and the unsupervised clustering, our FOCAL, whether utilizing single extractor (HRNet or ViT) or fused (HRNet + ViT), consistently leads to the best performance over all testing datasets in both F1 and IoU criteria. Particularly, FOCAL (Fusion) is shown to be effective in further boosting the performance (*e.g.*, +11.0%

6

and +4.6% IoU on `FF++` and `Coverage` respectively), via simple feature-level concatenation without the need of re-training. As can also be observed, FOCAL (Fusion) can avoid the bias of a single extractor backbone on some testing examples. Overall, FOCAL (Fusion) surpasses the best competing algorithm by big margins, *e.g.*, +24.3%, +18.6%, +17.5%, +14.2%, +13.5%, and +10.3% in IoU on datasets `Coverage`, `Columbia`, `FF++`, `MISD`, `CASIA`, and `NIST`, respectively.

4.3. Qualitative Comparisons

Fig. 6 presents forgery detection results on some representative testing images. More qualitative comparisons can be found in the supplementary file. As can be noticed, traditional clustering-based methods with hand-crafted noise features Lyu-NOI [31] and PCA-NOI [55] perform poorly; many forged regions cannot be detected and a large number of false alarms exist. The classification-based method PSCC-Net [28] also does not perform satisfactorily on these cross-domain testing data, where most of the forged regions are not detected. Similarly, CAT-Net [25] and MVSS-Net [8] miss many forged regions, resulting in inaccurate detection. TruFor [15] and IF-OSN [50] are slightly better in some examples; but many forged regions cannot be accurately identified and many untouched regions are falsely detected as forged. In contrast, our FOCAL (Fusion) not only accurately detects forged regions but also performs rather stably on cross-domain testing. Also, the false alarms have been remarkably suppressed.

Recall FOCAL implicitly assumes that all forged regions within one single image share similar features, though they could be made with different types of forgery (*e.g.*, splicing and inpainting). An interesting question arising is whether FOCAL can detect multiple types of forgery simultaneously. The answer is affirmative. The examples shown in the last two rows of Fig. 6 are from the `MISD` dataset, where multi-source splicing forgery is used. It can be noticed that FOCAL can still produce satisfactory detection results. The reason for the success in this challenging and practical scenario may be that there are more pristine than forged regions. The cluster with the largest amount of data will be directly marked as pristine, while the clusters with the smaller amount of data will be merged and all marked as forged.

4.4. Ablation Studies

We now analyze how each component contributes to the FOCAL framework in terms of extractor backbone, loss function, and clustering algorithm.

Framework	Extractor	Testing Datasets (F1 criterion)				Mean
		CASIA	MISD	NIST	FF++	
Traditional	HRNet	.718	.727	.569	.675	.672
FOCAL (Single)	HRNet	.860	.853	.702	.837	.813
	ViT	.897	.874	.724	.846	.838
	MiT	.680	.722	.659	.775	.709
	ConvNeXt	.468	.531	.453	.584	.509
	EffNet	.527	.598	.536	.648	.577
FOCAL (Fusion)	HRNet+ViT	**.898**	**.886**	**.737**	**.904**	**.856**

Table 2: Ablation studies regarding the extractor backbone.
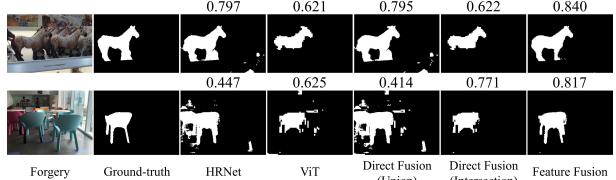

Figure 7: Impact of different extractors and fusions. The number above the mask represents the score of IoU.

4.4.1 Extractor Backbones

We start ablation studies with the selection of the FOCAL extractor, where the key point is how to select the backbone. Since our focus is *not* to design a brand new backbone, we directly adopt the most commonly-used backbones proposed in recent years, namely, HRNet [46], ConvNeXt [29], EffNet [44], ViT [24], and MiT [52], for the comparison. The corresponding detection results are shown in Table 2, where the first row gives the performance of traditional classification-based framework as a comparison. As can be observed, ViT leads to the superior detection performance among these compared backbones. This might be due to its attention ability to extract richer forgery features by globally modelling long-range dependencies. Also, note that the extractor backbone in FOCAL can be flexibly replaced by a more advanced architecture when it is available. Further, in Fig. 7, we compare the visual results of the predicted forgery masks when using different backbones and fusion strategies. It can be seen that the feature fusion surpasses naive result-level fusions (*i.e.*, union or intersection fusions) by a significant margin.

4.4.2 Loss Functions

The contrastive learning module plays a crucial role in FOCAL. We now evaluate the performance of FOCAL variants by replacing our adopted $\mathcal{L}_{InfoNCE++}$ (see (2)) with existing contrastive losses, such as Triplet [1], DCL [53], Circle [43], and the original InfoNCE [35]. Note that, for all variants, except the one marked as (Batch), the image-by-image manner of computing the overall contrastive loss $\mathcal{L}_{ct}$ in (3)

Loss	Testing Datasets (F1 criterion)				Mean
	CASIA	MISD	NIST	FF++	
Triplet	.023	.015	.031	.064	.033
DCL	.433	.471	.455	.586	.486
Circle	.862	.850	.692	.842	.811
InfoNCE	.858	.863	.704	.856	.820
$\mathcal{L}_{InfoNCE++}$ (Batch)	.674	.759	.561	.730	.681
$\mathcal{L}_{InfoNCE++}$	**.898**	**.886**	**.737**	**.904**	**.856**

Table 3: Ablation studies regarding the loss function.

Clustering	Testing Datasets (F1 criterion)				Mean
	CASIA	MISD	NIST	FF++	
BIRCH	.872	.858	.711	.865	.827
Hierarchical	.878	.863	.717	.875	.833
K-means	.892	.875	.729	.892	.847
B-K-means	.864	.873	.724	.895	.839
HDBSCAN	**.898**	**.886**	**.737**	**.904**	**.856**

Table 4: Ablation studies regarding the clustering.

Figure 8: Impact of different clustering algorithms.

keeps unchanged.

As can be seen from Table 3, not all of these loss functions are suitable for the forgery detection task. For example, Triplet restricts an equal penalty strength to the distance score of every query positive or negative pair [43], which results in the model collapse. By re-weighting each distance score under supervision, Circle has a more flexible optimization and definite convergence target, far exceeding Triplet. Although DCL and InfoNCE have the same supervision mechanism, the positive constraint removed by DCL makes it easy to get stuck in poor local optima, causing the performance of DCL far inferior to that of InfoNCE. Additionally, by supervising the distance between the query and multiple positive keys simultaneously, our improved $\mathcal{L}_{InfoNCE++}$ can further improve over the vanilla InfoNCE by an average of +3.6% F1 score. Finally, we give the results when the overall contrastive loss is computed by merging all features in a batch and calculating at a batch level. As expected, our image-by-image overall loss design significantly outperforms such a batch-level loss (+17.5% in F1). The big performance gap further indicates the necessity of explicitly using the relative definition of forged and pristine pixels within one single image.

4.4.3 Clustering Algorithms

Apart from contrastive learning, another key module of FOCAL is the clustering algorithm for generating the final predicted forgery mask. To explore the most suitable clustering algorithm for the FOCAL framework, we evaluate the most popular clustering algorithms, K-means [30], B-K-means [3], BIRCH [57], Hierarchical [20] and HDBSCAN [10], and report the results in Table 4.

For K-means, B-K-means, and BIRCH algorithms, the number of clusters to be formed is set to 2, while other parameters take their default values. As can be observed, the aforementioned algorithms exhibit comparable performance, attributed to the discriminative nature of the features **F** learned by the extractor. Among them, HDBSCAN performs the best, surpassing the second-place one by 0.9% F1. Recalling that **F** has $256 \times 256 = 65536$ elements to be clustered. Those clustering algorithms such as spectral clustering [41] and affinity propagation [12] that cannot

be extended to large-scale elements are extremely slow and thereby omitted. We also would like to point out a potential limitation of using fixed numbers of clusters in K-means, B-K-means, and BIRCH. For completely pristine images (no forged regions), these clustering methods still force to produce two clusters, inevitably resulting in false alarms (see last row in Fig. 8). In constract, our adopted density-based algorithm HDBSCAN can dynamically determine the number of final clusters, effectively suppressing the false alarms for pristine images.

5. Conclusion

We have explicitly pointed out the importance of the relative definition of forged and pristine pixels within an image, which has been severely overlooked by existing forgery detection methods. Inspired by this rethinking, we have proposed FOCAL, a novel, simple yet effective image forgery detection framework, based on pixel-level contrastive learning with the image-by-image overall loss function and unsupervised clustering. Extensive experimental results have been given to demonstrate our superior performance.

6. Appendix

6.1. Details of Training/Testing Datasets

Similar to the previous works [25, 15], we collect the training data based on the following datasets: SP-COCO [25], CM-COCO [25], CM-RAISE [25], CM-C-RAISE [25], CASIA-v2 [38], and IMD2020 [34]. Specifically, CASIA-v2 is a widely-adopted dataset that contains various *multi-source* splicing and copy-move forgeries, while IMD2020 collects real-world manipulated images from the Internet. Considering the insufficient numbers of images in these two datasets, Kwon *et al*. [25] utilized splicing and copy-move methods to produce a large amount of forged

| Datasets | #Data | Forgery Types | | | | | Resolution |
		SP	CM	SW	PP	GAN	(Average)
Training Datasets							
-SP-COCO [25]	200K	✓			✓		640×480
-CM-COCO [25]	200K		✓		✓		640×640
-CM-RAISE [25]	200K		✓		✓		512×512
-CM-C-RAISE [25]	200K		✓		✓		512×512
-CASIA-v2 [38]	5105	✓	✓	✓	✓		384×256
-IMD2020 [34]	2010	✓	✓	✓	✓		1920×1200
Testing Datasets							
-Coverage [48]	100		✓	✓	✓		520×430
-Columbia [18]	160	✓					1152×768
-NIST [14]	540	✓	✓	✓	✓		5616×3744
-CASIA [9]	920	✓	✓	✓	✓		384×256
-MISD [21]	227	✓			✓		384×256
-FF++ [40]	1000				✓	✓	480×480

Table 5: Lists of training and testing datasets. SP, CM, SW, PP are short for splicing, copy-move, software (*e.g.* Photoshop), and post-processing, respectively.

images based on pristine datasets COCO [27] and RAISE [7]. To better mimic the distribution of real-world images, a variety of post-processing operations such as resizing, rotation, and compression are involved.

For better measuring the generalization of forensic algorithms, we adopt several cross-domain datasets for testing, namely, Coverage [48], Columbia [18], NIST [14], CASIA [9], MISD [21], and FF++ [40]. These datasets accommodate numerous skillfully-forged images with a wide distribution of data sources, resolutions, and formats. In particular, FF++ contains faces synthesized by various Generative Adversarial Network (GAN) algorithms, which are unknown during the training phase. Overview statistics for these datasets are summarized in Table 5.

7. Detailed Calculation of Metrics

To evaluate the detection performance, we utilize two commonly-used metrics, F1 and IoU. Formally, the macro-averaged F1 is defined as

$$F1 = \frac{1}{Y} \sum_{y=1}^{Y} \frac{2 \times \text{TP}_y}{2 \times \text{TP}_y + \text{FP}_y + \text{FN}_y}, \quad (5)$$

where TP_y, FP_y, and FN_y represent True Positive, False Positive, and False Negative for a given class y ("pristine" or "forged"), respectively.

The IoU can be calculated as follows:

$$\text{IoU} = \frac{\mathbf{P} \cap \mathbf{Y}}{\mathbf{P} \cup \mathbf{Y}}, \quad (6)$$

where $\mathbf{P}$ and $\mathbf{Y}$ are the prediction and ground-truth masks, respectively. Please refer to our code for the specific implementation, which is mainly based on the scikit-learn [37] extension package.
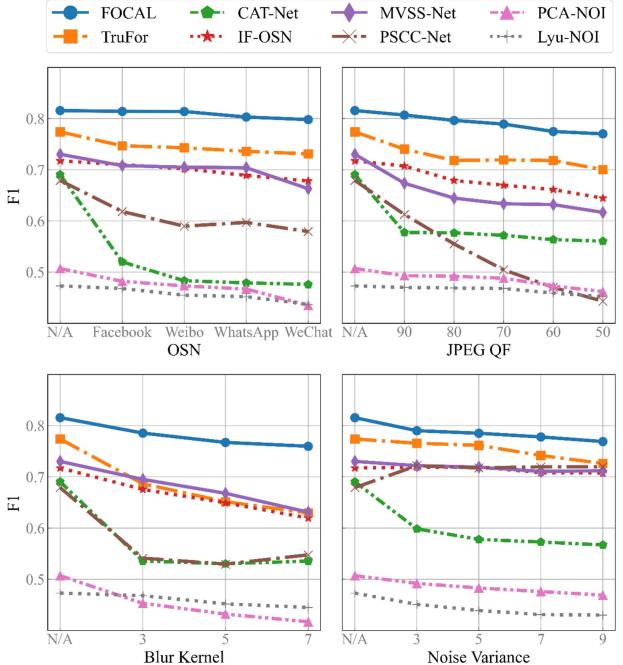
Figure 9: Robustness evaluations against OSN transmission, JPEG compression, Gaussian blurring and Gaussian noise addition.

7.1. Robustness Evaluation

The forged images often undergo a series of post-processing operations, such as compression, blurring, noise addition, attempting to eliminate forgery traces or mislead forgery detection algorithms. In addition, online social networks (OSNs), as prevailing transmission channels for images, have been shown to seriously affect image forensic algorithms [50]. It is therefore important to evaluate the robustness of all competing algorithms against post-processing operations and OSN transmission. Specifically, we apply the aforementioned degradation to the original testing datasets, and plot the results in Fig. 9. It can be observed that although CAT-Net [25] achieves good performance on the original dataset, it is vulnerable to post-processing and OSN transmissions. TruFor [15], MVSS-Net [8] and IF-OSN [50] exhibit certain degrees of robustness against these distortions. In contrast, our FOCAL still consistently achieves the best performance and robustness over these competitors. For instance, FOCAL only suffers ~0.2% performance degradation against Facebook or Weibo transmissions.

7.2. Additional Qualitative Comparisons

More comparisons over testing datasets Coverage [48], Columbia [18], NIST [14], CASIA [9], MISD [21],

FF++ [40] are given in Figs. 10-15, respectively. For each row in these comparisons, the images from left to right are forgery (input), ground-truth, detection result (output) generated by Lyu-NOI [31], PCA-NOI [55], PSCC-Net [28], MVSS-Net [8], IF-OSN [50], CAT-Net [25], TruFor [15] and our FOCAL, respectively. † means the method retrained by using CAT-Net [25] datasets.

References

[1] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proc. British Mach. Vis. Conf.*, pages 1–11, 2016. 7

[2] X. Bi, W. Yan, B. Liu, B. Xiao, W. Li, and X. Gao. Self-supervised image local forgery detection by jpeg compression trace. In *Proc. AAAI Conf. Arti. Intell.*, volume 37, pages 232–240, 2023. 2

[3] A. Bilge and H. Polat. A scalable privacy-preserving recommendation scheme via bisecting k-means clustering. *Inf. Process. Manag.*, 49(4):912–927, 2013. 8

[4] L. Bondi, S. Lameri, D. Guera, P. Bestagini, E. Delp, S. Tubaro, et al. Tampering detection and localization through clustering of camera-based cnn features. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. Workshops*, pages 43–52, 2017. 3

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *Proc. Int. Conf. Mach. Learn.*, pages 1597–1607, 2020. 4

[6] D. Cozzolino and L. Verdoliva. Noiseprint: a cnn-based camera model fingerprint. *IEEE Trans. Inf. Forensics and Security*, 15(1):114–159, 2020. 2, 3, 4

[7] D. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato. Raise: A raw images dataset for digital image forensics. In *Proc. ACM Multimed. syst. Conf.*, pages 219–224, 2015. 9

[8] C. Dong, X. Chen, R. Hu, J. Cao, and X. Li. Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 45(3):3539–3553, 2023. 2, 5, 6, 7, 9, 10

[9] J. Dong, W. Wang, and T. Tan. Casia image tampering detection evaluation database. In *IEEE China Summit Inter. Conf. Signal Info. Proc.*, pages 422–426. IEEE, 2013. 5, 9, 12

[10] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. Knowl. Discov. Data Min.*, pages 226–231, 1996. 5, 8

[11] Y. Fan, P. Carre, and C. Fernandez-Maloigne. Image splicing detection with local illumination estimation. In *Proc. IEEE Int. Conf. Image Proc.*, pages 2940–2944. IEEE, 2015. 2

[12] B. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007. 8

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. Neural Info. Process. Syst.*, pages 2672–2680, 2014. 5

[14] H. Guan, M. Kozak, E. Robertson, Y. Lee, A. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus. Mfc

datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, pages 63–72, 2019. 5, 9, 13

[15] F. Guillaro, D. Cozzolino, A. Sud, N. Dufour, and L. Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 20606–20615, 2023. 2, 3, 4, 5, 6, 7, 8, 9, 10

[16] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 9729–9738, 2020. 4

[17] Z. He, W. Lu, W. Sun, and J. Huang. Digital image splicing detection based on markov features in dct and dwt domain. *Pattern Recognition*, 45(12):4292–4299, 2012. 2

[18] Y. Hsu and S. Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *IEEE Inter. Conf. Multim. Expo*, pages 549–552. IEEE, 2006. 5, 9, 12

[19] M. Huh, A. Liu, A. Owens, and A. A. Efros. Fighting fake news: image splice detection via learned self-consistency. In *Proc. Eur. Conf. Comput. Vis.*, pages 101–117, 2018. 2, 4

[20] JH Ward Jr. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.*, 58(301):236–244, 1963. 8

[21] K. Kadam, S. Ahirrao, and K. Kotecha. Multiple image splicing dataset (misd): a dataset for multiple splicing. *Data*, 6(10):102–113, 2021. 5, 9, 13

[22] X. Kang, M. C. Stamm, A. Peng, and K. R. Liu. Robust median filtering forensics using an autoregressive model. *IEEE Trans. Inf. Forensics and Security*, 8(9):1456–1468, 2013. 2

[23] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. *arXiv preprint:1412.6980*, 2014. 5

[24] A. Kolesnikov, A. Dosovitskiy, D. Weissenborn, G. Heigold, J. Uszkoreit, L. Beyer, M. Minderer, M. Dehghani, N. Houlsby, S. Gelly, T. Unterthiner, and X. Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. Int. Conf. Learn. Representat.*, pages 1–22, 2021. 5, 7

[25] M. Kwon, S. Nam, I. Yu, H. Lee, and C. Kim. Learning jpeg compression artifacts for image manipulation detection and localization. *Int. J. Comput. Vis.*, 130(8):1875–1895, 2022. 2, 5, 6, 7, 8, 9, 10

[26] Y. Li and J. Zhou. Fast and effective image copy-move forgery detection via hierarchical feature point matching. *IEEE Trans. Inf. Forensics and Security*, 14(5):1307–1322, 2019. 2

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: common objects in context. In *Proc. Eur. Conf. Comput. Vis.*, pages 740–755, 2014. 9

[28] X. Liu, Y. Liu, J. Chen, and X. Liu. Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. *IEEE Trans. Circuits Syst. Video Technol.*, 32(11):7505–7517, 2022. 2, 5, 6, 7, 10

[29] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 11976–11986, 2022. 7

[30] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inform. Theor.*, 28(2):129–137, 1982. 8

[31] S. Lyu, X. Pan, and X. Zhang. Exposing region splicing forgeries with blind local noise estimation. *Int. J. Comput. Vis.*, 110(1):202–221, 2014. 2, 3, 5, 6, 7, 10

[32] F. Niloy, K. Bhaumik, and S. Woo. Cfl-net: Image forgery localization using contrastive learning. In *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, pages 4642–4651, 2023. 2, 3, 4

[33] Y. Niu, B. Tondi, Y. Zhao, R. Ni, and M. Barni. Image splicing detection, localization and attribution via jpeg primary quantization matrix estimation and clustering. *IEEE Trans. Inf. Forensics and Security*, 16:5397–5412, 2021. 3

[34] A. Novozamsky, B. Mahdian, and S. Saic. Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. In *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, pages 71–80, 2020. 8, 9

[35] A. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint:1807.03748*, 2018. 4, 7

[36] X. Pan, X. Zhang, and S. Lyu. Exposing image forgery with blind noise estimation. In *Proc. ACM Multimed. Workshop Multimed. Security*, pages 15–20, 2011. 3

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011. 9

[38] N. Pham, J. Lee, G. Kwon, and C. Park. Hybrid image-retrieval method for image-splicing validation. *Symmetry*, 11(1):83–98, 2019. 8, 9

[39] S. Pyatykh, J. Hesser, and L. Zheng. Image noise level estimation by principal component analysis. *IEEE Trans. Image Process.*, 22(2):687–699, 2012. 3

[40] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: learning to detect manipulated facial images. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1–11, 2019. 5, 9, 10, 13

[41] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 22(8):888–905, 2000. 8

[42] K. Sun, T. Yao, S. Chen, S. Ding, J. Li, and R. Ji. Dual contrastive learning for general face forgery detection. In *Proc. AAAI Conf. Arti. Intell.*, volume 36, pages 2316–2324, 2022. 2

[43] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 6398–6407, 2020. 7, 8

[44] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proc. Int. Conf. Mach. Learn.*, pages 6105–6114, 2019. 7

[45] D. T. Trung, A. Beghdadi, , and M. Larabi. Blind inpainting forgery detection. In *Proc. IEEE Global Conf. Signal Inf. Process*, pages 1019–1023, 2014. 2

[46] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, and X. Wang. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 43(10):3349–3364, 2020. 5, 7

[47] M. Wang, X. Fu, J. Liu, and Z. Zha. Jpeg compression-aware image forgery localization. In *Proc. ACM Int. Conf. Multimed.*, pages 5871–5879, 2022. 3

[48] B. Wen, Y. Zhu, R. Subramanian, T. Ng, X. Shen, and S. Winkler. Coverage: A novel database for copy-move forgery detection. In *Proc. IEEE Int. Conf. Image Proc.*, pages 161–165, 2016. 5, 9, 12

[49] H. Wu and J. Zhou. Iid-net: image inpainting detection network via neural architecture search and attention. *IEEE Trans. Circuits Syst. Video Technol.*, 32(3):1172–1185, 2021. 2

[50] 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 and Security*, 17(1):443–456, 2022. 2, 5, 6, 7, 9, 10

[51] Z. Wu, Y. Xiong, S. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 3733–3742, 2018. 4

[52] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. Alvarez, and P. Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Proc. Neural Info. Process. Syst.*, pages 12077–12090, 2021. 7

[53] C. Yeh, C. Hong, Y. Hsu, T. Liu, Y. Chen, and Y. LeCun. Decoupled contrastive learning. In *Proc. Eur. Conf. Comput. Vis.*, pages 668–684, 2022. 7

[54] Q. Yin, J. Wang, W. Lu, and X. Luo. Contrastive learning based multi-task network for image manipulation detection. *Signal Processing*, 201:108709, 2022. 3, 4

[55] H. Zeng, Y. Zhan, X. Kang, and X. Lin. Image splicing localization using pca-based noise level estimation. *Multimed. Tools. Appl.*, 76:4783–4799, 2017. 3, 5, 6, 7, 10

[56] Y. Zeng, B. Zhao, S. Qiu, T. Dai, and S. Xia. Towards effective image manipulation detection with proposal contrastive learning. *IEEE Trans. Circuits Syst. Video Technol.*, pages 1–12, 2023. 3, 4

[57] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: an efficient data clustering method for very large databases. *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 25(2):103–114, 1996. 8

[58] L. Zheng, Y. Zhang, and V. Thing. A survey on image tampering and its detection in real-world photos. *J. Vis. Commun. Image Represent.*, 58:380–399, 2019. 5

[59] N. Zhu and Z. Li. Blind image splicing detection via noise level function. *Signal Process. Image Commun.*, 68:181–192, 2018. 3

[60] D. Zoran and Y. Weiss. Scale invariance and noise in natural images. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 2209–2216, 2009. 3
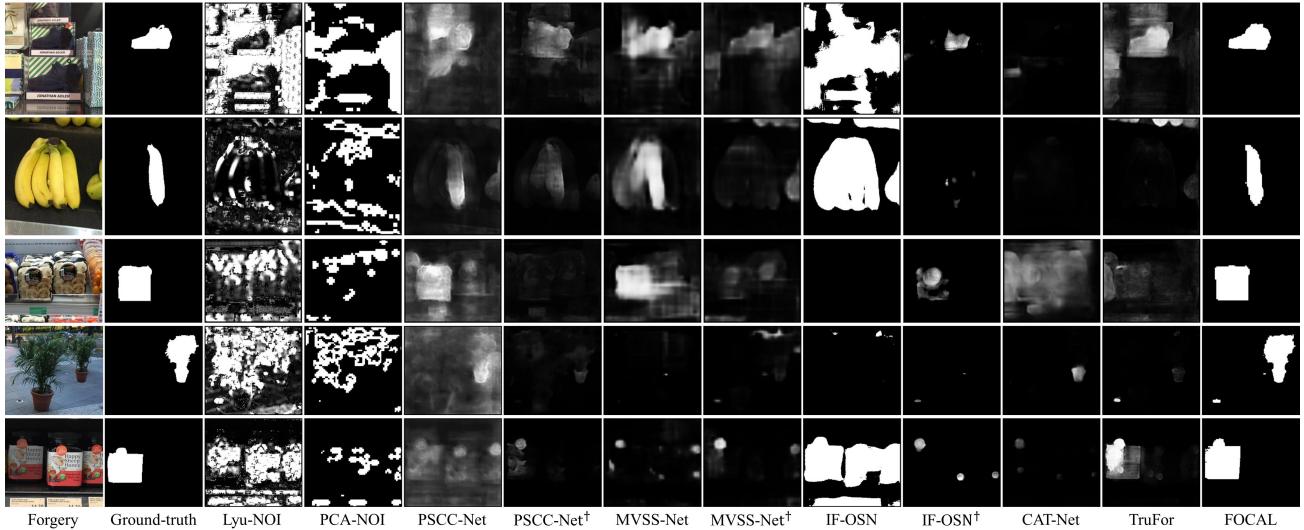
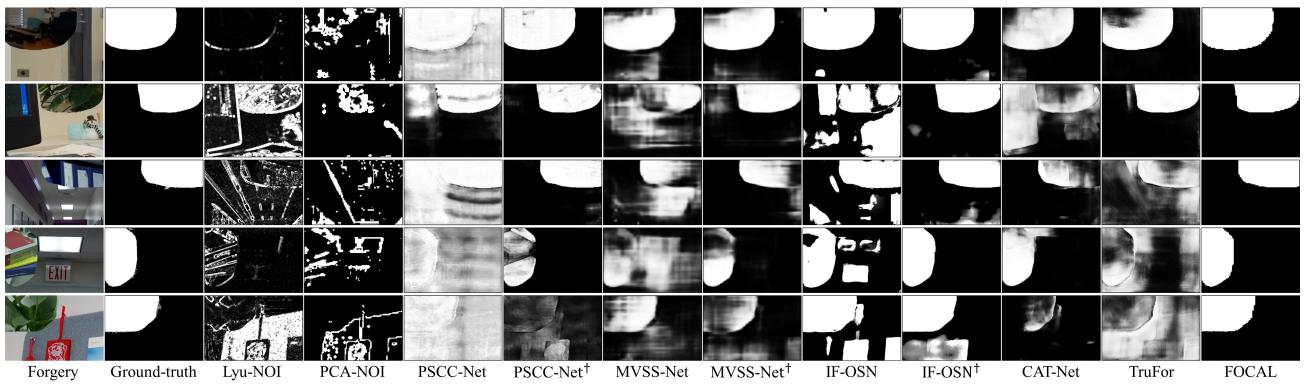Figure 10: Qualitative comparisons on Coverage [48] dataset.

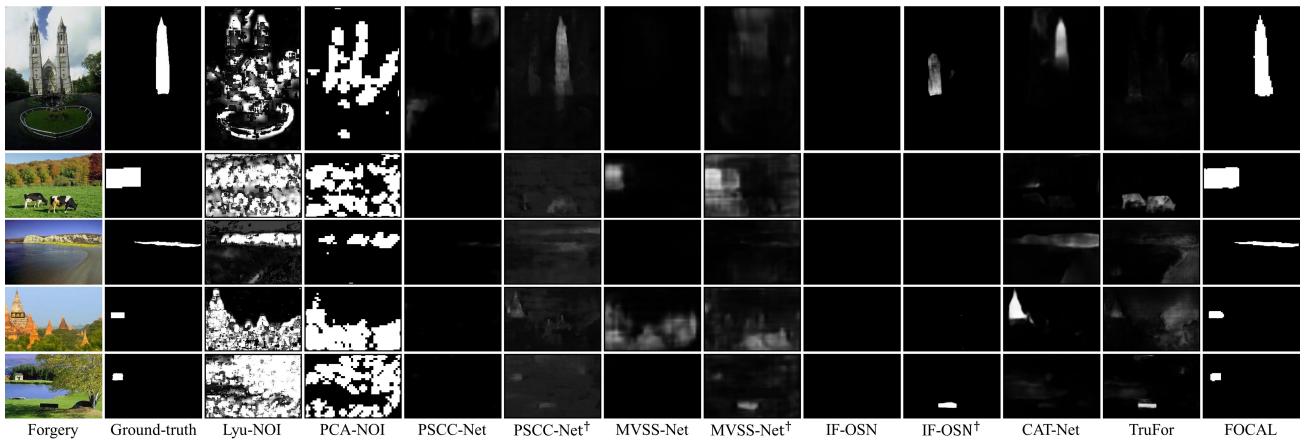Figure 11: Qualitative comparisons on Columbia [18] dataset.

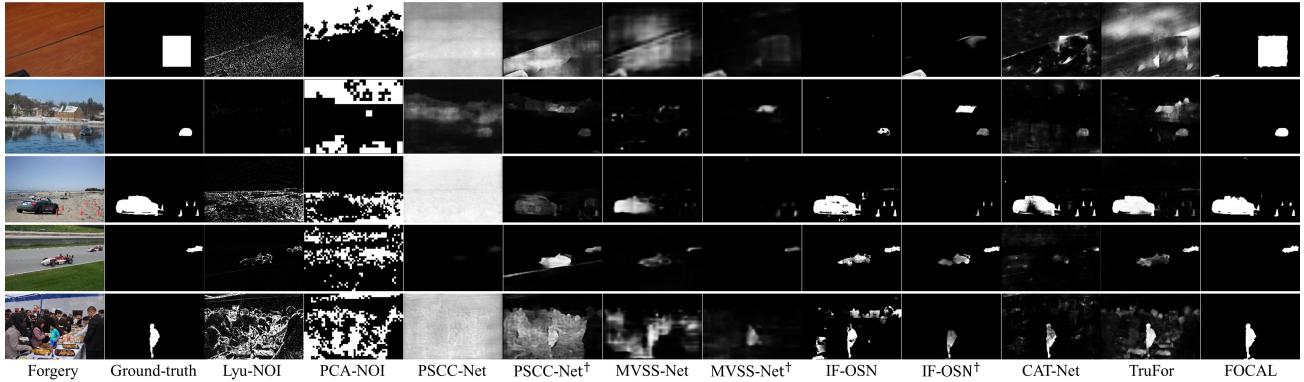Figure 12: Qualitative comparisons on CASIA [9] dataset.

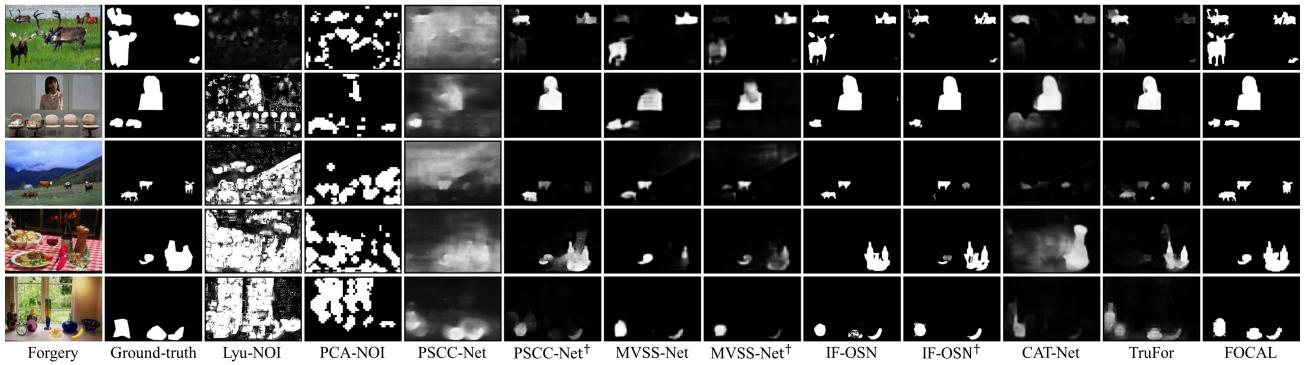Figure 13: Qualitative comparisons on NIST [14] dataset.
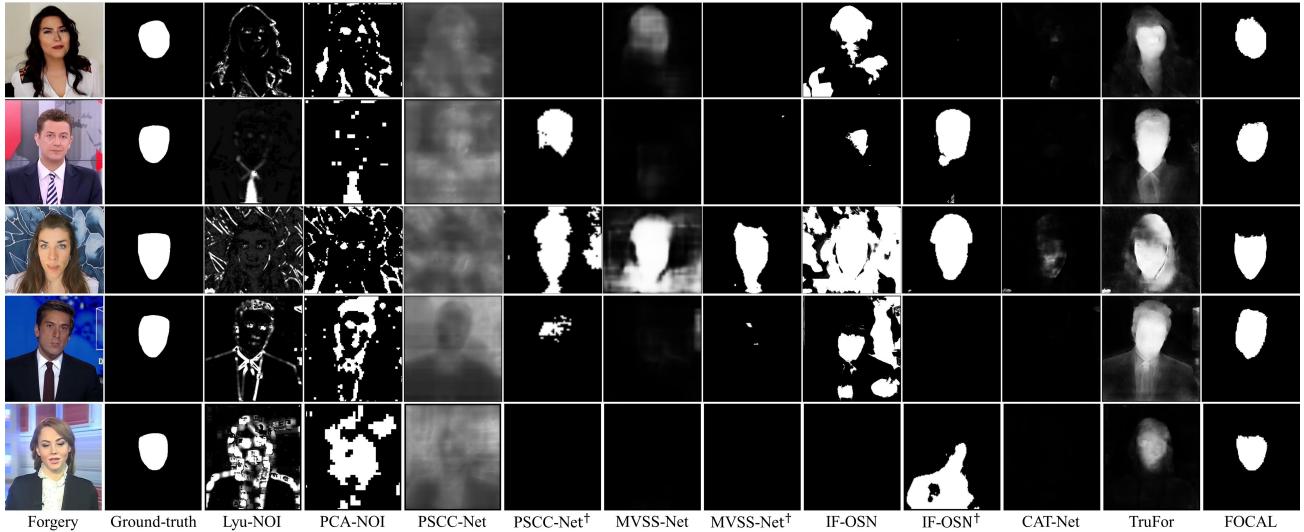

Figure 14: Qualitative comparisons on MISD [21] dataset.

Figure 15: Qualitative comparisons on FF++ [40] dataset.