SISL:Self-Supervised Image Signature Learning for Splicing Detection & Localization

Susmit Agrawal[1], Prabhat Kumar[3]*, Siddharth Seth[2]*, Toufiq Parag[2]†, Maneesh Singh[2], Venkatesh Babu[1]
[1]Indian Institute of Science, India [2]Verisk AI Research, US [3]Ola Electric, India

Abstract

Recent algorithms for image manipulation detection almost exclusively use deep network models. These approaches require either dense pixelwise groundtruth masks, camera ids, or image metadata to train the networks. On one hand, constructing a training set to represent the countless tampering possibilities is impractical. On the other hand, social media platforms or commercial applications are often constrained to remove camera ids as well as metadata from images. A self-supervised algorithm for training manipulation detection models without dense groundtruth or camera/image metadata would be extremely useful for many forensics applications. In this paper, we propose a self-supervised approach for training splicing detection/localization models from frequency transform of images. To identify the spliced regions, our deep network learns a representation to capture an image-specific signature by enforcing (image) self consistency. We experimentally demonstrate that our proposed model can yield similar or better performances as compared to multiple existing methods on standard datasets without relying on labels or metadata.

1. Introduction

The history of image manipulation dates back almost as early as the invention of photography itself [68]. Rapid advances in photographic devices and editing software in recent years have empowered the general population to easily alter an image. Photo tampering has crucial implications on legal arbitration [59,64], journalism [27,52] (thereby public opinion and politics), fashion [11], advertising [42], insurance [9] industries among others. The impact of content fabrication on social media platforms, which allow manipulated content to be uploaded and disseminated extremely fast, is even more critical [31,69].

Researchers have been investigating digital forensics for almost two decades [23, 25, 62, 63]. One particular variant of image tampering, image splicing, garnered significant attention in the digital forensic community. In this mode of image manipulation, parts of different images are spliced together, and subsequently edited manually (with e.g., GIMP, Adobe Photoshop) or computationally [61]. In this paper, we also address the problem of image splicing detection and localization.

Many recent methods employ neural networks to detect image splicing and predict a pixelwise mask of the spliced region in an end-to-end fashion [3–5, 38, 43, 65, 72, 73, 80]. For training the detection/localization network, these algorithms require pixelwise (dense) groundtruth masks of spliced regions that are remarkably tedious and expensive to annotate. More importantly, the feasibility of generating a large enough representative dataset for fully supervised manipulation learning is questionable since the space of forgery operations is vast and extremely diverse (if not infinite) [39,43]. It is therefore difficult to guarantee the robustness of end-to-end approaches on real world data despite their excellent performances on the public datasets [68].

A surrogate approach to circumvent the need for dense pixelwise groundtruth is to identify the micro-level signature imprinted by device hardware [50, 51], image processing software [54] or by the GAN based artificial generators [53]. In a spliced (or edited) image, it is rational to expect the manipulated and pristine regions to possess different fingerprints. Several studies [12, 17, 18, 20, 54, 55] proposed elegant methods to train a CNN to distinguish between the different traces of authentic and forged areas. These methods rely on camera/device IDs to train the CNN.

Huh et al. [39] pushed the envelope further in this direction by learning the consistency between authentic and forged regions under the supervision of image metadata. In [39], a CNN is trained to match the latent space representations for a pair of image blocks with the same EXIF data and contrast those for patches with different metadata. However, social media platforms, image hosting services and commercial applications are forced to strip the metadata (EXIF) and camera id for various reasons [76]. An algorithm to learn the representation for forensics pur-

*Equal contribution
†Corresponding author

poses without camera ID or metadata – perhaps in a self-supervised fashion – would be extremely appealing for applications where these information are not available.

Self-supervised learning algorithms [13, 16, 30, 35, 77] precipitated a breakthrough in representation learning with minimal or no annotated examples. Self-supervision has not yet gained widespread attention in forensics with the notable exception of [39]. Huh et al. [39] also discuss training a siamese network to determine whether a pair of image blocks were extracted from the same or different image without using EXIF metadata. The reason for the inferior performance of the ensuing model was surmised to be the lack of a large training dataset. We believe the compelling reason instead to be the propensity of CNN to learn image characteristics (e.g., color histograms [39]) or semantic content as opposed to device signature even with a large dataset.

Frequency transform is an alternative source of information for tracing image manipulation. Frequency transform (FT) largely discards the spatial and semantic details but retains significant information to detect source or manipulation signature. Classical works on image manipulation detection thoroughly investigated cues of image source as well as any subsequent manipulation in frequency domain [6, 7, 22, 47–49, 57, 71]. Frank et al. [24] have lately demonstrated impressive success in identifying source signature from FT of artificially manipulated images produced by generative models, e.g., GAN [8, 40, 56]. GAN generated images have been shown to be relatively easier to detect [70]. The study of [24] did not report its performance on manually tampered images and requires camera id for training (not self-supervised).

In this paper, we propose a self-supervised training method to learn feature (latent) representation for image forensics. Our approach learns the latent representations from the frequency transformation of image patches (blocks). Given the FTs of two patches, we utilize a CNN and contrastive loss – inspired by those proposed in SimCLR [13] – to learn whether they originate from the same or different *images*. In effect, our method aims to learn an image specific signature from the frequency domain to identify traces of tampering. For inference, we apply a mean-shift based clustering algorithm to group the authentic & fake patches based on the cosine similarity of the learned latent features.

Our experimental results suggest that the use of representation learning to capture image trace in the frequency domain is very effective for manipulation detection/localization. The representations learned in a self-supervised fashion from FT of image blocks are shown to achieve similar or better accuracy than EXIF-SC [39], MantraNet [73] in a realistic environment. We also demonstrate that features learned from RGB values by the same architecture and training cannot achieve the same performance.

In contrast to all aforementioned studies, our approach learns only from the FT content of an image and does not require pixelwise masks, camera IDs, or EXIF metadata. The simplicity of our model and the use of standard architecture/hyperparameters make our results easily reproducible. All these characteristics are highly desirable for large scale training of robust models to build practical solutions.

2. Related work

Dense Splicing Prediction with CNN: One of the early works on dense prediction for manipulation detection couples an LSTM with CNN to discover the tampering location [3]. A number of studies have followed this particular direction since then. MantraNet [73] exploits a localization network operating on the features from initial convolutional layers to identify manipulation. Wu et al. [73] also proposed an interesting approach for artificially generating the spliced images for training its model. Multiple studies built upon this idea and adopted an adversarial strategy to train the forgery detection CNN. Both Kniaz et al. [43] and Bi et al. [5] incorporate a generator that seeks to deceive the manipulation detector by conjuring more and more realistic manipulations. The SPAN localization technique [38] adopts ManTraNet features and applies a spatial attention network. The RRU-Net model [4] employs a modified U-Net for splicing detection instead.

All aforementioned algorithms require dense pixelwise masks for their training. In addition to the intense and expensive effort to annotate, it has been argued that creating a large representative dataset for supervised dense prediction is extremely difficult due to the nearly unlimited ways to alter an image [39, 43]. The synthetic tampered images constructed by applying random edits in [73] or generated in an adversarial fashion [5, 43] would be biased, if not limited, by the elementary operations or the source dataset used.

Splicing Detection from Device Fingerprint: There are strong evidences that every device that captures an image or every manual or automatic manipulation (GAN) editing leaves its trace on the image [50, 51, 53, 54]. Cozzolino et al. dubbed these signatures NoisePrint [20] and applied a siamese network consisting of denoising CNN to learn these noiseprints from the image using camera ids. Bondi et al. [10] instead utilized the deep features of image patches learned through camera identification task and applied a clustering algorithm to separate authentic parts from manipulated regions. The forensic graph approach of [54, 55] trains a CNN to explicitly distinguish between image blocks from different devices. Under the assumption that spliced patches possess a different fingerprint than the authentic region, this similarity function is utilized to locate manipulation through clustering. The EXIF-SC algorithm [39] aims

to learn representations of image patches such that the latent features from images with the same EXIF metadata are similar to each other and those from different EXIF metadata are different.

Models of [19, 58] have lately exhibited impressive performance to erase or swap the device/source trace that could deceive a manipulation detection mechanism. It would be interesting to investigate whether a similar approach can also succeed in erasing or swapping image fingerprints that our detection method relies on.

Frequency Domain Analysis for Manipulation Detection: Early studies on manipulation detections [22, 71] examine the double quantization effect hidden among DCT coefficients. Later studies explored hand-picked feature responses such as LBP [2, 34, 79] in conjunction with DCT to identify splicing. [33] also experiment with Markov features in DCT domain to expose tampering. Li et al. [47] propose a blind forensics approach based on DWT and SVD to detect duplicated regions as a sign of forgery.

Recent methods also involve the use of deep neural networks for predicting spliced regions. The CAT-Net approach [46] proposes to learn to predict localized regions using images in RGB and DCT domains. A follow-up study [45] trains a network to focus on JPEG compression artifacts in the DCT domain for learning to localize spliced regions.

Artificial Fakes and their Detections: There have been numerous works on generating deep fakes through generative networks e.g., GANs [8, 40, 56]. A very insightful work by Marra et al. [53] demonstrated that GAN also leave their fingerprint on the artificially generated images. Yu et al. [75] presented an algorithm to learn the GAN signature using a CNN. A subsequent study reported remarkable success in identifying source-specific artifacts in GAN generated images [24].

GAN generated images have been shown to be relatively easier to detect [70]. While there is evidence that camera trace-based manipulation detection methods can spot automatically generated fakes [53], the converse has not yet been demonstrated. Although in this study, we have not experimented on GAN generated tampering, there is no conceptual obstruction preventing it from working on them.

Self Supervised Learning: Self-supervised learning generally learns a latent feature representation under the guidance from pretext tasks and contrastive losses. Examples of the pretext tasks comprise the classification of images transformed by data augmentation techniques, e.g., rotation [28, 77], colorization [78]. Utilization of contrastive loss and appropriate architecture paved the way to highly useful representation learning [13, 16, 30, 35]. The benefit of these representations have already been substantiated in core vision tasks, e.g., classification, object detection, and segmentation [14, 15, 74].

The works of [39,55] have already demonstrated the benefit of representation learning for splicing detection. Learning these representations from self-supervision would be hugely beneficial where device id or image metadata are not available. Huh et al. [39] indeed mentions an approach to learning latent features without using EXIF metadata. A siamese network – operating in the RGB domain – is trained to distinguish between the patches extracted from the different images. This model was shown to be less effective for manipulation detection/localization and the lack of sufficient and diverse training data needed for generalization was speculated to be the reason for its deficiency. In this work, we show that the performance of CNNs utilizing RGB information does not improve with the number and diversity of the training set. But a relatively simple CNN trained in a self-supervised manner from FT of images can indeed match or exceed the detection performance of EXIF-SC.

3. Self-supervised Signature Learning

The core concept behind our approach is to learn a latent space where representations of patches from the same image are closer to each other than those from different images. We learn this latent representation with a CNN through self-supervision from the FT of an image patch. In essence, the CNN learns to capture an image-specific signature in feature representation that is exploited during the inference for distinguishing the tampered regions from the authentic ones. In the next few sections, we elaborate on the input to our CNN, its training, and inference for splicing detection.

3.1. DFT for Learning Signature

Let p_j^k denote the j-th patch from image I^k. We utilize the information in the real valued part of the discrete Fourier transform (DFT) of p_j^k as input to our CNN model.

$$f_j^k(m,n) = \frac{1}{\sqrt{UV}} \sum_{u=0}^{U-1} \sum_{v=0}^{V-1} p_j^k(u,v)\, cos\Big\{2\pi\Big(\frac{mu}{U}+\frac{nv}{V}\Big)\Big\}, \tag{1}$$

for $m = 0, 1, \ldots, U-1$, $n = 0, 1, \ldots, V-1$ where U, V are the dimensions of p_j^k. The resulting f_j^k contains the coefficients of different basis functions at each of its pixel locations (m, n). For the computation of the DFT, we utilize the PyTorch [60] implementation of real valued fast Fourier transform (RFFT) algorithm [67]. This implementation removes the symmetric values of the power spectrum in real valued inputs. It is typical for the high frequency coefficients to be much smaller than those of low frequencies.

3.2. Model Architecture and Training

Given the RFFT $f_j^k, j = 1, \ldots, J$ of patches from images $I^k, k = 1, \ldots, K$, we wish to learn a representation or

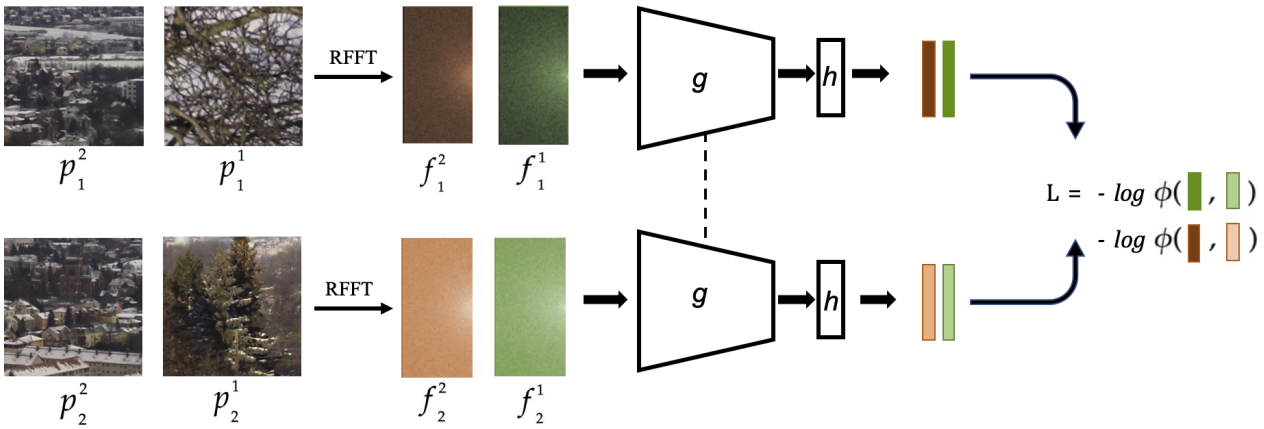

Figure 1. Proposed self-supervised training from RFFT of image patches. The pairs $\{p_1^1, p_2^1\}$ and $\{p_1^2, p_2^2\}$ are extracted from image I^1 and I^2 respectively. Green and brown colors were superimposed on their respective RFFTs $\{f_1^1, f_2^1\}$ and $\{f_1^2, f_2^2\}$ to distinguish between the two images. Different shades of the same color were used to indicate different patches from the same image. The contrastive loss L is calculated between representations learned by the backbone g and projector h. Best viewed in color.

encoding z_j^k by a CNN. The CNN consists of a backbone g followed by a projector h. we wish to learn a representation $z_j^k = h(g(f_j^k))$ such that:

- similarity between z_j^k and $z_{j'}^k$ of two patches from the same image k is high; and
- similarity between z_j^k and $z_{j'}^{k'}$ of patches extracted from different images k and k' is low.

We take advantage of the architecture and loss proposed in Chen et al. [13] (SimCLR) to design and train our model. However, we have modified the input, architecture, and loss function to suit our need to learn image specific signatures and to simplify the model. In particular, as opposed to different augmentation of the same image (e.g., resize, crop, color distortion, etc.), our model takes the RFFT of patches from the same or different images as input. The encoder g and projector h consist of a ResNet-18 backbone and a single linear layer respectively.

Each batch of examples in our training approach comprises B pairs of RFFT representations. Each of these pairs consists of RFFTs $\{f_j^k, f_{j'}^k\}$ computed from patches of the same image k. For any pair of representations $\{z_j^k, z_{j'}^{k'}\}$, we define the indicator vector $y^{kk'} = 1$ if $k = k'$ and 0 otherwise. The subsequent loss functions for pairs of encoding are defined as follows to facilitate learning the desired signature.

$$\phi_{jj'}^{kk'} = \frac{\exp(\text{sim}(z_j^k, z_{j'}^{k'})/\tau)}{\sum_{\kappa=1}^{B} \exp(\text{sim}(z_j^k, z_{j'}^{\kappa})/\tau)} \tag{2}$$

$$L(\{f_j^k, f_{j'}^k\}, y^{kk'}) = -\sum_{k,k'=1}^{B} y^{kk'} \log(\phi_{jj'}^{kk'}) \tag{3}$$

where $\text{sim}(a, b) = \frac{a^T b}{\|a\|\|b\|}$ is the cosine similarity and τ is the temperature weight. The loss function in Eqn 3 encourages the representations $z_j^k, z_{j'}^k$ from the patches of the same image k to be similar to each other and those from patches of different images to be different. The overall architecture and loss has also been depicted in Figure 1.

4. Image Splicing Detection and Localization

4.1. Patch Similarity to Response Map

After training, our model produces the latent representation z_j from the RFFT f_j of a patch p_j[1]. Our goal is to compute a pixelwise response map R for image I such that $R(u, v) = 1$ if $R(u, v)$ is manipulated and $R(u, v) = 0$ otherwise. We follow the standard practice of dividing the image [39, 54] of size $H \times W$ into overlapping patches p_j with a stride s. The patch consistency between all pairs of patches $\{p_j, p_{j'}\}$, $j, j' = 1, \ldots \lfloor \frac{H}{s} \rfloor \lfloor \frac{W}{s} \rfloor$, $j \neq j'$ are computed with cosine similarity $\text{sim}(z_j, z_j')$. The patch consistencies are aggregated to form the image level consistency, which we utilize as response R^k, by mean-shift based clustering and bilinear upsampling as proposed in [39]. Using cosine similarity as opposed to a dedicated network as used in [39, 54] significantly reduces the inference time when we consider the number of pairs of patches $\lfloor \frac{H}{s} \rfloor \lfloor \frac{W}{s} \rfloor \times \lfloor \frac{H}{s} \rfloor \lfloor \frac{W}{s} \rfloor$ to be compared.

4.2. Detection & Localization from Response Map

Given the response image R, we devise two approaches to detect whether an image has been manipulated. The first, dubbed as SpAvg, averages R spatially and detects an image to be tampered with by thresholding $\text{mean}(R)$. In the second approach, PctArea, a binary mask is created by thresholding $R > \delta_b$. We then transform the resultant binary mask to the fraction of pixels that are masked to get the detection score, that is, $\rho_b = \frac{\sum R > \delta_b}{HW}$. For localization, a binary mask

[1] Dropping superscript k to remove clutter and confusion.

is created by thresholding $R > \delta_l$ to delineate the spliced area.

In accordance with common practice [39], the values of response map R are inverted by $1 - R$ *before detection* if mean$(R) > 0.5$, indicating the area of the spliced region is larger than that of authentic region. This is based on the assumption that spliced region should be smaller than the pristine part of the image.

5. Experiments & Results

5.1. Implementation Details

We use a ResNet-18 [36] as the backbone g and project to a 256 dimensional representation through a single layer h. The input f_j^k to ResNet-18 is computed from image patch p_j^k by the PyTorch implementation of RFFT. For the self-supervised contrastive training, each batch consists of 256 pairs of RFFT coefficients, and temperature τ is set to 0.9. The model is optimized using ADAM [41] with $\alpha = 0.9, \beta = 0.99$. The learning rate was decayed from 0.001 to $1e-5$ via cosine annealing after an initial warmup period.

In all experiments, the size of image crops p_j is 128×128. During inference, the patches are cropped with a stride of 64 pixels (i.e., 50% patch overlap).

5.2. Datasets

Training set: Images from 5 public datasets have been used to train our model: Dresden [29] (16961 images), Vision [66] (34427 images), Socrates [26] (8742 images), FODB [32] (23106 images), Kaggle [1] (2750 images). Although these datasets were collected for camera/device identification purposes, *we do not use the camera ids* in any part of our training. From these datasets, we gathered 85984 images captured by different devices with diverse appearances, and scenes from various locations around the world. From each of these images, 100 patches were cropped arbitrarily to create the training set. During training, we randomly select 256 images and then select 2 patches from the 100 pre-cropped patches of the same image to generate a batch of training pairs.

Test set: Our algorithm was tested on the popular Columbia [37] (363 images, 180 spliced), Carvalho/DSO [21] (200 images, 100 spliced) and Realistic Tampering (RT)/Korus [44] (440 images, 220 spliced) datasets that provide the groundtruth masks for splicing operation. One can observe from inspecting the datasets that manipulations in Carvalho/DSO are more deceiving than the spliced images in Columbia. RT provides a multivalued mask for each image, with different values corresponding to the spliced images and the subsequent alterations. We mark all nonzero values as manipulated regions.

5.3. Evaluation

Our evaluation setting attempts to emulate the scenario of a real-life application as closely as possible. To achieve this and, to promote reproducibility, we try to use standard evaluation measures (and their public implementations) and keep the configuration/parameters fixed as much as possible.

In practical applications, a forensic solution will use a fixed value for thresholds used for recognizing and localizing tampered image regions. It is not reasonable to assume, and we are not aware of, a method to select image specific thresholds for real world forensic applications. However, although not ideal, it is not impractical to allow δ_b and δ_l to be different for detection and localization respectively, because these two procedures will perhaps be executed sequentially. The detection performances of our method as baselines are computed with fixed δ_b for all images and the localization performances are calculated with a fixed δ_l for all spliced images.

For splicing detection, we report the average precision (AP) for the binary task of classifying whether an image is tampered or authentic. This value is computed from the outputs of two detection techniques, SpAvg and PctArea, against the binary ground truth label using a standard AP implementation (from scikit-learn).

For splicing localization, the output binary masks are compared with GT masks to compute true & falses positive (TP & FP) and false negative (FN) pixels. We adopt the standard Matthew's coefficient MCC = $\frac{TP \times TN}{\sqrt{(TP+TP)(TP-FN)(TN+FP)(TN+FN)}}$, F1 score and Intersection over Union (IoU) measure averaged over each dataset to evaluate localization accuracy. The optimal detection and localization thresholds δ_b and δ_l are calculated empirically for each dataset and method and are kept fixed for all images in one dataset (δ_b may not necessarily be equal to δ_l). As a result, the values reported in the following sections may vary from those in the past studies.

5.4. Results

We show the forgery detection and localization accuracies of our and baseline algorithms on the 3 test datasets in Tables 1 and 2 respectively. The performance of the proposed algorithm is compared against 3 baselines: 1) EXIF-SC [39] algorithm for learning representation given EXIF metadata, 2) forensic graph (FG) [54] algorithms that learn device signatures from camera id, 3) pixelwise prediction by MantraNet trained in a fully supervised manner [73]. The detection and localization results of these methods were computed from their publicly available implementations and evaluated with the measures explained in Section 5.3. Among the baselines, EXIF-SC performance is more relevant than other methods because, like the pro-

Table 1. Manipulation detection performance comparison on Columbia, DSO/Carvalho, RT/Korus datasets.

Alg	Supervision	Det Methd	Columbia		DSO/ Carvalho		RT/ Korus	
			δ_b	AP	δ_b	AP	δ_b	AP
MantraNet [73]	Dense GT	SpAvg		0.712	-	0.906	-	0.535
		PctArea	0.005	0.835	0.5075	0.935	0.990	0.633
FG [54]	Camera ID	SpecG	-	0.955	-	0.947	-	0.688
EXIF-SC [39]	EXIF Metadata	SpAvg	-	0.962	-	0.75	-	0.534
		PctArea	0.185	0.945	0.47	0.784	0.46	0.545
Proposed	Self consist.	SpAvg	-	0.871	-	0.837	-	0.538
		PctArea	0.25	0.918	0.285	0.946	0.291	0.537

Table 2. Forgery localization performance comparison on Columbia, DSO/Carvalho, RT/Korus

Method	Supervision	Columbia				DSO/ Carvalho				RT/ Korus			
		δ_l	MCC	F1	IOU	δ_l	MCC	F1	IOU	δ_l	MCC	F1	IoU
MantraNet [73]	Dense GT	0.005	0.198	0.486	0.302	0.50	0.349	0.363	0.528	0.99	0.07	0.08	0.25
		0.10	0.486	0.599	0.596	0.40	0.369	0.392	0.545	0.41	0.190	0.208	0.424
FG [54]	Camera ID	0.30	0.860	0.884	-	0.25	0.744	0.760	-	0.20	0.265	0.274	-
EXIF-SC [39]	EXIF Metadata	0.18	0.778	0.837	0.793	0.47	0.358	0.758	0.5	0.46	0.077	0.118	0.158
		0.22	0.785	0.837	0.803	0.36	0.381	0.795	0.519	0.16	0.109	0.126	0.244
Ours	Self consist.	0.25	0.481	0.524	0.572	0.285	0.514	0.532	0.594	0.29	0.05	0.1	0.114
		0.18	0.71	0.786	0.738	0.2	0.65	0.67	0.7	0.12	0.154	0.152	0.3

Table 3. Inference time (sec/image) comparison.

Alg	Columbia (sec/img)	DSO (sec/img)
FG [54] detect	0.3	3.63
FG [54] localize	0.75	3.97
EXIF-SC [39]	81.59	99.15
ManTraNet [73]	0.707	3.729
Ours	0.35	8.05

posed approach, it does not utilize the device ids.

The detection accuracy is calculated by comparing the binary groundtruth label of the image (authentic vs fake) with the prediction generated by SpAvg and PctArea for EXIF-SC, MantraNet, and the proposed method. For FG, we use the output of the spectral gap technique with the crop size of 128×128 and stride $s = 64$. The detection performances of the baselines and proposed method are reported in Table 1. We also mention the type of groundtruth annotation needed for training the CNNs in each algorithm.

As displayed in Table 1, the proposed method achieves similar or better AP values than EXIF-SC on DSO/Carvalho and RT/Korus datasets but trails in Columbia dataset by 0.03. Our method exhibit better performance with PctArea technique than SpAvg for forgery detection. The optimal detection threshold δ_b^* for our model resides within a small range $[0.25, 0.291]$, which implies consistency in output response values on different test sets. FG [54] consistently outperformed all methods in all datasets suggesting that source ids contribute to performance improvement. As anticipated earlier, MantraNet [73] was unable to generalize well on all datasets. We believe this is due to the inability for the artificially generated training set to encompass the variations in forgeries that appear in real world.

It is worth mentioning here that, our proposed method applies cosine similarity which is a simpler operation than the MLPs used to compute patch similarity in EXIF-SC and FG. The fact that our method attains close or superior performances to those of EXIF-SC and FG with simpler patch consistency function demonstrates the strength of the representations learned by the proposed approach. This provides strong evidence that self-supervised learning of representation from FT content is an effective strategy for confronting image forgeries.

For localization, we generate the binary prediction mask using two threshold values of δ_l. One of the output masks was produced by setting $\delta_l = \delta_b^*$ where δ_b^* is the best threshold for manipulation detection (refer to Table 1). The other prediction map was computed by searching δ_l over a range (centered at δ_b^*) that yield the highest MCC score. The performance of the proposed method for localization conforms to that for detection – it achieves similar or higher accuracy than those of EXIF-SC, MantraNet at the best threshold value (Table 2). Operating on two different threshold values for detection and localization is not an impractical decision to make as we discussed in Section 5.3.

Figure 2 displays qualitative results from the proposed method and baselines. Our model performs as good as or better than baselines in these images. One can notice a few small false positive blobs on the output mask of our method

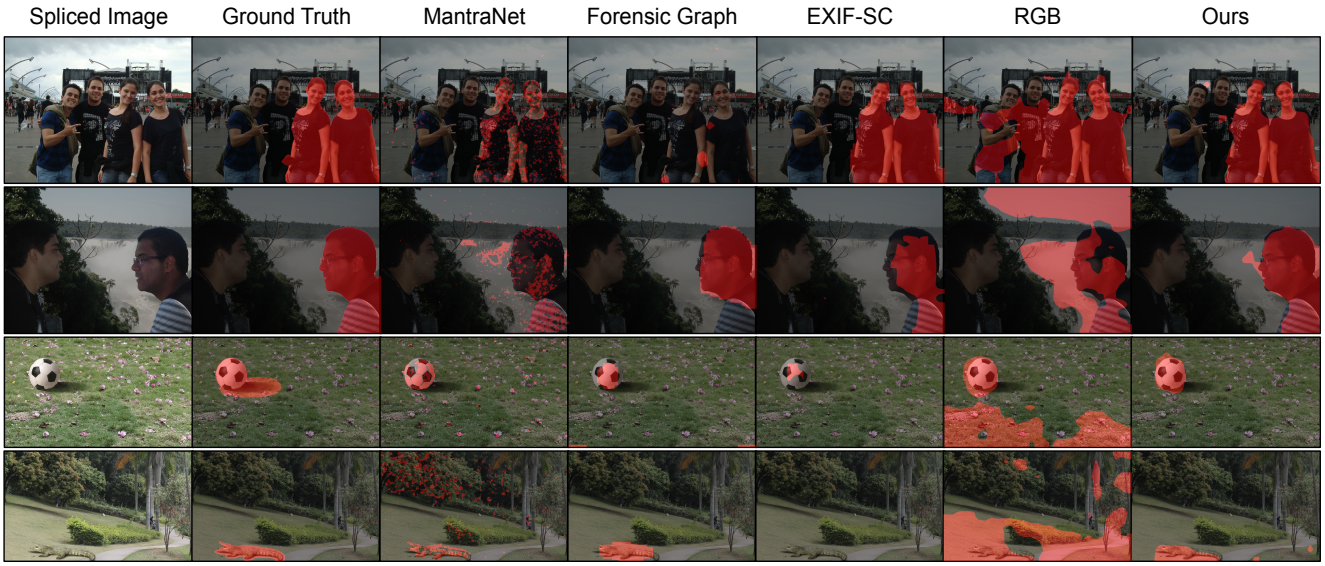

Figure 2. Localization results on DSO/Carvalho (row 1 and 2) and RT/Korus (row 3 and 4) datasets. Our self-supervised approach performs comparably, if not better than other methods. Best viewed in color.

Table 4. Manipulation detection performances of the RGB, Fusion model with same architecture and RFFT models with different architecture. All models were learned with self-supervision.

Model	Det Methd	Columbia		DSO/ Carvalho		RT/ Korus	
		δ_b	AP	δ_b	AP	δ_b	AP
RGB	SpAvg	-	0.69	-	0.836	-	0.514
	PctArea	0.0947	0.678	0.275	0.88	0.052	0.531
RGB-RFFT	SpAvg	-	0.89	-	0.88	-	0.531
	PctArea	0.20	0.935	0.20	0.955	0.247	0.537
ResNet50	SpAvg	-	0.852	-	0.852	-	0.525
	PctArea	0.24	0.96	0.24	0.907	0.18	0.531
SimCLR	SpAvg	-	0.883	-	0.874	-	0.524
	PctArea	0.12	0.94	0.12	0.89	0.12	0.53

on the top 2 images from DSO/Carvalho dataset. Our model leads to an F1 accuracy lower than but IoU values higher than those of EXIF-SC. This suggests our method produces more false positive pixels than EXIF-SC on DSO/Carvalho but the sizes of these false positive pixel blobs are small and can be removed by subsequent post-processing based on, e.g., size or number of regions.

5.5. Inference Speed

In Table 3, we report the average time to detect and localize the spliced area in each image in Columbia and DSO/Carvalho datasets. The inference speed was calculated for all algorithms on the same machine with an NVIDIA V100 GPU. We used the same image block size 128×128 and stride $s = 64$ for the proposed, FG and EXIF-SC methods. Since FG uses different techniques for detection (spectral gap) and localization (community detection), one must run both inference operations to generate values reported in Tables 1 and 2.

Our proposed approach is at least an order of magnitude faster than EXIF-SC. This is due to the adoption of a lighter backbone (ResNet-18) and the use of cosine similarity for inference. A closer examination revealed that 90% of the inference time of our method is spent on the mean-shift clustering algorithm. One can utilize an efficient clustering/agglomerative method or implementation to further reduce the latency of the proposed technique.

5.6. Analysis & Ablations

5.6.1 RFFT vs RGB

For our first ablation experiment, we train two models: one takes the RGB values of the image patches as input while the other is a fusion model that operates on both the RGB values and the RFFT values of the image patches. The RGB model has the same architecture as described in Section 3.2. In the fusion model, the RGB and RFFTs values are processed by two different backbones and projections and then are combined at the end to yield the final representation (late fusion). Both models are trained with the same contrastive

Table 5. Forgery localization accuracy of fusion (RGB-RFFT) model.

Method	Columbia				DSO/ Carvalho				RT/ Korus			
	δ_l	MCC	F1	IoU	δ_l	MCC	F1	IoU	δ_l	MCC	F1	IoU
RGB-RFFT	0.2	0.5	0.65	0.6	0.2	0.544	0.578	0.68	0.24	0.032	0.118	0.48
	0.14	0.642	0.75	0.696	0.16	0.645	0.68	0.736	0.12	0.137	0.18	0.42

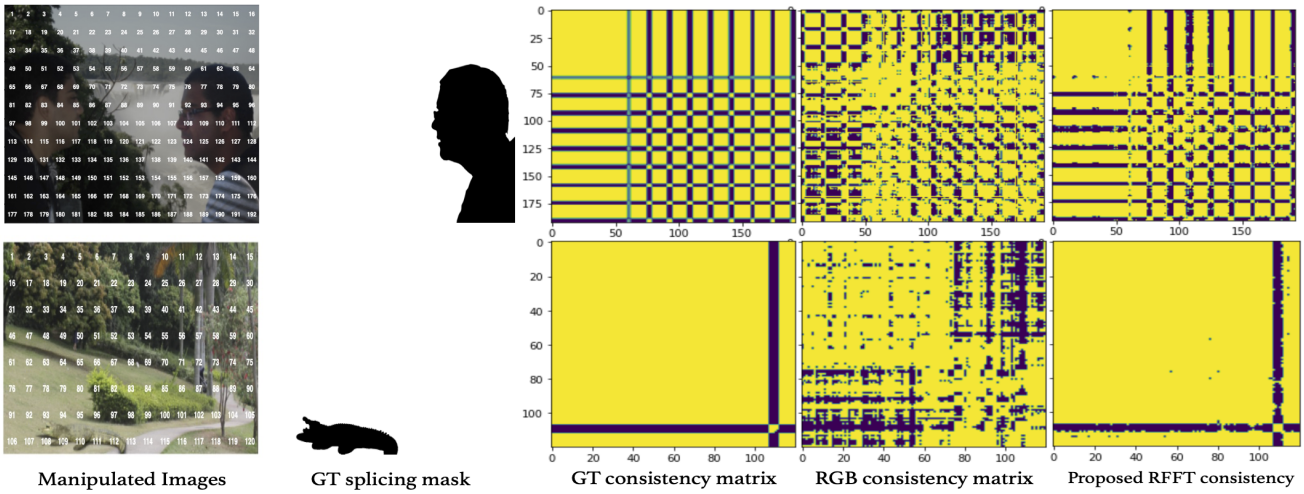

Figure 3. Left to right: a sample fake image, its GT mask, and consistency matrices from GT, cosine similarity from RGB model, and that from RFFT model. The numbers on input image indicate patch indices. Yellow indicates high similarity, purple indicates low consistency (green should be perceived as purple, was created as an artifact of downscaling). Best viewed in color.

loss (Eqn 3) and optimization technique.

The tampering detection results from the proposed RFFT based model are compared in Table 4 with the RGB and Fusion model. It is interesting to observe that the fusion model, which combines information from RGB and RFFT, achieves slight improvement over the proposed RFFT based model. However, as Table 5 shows, the fusion model was unable to achieve the same localization quality of the RFFT model. The fusion model also increases the model size by almost a factor of two with $\leq 2\%$ improvement in detection accuracy.

We also compare qualitatively the patch consistency matrices produced by the cosine similarities from ResNet-18 trained on RGB and RFFT in Figure 3. The consistency values from every image block to all other blocks are computed from the groundtruth labels, cosine similarity from RGB model, and the RFFT model respectively (yellow = high similarity). It is evident from the consistency matrices that, while the proposed RFFT can correctly distinguish the manipulated patches from authentic ones, the RGB based model is confused by appearance features. For example, the RGB model appears to be separating the vegetation, waterfall, and sky in the pristine part of the image in the top row of Figure 3. As a result, the output from RGB based model produces large false positive detections, see the column labeled RGB in Figure 2.

5.6.2 Model Variation

We have also tested out the model by replacing the backbone network to ResNet-50 instead of ResNet-18 and with the exact model proposed in the SimCLR study [13]. The detection performances of these models are presented in Table 4. Although it may be possible to match the accuracy of the proposed architecture with the further tuning of hyperparameters and training procedures, we speculate the improvement may not justify the costs ResNet-50 based models incur.

6. Conclusion

This paper presents an effective approach for training a splicing detection/localization CNN in a self-supervised fashion from FT of images. Given the FT, the model is designed to learn an image fingerprint to be exploited to identify spliced regions extracted from different images. Our experiments suggest that the proposed model learned under self-supervision can achieve the accuracy and speed of multiple standard algorithms on different benchmarks. Our findings will not only facilitate model training in scenarios where the camera and image metadata are not available but also enable expanding the training set to learn a more robust network. We hope our work will encourage further research in similar directions toward robust and scalable manipulation detection techniques.

Acknowledgement: We thank Aurobrata Ghosh for helpful discussions and Ya-Fang Shih for sharing an earlier version of evaluation scripts.

References

[1] Kaggle camera model identification. https://www.kaggle.com/c/sp-society-camera-model-identification/overview. 5

[2] Amani A. Alahmadi, Muhammad Hussain, Hatim Aboalsamh, Ghulam Muhammad, and George Bebis. Splicing image forgery detection based on dct and local binary pattern. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 253–256, 2013. 3

[3] Jawadul H. Bappy, Amit K. Roy-Chowdhury, Jason Bunk, Lakshmanan Nataraj, and B. S. Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 2

[4] Xiuli Bi, Yang Wei, Bin Xiao, and Weisheng Li. Rru-net: The ringed residual u-net for image splicing forgery detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 30–39, 2019. 1, 2

[5] Xiuli Bi, Zhipeng Zhang, and Bin Xiao. Reality transform adversarial generators for image splicing forgery detection and localization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[6] Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Improved dct coefficient analysis for forgery localization in jpeg images. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2444–2447, 2011. 2

[7] Tiziano Bianchi and Alessandro Piva. Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3):1003–1017, 2012. 2

[8] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. 2, 3

[9] Blog. New algorithms to spot fake pictures for insurance claim verification. *Marsh McLennan Agency*. 1

[10] Luca Bondi, Silvia Lameri, David Güera, Paolo Bestagini, Edward J. Delp, and Stefano Tubaro. Tampering detection and localization through clustering of camera-based cnn features. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017. 2

[11] Carolyn Cage. Confessions of a retoucher: how the modelling industry is harming women. *The Sydney Morning Herald*. 1

[12] Chang Chen, Zhiwei Xiong, Xiaoming Liu, and Feng Wu. Camera trace erasing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 3, 4, 8

[14] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 3

[15] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 3

[16] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15745–15753, 2021. 2, 3

[17] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Splicebuster: A new blind image splicing detector. In *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2015. 1

[18] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Extracting camera-based fingerprints for video forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1

[19] D. Cozzolino, J. Thies, A. Rossler, M. Niesner, and L. Verdoliva. Spoc: Spoofing camera fingerprints. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society. 3

[20] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, 15:144–159, 2020. 1, 2

[21] Tiago José de Carvalho, Christian Riess, Elli Angelopoulou, Hélio Pedrini, and Anderson de Rezende Rocha. Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security*, 8(7):1182–1194, 2013. 5

[22] Hany Farid. Exposing digital forgeries from jpeg ghosts. *IEEE Transactions on Information Forensics and Security*, 4:154–160, 2009. 2, 3

[23] Hany Farid and Siwei Lyu. Higher-order wavelet statistics and their application to digital forensics. In *2003 Conference on Computer Vision and Pattern Recognition Workshop*, volume 8, pages 94–94, 2003. 1

[24] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 3247–3258. PMLR, 2020. 2, 3

[25] Jessica Fridrich, David Soukal, and Jan Lukás. Detection of copy-move forgery in digital images. *Int. J. Comput. Sci. Issues*, 3:55–61, 01 2003. 1

[26] Chiara Galdi, Frank Hartung, and Jean-Luc Dugelay. Socrates: A database of realistic data for source camera recognition on smartphones. In *International Conference on Pattern Recognition Applications and Methods*, 2019. 5

[27] Nancy Gibbs. Crime: O.j. simpson: End of the run. *Time*, 143(26). 1

[28] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ArXiv*, abs/1803.07728, 2018. 3

[29] Thomas Gloe and Rainer Böhme. The 'dresden image database' for benchmarking digital image forensics. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, page 1584–1590, New York, NY, USA, 2010. Association for Computing Machinery. 5

[30] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. 2, 3

[31] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. Faking sandy: Characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd International Conference on World Wide Web*, page 729–736, New York, NY, USA, 2013. Association for Computing Machinery. 1

[32] Benjamin Hadwiger and Christian Riess. The forchheim image database for camera identification in the wild. In *ICPR Workshops*, 2020. 5

[33] Jong Goo Han, Tae Hee Park, Yong Ho Moon, and Il Kyu Eom. Efficient Markov feature extraction method for image splicing detection using maximization and threshold expansion. *Journal of Electronic Imaging*, 25(2):1 – 8, 2016. 3

[34] Mahdi Hariri. Image-splicing forgery detection based on improved lbp and k-nearest neighbors algorithm. *Electronics Information and Planning*, 3, 09 2015. 3

[35] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 2, 3

[36] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5

[37] Yu-Feng Hsu and Shih-Fu Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *Interational Conference on Multimedia and Expo (ICME)*, Toronto, Canada, July 2006. 5

[38] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. SPAN: spatial pyramid attention network forimage manipulation localization. In *European Conference on Computer Vision ECCV*, 2020. 1, 2

[39] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1, 2, 3, 4, 5, 6

[40] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 2, 3

[41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*. 5

[42] Melissa Kirby. Food photography and manipulation in advertising: Why do we accept knowingly being lied to? In *Truly Deeply - Brand Agency Melbourne*. 1

[43] Vladimir V. Kniaz, Vladimir Knyaz, and Fabio Remondino. The point where reality meets fantasy: Mixed adversarial generators for image splice detection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, 2019. 1, 2

[44] Pawel Korus and Jiwu Huang. Multi-scale analysis strategies in prnu-based tampering localization. *Trans. Info. For. Sec.*, 12(4):809–824, apr 2017. 5

[45] Myung-Joon Kwon, Seung-Hun Nam, In-Jae Yu, Heung-Kyu Lee, and Changick Kim. Learning jpeg compression artifacts for image manipulation detection and localization, 2021. 3

[46] Myung-Joon Kwon, In-Jae Yu, Seung-Hun Nam, and Heung-Kyu Lee. Cat-net: Compression artifact tracing network for detection and localization of image splicing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 375–384, January 2021. 3

[47] Guohui Li, Qiong Wu, Dan Tu, and Shaojie Sun. A sorted neighborhood approach for detecting duplicated regions in image forgeries based on dwt and svd. In *2007 IEEE International Conference on Multimedia and Expo*, pages 1750–1753, 2007. 2, 3

[48] Zhouchen Lin, Junfeng He, Xiaoou Tang, and Chi-Keung Tang. Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis. *Pattern Recognit.*, 42:2492–2501, 2009. 2

[49] Jan Lukás and Jessica Fridrich. Estimation of primary quantization matrix in double compressed jpeg images. In *Digital Forensics Research Workshop*, 2003. 2

[50] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, 2006. 1, 2

[51] Jan Lukás, Jessica Fridrich, and Miroslav Goljan. Detecting digital image forgeries using sensor pattern noise - art. no. 60720y. *Proceedings of SPIE - The International Society for Optical Engineering*, 6072:362–372, 02 2006. 1, 2

[52] Carla Marinucci. Doctored kerry photo brings anger, threat of suit. *San Francisco Chronicle*. 1

[53] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi. Do gans leave artificial fingerprints? In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, Los Alamitos, CA, USA, mar 2019. IEEE Computer Society. 1, 2, 3

[54] O. Mayer and M. C. Stamm. Exposing fake images with forensic similarity graphs. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1049–1064, 2020. 1, 2, 4, 5, 6

[55] Owen Mayer and Matthew C. Stamm. Forensic similarity for digital images. *Trans. Info. For. Sec.*, 15:1331–1346, jan 2020. 1, 2, 3

[56] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative ad-

versarial networks. In *International Conference on Learning Representations*, 2018. 2, 3

[57] Yakun Niu, Benedetta Tondi, Yao Zhao, and Mauro Barni. Primary quantization matrix estimation of double compressed jpeg images via cnn. *IEEE Signal Processing Letters*, 27:191–195, 2020. 2

[58] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. In *Advances in Neural Information Processing Systems*, 2020. 3

[59] Zachariah B. Parry. Digital manipulation and photographic evidence: Defrauding the courts one thousand words at a time. *University of Illinois Journal of Law, Technology Policy*, 2009. 1

[60] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 3

[61] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, jul 2003. 1

[62] A.C. Popescu and H. Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, 2005. 1

[63] Alin C. Popescu and Hany Farid. Statistical tools for digital forensics. In Jessica Fridrich, editor, *Information Hiding*, pages 128–147, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 1

[64] Elizabeth G. Porter. Taking images seriously. *Columbia Law Review*, 114(7):1687–1782, 2014. 1

[65] Ronald Salloum, Yuzhuo Ren, and C.-C. Jay Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *J. Vis. Commun. Image Represent.*, 51:201–209, 2018. 1

[66] Dasara Shullani, Marco Fontani, Massimo Iuliani, Omar Al Shaya, and Alessandro Piva. Vision: a video and image dataset for source identification. *EURASIP Journal on Information Security*, 2017:1–16, 2017. 5

[67] H. Sorensen, D. Jones, M. Heideman, and C. Burrus. Real-valued fast fourier transform algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(6):849–863, 1987. 3

[68] Luisa Verdoliva. Media forensics and deepfakes: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 14:910–932, 2020. 1

[69] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018. 1

[70] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot...for now. In *CVPR*, 2020. 2, 3

[71] Wei Wang, Jing Dong, and Tieniu Tan. Exploring dct coefficient quantization effects for local tampering detection. *IEEE Transactions on Information Forensics and Security*, 9(10):1653–1666, 2014. 2, 3

[72] Yue Wu, Wael Abd-Almageed, and Prem Natarajan. Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection. In *Proceedings of the 25th ACM International Conference on Multimedia*, page 1480–1502, New York, NY, USA, 2017. Association for Computing Machinery. 1

[73] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 5, 6

[74] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. 2021. 3

[75] Ning Yu, Larry Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7555–7565, 2019. 3

[76] M. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris. Detecting image splicing in the wild (web). In *2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1–6, Los Alamitos, CA, USA, jul 2015. IEEE Computer Society. 1

[77] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1476–1485, 2019. 2, 3

[78] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 3

[79] Yujin Zhang, Chenglin Zhao, Yiming Pi, Shenghong Li, and Shilin Wang. Image-splicing forgery detection based on local binary patterns of dct coefficients. *Sec. and Commun. Netw.*, 8(14):2386–2395, sep 2015. 3

[80] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning rich features for image manipulation detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 1