TruFor: Leveraging all-round clues for trustworthy image forgery detection and localization

Fabrizio Guillaro[1] Davide Cozzolino[1] Avneesh Sud[2] Nicholas Dufour[2] Luisa Verdoliva[1]

[1]University Federico II of Naples [2]Google Research

Abstract

In this paper we present TruFor, a forensic framework that can be applied to a large variety of image manipulation methods, from classic cheapfakes to more recent manipulations based on deep learning. We rely on the extraction of both high-level and low-level traces through a transformer-based fusion architecture that combines the RGB image and a learned noise-sensitive fingerprint. The latter learns to embed the artifacts related to the camera internal and external processing by training only on real data in a self-supervised manner. Forgeries are detected as deviations from the expected regular pattern that characterizes each pristine image. Looking for anomalies makes the approach able to robustly detect a variety of local manipulations, ensuring generalization. In addition to a pixel-level localization map and a whole-image integrity score, our approach outputs a reliability map that highlights areas where localization predictions may be error-prone. This is particularly important in forensic applications in order to reduce false alarms and allow for a large scale analysis. Extensive experiments on several datasets show that our method is able to reliably detect and localize both cheapfakes and deepfakes manipulations outperforming state-of-the-art works. Code is publicly available at https://grip-unina.github.io/TruFor/.

1. Introduction

Manipulating images has never been easier, with new powerful editing tools appearing by the day. These new opportunities stimulate the creativity of benign and malicious users alike. Previously, crafting a multimedia disinformation campaign required sophisticated skills, and attackers could do little more than copy, replicate or remove objects in an image, classic forms of image manipulations also known as “cheapfakes”. With the explosive growth of deep learning, image manipulation tools have become both easier to use and more powerful, allowing users to generate on-the-fly images of persons that do not exist or to realize credible deepfakes. Diffusion models enable the creation of realistic image edits using natural language prompts, photorealistically adapting the inserted manipulation to the style and lighting of the context [1, 33].

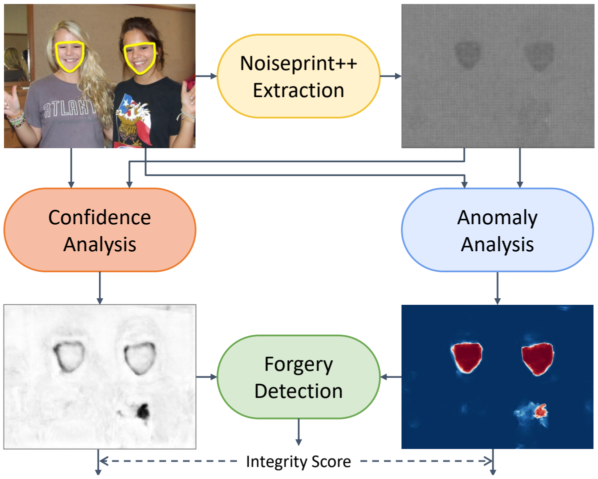

Figure 1. TruFor detects and localizes image forgeries (in yellow). It is based on the extraction of a learned noise-sensitive fingerprint, Noiseprint++, which is combined with the RGB image to output an anomaly localization map. Noiseprint++ is also used jointly with the image to compute the confidence map, which estimates the less reliable regions of the anomaly heatmap (black areas), e.g. the false positive region in lower right. The confidence and anomaly maps are then used together to produce a global integrity score.

The risks posed by such tools in the wrong hands are obvious. Indeed, in recent years there has been a growing interest on the part of governments and funding agencies in developing forensic tools capable of countering such attacks. A major focus is on local image edits, particularly partial modifications that change the image semantics (for example the partially manipulated image in Fig. 1, where the two real faces have been replaced with GAN-generated ones [26]). Multimedia forensics and related scientific fields have seen a rapid increase in activity in response to such challenges, with a large number of methods and tools proposed for image forgery detection and localization [38]. Despite considerable advances in the area, current SOTA detectors are not yet performant enough for in-the-wild deployment, due mainly to deficiencies in several areas subject to intense research: *i)* limited generalization; *ii)* limited robustness; *iii)* insufficient detection performance.

Limited generalization is the inability of detectors to cope with out-of-distribution manipulations. Some detectors are built to exploit well-defined low-level features, e.g., traces of JPEG compression, demosaicking or interpolation [2, 6, 34], while others are typically developed to work well only on specific types of manipulations, like splicing [25, 37]. In addition, in a realistic scenario images also undergo numerous forms of non-malicious degradation, (e.g. recompression, resizing, etc) - also called *laundering*. For example, social networks compress and resize uploaded images, both of which can easily remove forensic traces. Finally, most SOTA methods perform image forgery localization, leaving detection as an afterthought [11], which is typically derived as a global integrity score from the localization heatmap itself [22, 36, 42]. Few methods address the detection task directly [8, 31, 39, 46]. As a result, detection accuracy is poor, with a high false alarm rate. In a realistic setting where manipulated images are rare, such performance could cause more problems than it solves, with false positives drastically outnumbering true positives.

This work addresses such shortcomings, with a focus on robust detection under varied manipulations. Our aim is to first establish whether the image under analysis has been manipulated or not, and subsequently consider forgery localization only for images where a forgery has been detected. To perform in a real-world scenario where images undergo many post-processing steps that may attenuate forensic traces, our design was guided by the need to leverage information at multiple scales (both low and high-level features) even in complex scenarios. Our framework estimates a confidence map that associates localization results with region-specific uncertainty, allowing many potential false alarms to be rejected. The block diagram of our method is presented in Fig. 1. Overall, in this work we make the following key contributions:

- we propose a new framework, TruFor, which outputs a global integrity score, an anomaly-based localization map and an associated confidence map;
- we propose a new noise-sensitive fingerprint, Noiseprint++, with enhanced robustness to image laundering;
- we combine low-level and high-level evidence to perform anomaly analysis, which together with the confidence analysis provide more reliable decisions;
- we carry out extensive experiments on several benchmarks, considering new and challenging scenarios, and demonstrate that our method achieves state-of-the-art performance in both detection and localization tasks.

2. Related Work

Forensic artifacts. Low-level artifacts are caused by the in-camera acquisition process, such as the sensor, the lens, the color filter array or the JPEG quantization tables. In all cases, these are very weak traces, that can be highlighted by suppressing the image content by means of high-pass filters or denoising. The most common filters used for this task are the spatial rich models (SRM) [16], often included as a pre-processing step in some CNN models for forensic analysis. In [35] a set of around 30 fixed high-pass filters are used, instead in [3] the high-pass filters are learnt during training. These fixed and trainable filters have been used in many other subsequent works to perform a noise sensitive analysis [8, 21, 42, 44, 47]. A different perspective is considered in [12], where the extraction of low-level artifacts is carried out by learning a sort of "camera model fingerprint", the noiseprint, that bears traces of in-camera processing steps. When a manipulation is present, the noiseprint structure is absent and this anomaly is interpreted as a forgery. In this work we leverage noiseprint and further enhance it so as to make it work in more challenging scenarios.

In general, low-level features are combined with high-level ones to carry out a more effective detection. Pioneering work in the field is the two-branch approach proposed in [47], where the features of the noise and RGB stream are combined together through bilinear pooling. Other works also propose late fusion [8], while others [21, 39, 42] perform early fusion or even middle fusion [24]. We belong to this last category, but use an approach that fuses noise and RGB channels using cross-modal feature calibration [28].

Forgery detection vs localization. The majority of the state-of-the-art methods focus on image localization, with architectures often inspired by semantic segmentation, and detection is a byproduct of such analysis [11]. The integrity score is computed by a suitable post-processing of the localization heatmap aimed at extracting a global decision statistic, such as the average or the maximum value of the heatmap [5, 22, 42]. Only a few works explicitly treat the detection problem. In particular, some recent approaches [8,29,39,46] jointly train the model both for localization and detection through suitable losses at image-level. In [39, 46] global average pooling is applied to the middle features, while in [8] max average pooling is carried out on the localization heatmap. A different perspective can be found in [31], where it is proposed to analyze the whole image avoiding resizing (so as not to lose precious forensics traces) through a gradient checkpointing technique, that helps for the joint optimization of patch-level feature extraction and image-level decision.

Different from current literature, in this paper we explicitly design a forgery detection module that takes as input the anomaly-based map and the confidence map. This addi-

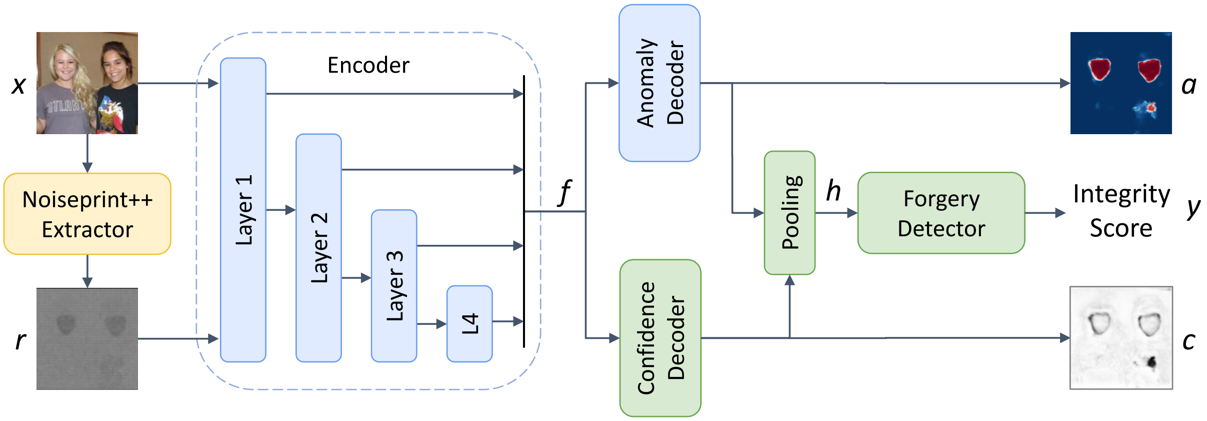

Figure 2. TruFor framework. The Noiseprint++ extractor takes the RGB image to obtain a learned noise-sensitive fingerprint. The encoder uses both the RGB input and Noiseprint++ for jointly computing the features that will be used by the anomaly decoder and the confidence decoder for pixel-level forgery localization and confidence estimation, respectively. The forgery detector exploits the localization map and the confidence map to make the image-level decision. The different colors identify the modules learned in each of the three training phases.

tional input is crucial to reduce the number of false alarms on pristine data and provide a more trustworthy tool.

Reliability in multimedia forensics Designing reliable detectors is important in several computer vision applications, however, it is even more critical for our task, since forensic traces are often imperceptible to visual inspection. The problem is even more relevant when deep learning based methods are used, since image forensics tools are challenged by out-of-distribution data [38]. In the context of JPEG artifacts and resampling analysis, initial efforts to develop reliable forensics detectors are carried out in [4, 30], where it is proposed to use Bayesian neural networks that provide an uncertainty range with every prediction. In this way, the user can quantify trust on the final prediction.

Inspired by [9], our work aims at making a further step in this direction and proposes a method using external uncertainty quantification [18] to design a confidence map from the anomaly localization heatmap.

3. Method

In this Section we begin by presenting an overview of TruFor, which is illustrated in Fig. 2. Subsequent subsections will provide the details of each component. First of all, from the input RGB image, x, we extract its Noiseprint++, $r = \mathcal{R}(x)$, a learned noise-sensitive fingerprint of the same resolution as x. Then, both x and r feed two networks that extract the anomaly map a and the confidence map c of the image. These networks have the same encoder-decoder architecture, with a shared encoder that extracts suitable dense features, $f = \mathcal{E}(x, r)$, which are processed by the anomaly decoder to extract the anomaly map, $a = \mathcal{D}_{\mathcal{A}}(f)$, and by the confidence decoder to extract the confidence map, $c = \mathcal{D}_{\mathcal{C}}(f)$. The information gathered in the anomaly map is summarized in a compact descriptor, $h = \mathcal{P}(a, c)$, by means of a weighted pooling block, with weights depending on the confidence information. Finally, this descriptor is processed by a classifier which computes an integrity score, $y = \mathcal{C}(h)$.

Integrity score, anomaly map and confidence map are all provided to the final user for further analyses. At a first level, only the integrity score is necessary to perform automated forgery detection. In case a fake is detected, the user can dive deeper using the anomaly map to identify manipulated suspected regions, along with the confidence map to distinguish valid predictions of forged regions from random anomalies. For pristine images, instead, the anomaly map does not localize possible forgeries but only random statistical anomalies, and should be discarded.

3.1. Noiseprint++

Motivation. Digital images are marked by a long trail of subtle, invisible traces. These may have many distinct origins, from the unavoidable imperfections of the camera hardware, to the in-camera processing steps of image acquisition, to all the out-camera processes encountered by the image during its lifetime. When images are manipulated, these telltale traces may be corrupted, an event that, if detected, allows one to carry out powerful forensic analyses.

In [12] a deep learning-based method has been proposed to extract from each image its noiseprint, an image-size pattern where all traces related to in-camera processing steps are collected and emphasized. This is trained in a self-supervised manner using only pristine images. While this ensures it can be trained on a large corpus, it shows limited robustness to image impairments induced by out-camera processes. This is a significant shortcoming, considering

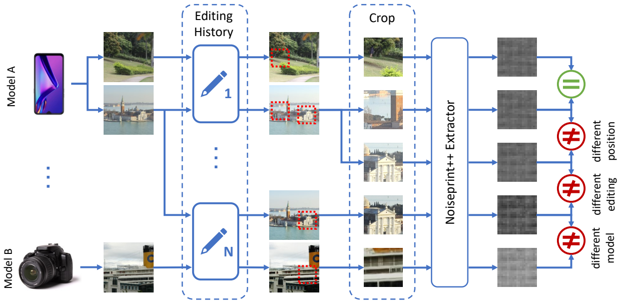

Figure 3. Noiseprint++ training procedure. Each image is subjected to a different combination of processing operations, namely *editing history*. Different crops are extracted from real images taken from many different cameras. During training, the distance between the outputs is minimized for patches coming from the same camera model, same position and same editing history.

that many forms of impairments are possible during the lifetime of an image. To overcome this limitation, we propose Noiseprint++, an improved image fingerprint which highlights traces related not only to in-camera but also to out-camera processes. In other words, Noiseprint++ captures information not only on the camera model but also on its editing history, improving its reliability.

Self-supervised contrastive learning. The proposed Noiseprint++ extractor learns patch-level self-similarities by means of contrastive learning. Similar to [12], we adopt the DnCNN architecture [45] with 15 trainable layers, 3 input channels, 1 output channel. The extractor is trained on patches of 64×64 pixels randomly extracted from images of the dataset. Training is aimed at obtaining the same noise-sensitive fingerprint for patches that share the same properties and different noise residuals for patches that are different under some respect. Figure 3, in particular, highlights that two patches are considered different, and hence characterized by different noise residuals, when they *(i)* come from different sources; *(ii)* are drawn from different spatial positions; *(iii)* have different editing histories. These constraints, in turn, aim at telling apart patches *(i)* generated by different cameras, *(ii)* moved from one spatial location to another and *(iii)* coming from images that have been differently post-processed. This latter property, in particular, distinguishes Noiseprint++ from its ancestor and improves its effectiveness. We adopt the InfoNCE contrastive loss [23]:

$$\mathcal{L}_{contr} = -\sum_{i\in\mathcal{B}} \log \frac{\sum_{j\in\mathcal{N}_i} e^{-s(i,j)}}{\sum_{j\in\mathcal{B}-\{i\}} e^{-s(i,j)}} \tag{1}$$

where $\mathcal{B}$ is a batch of patches, $s(i,j)$ is the squared Euclidean distance between i-th and j-th residual patches, and $\mathcal{N}_i$ is the subset of patches with the same origin, position and editing history as the i-th patch. During contrastive learning, we introduce a large variety of possible editing operations, such as resizing, compression and illumination changes, for a total of 512 different history pipelines.

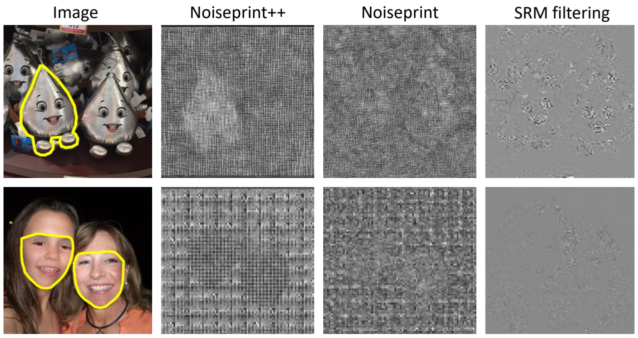

Figure 4. From left to right: manipulated image, Noiseprint++, Noiseprint and residual obtained through SRM-based filtering. We can notice that forensic artifacts are much more enhanced using our learned noise-sensitive fingerprint. In particular, we can observe the typical 8×8 grid that characterizes JPEG compressed images and Noiseprint++ can highlight the grid inconsistencies over the forged area better than Noiseprint.

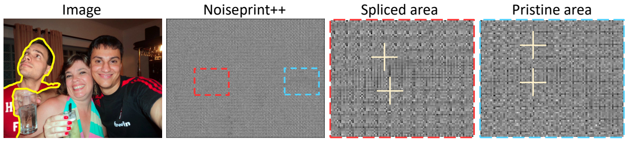

Figure 5. Example of a manipulated image and its Noiseprint++, which clearly shows a JPEG grid misalignment caused by the composition. This is visible in the spliced area (red), but not in the pristine one (blue), as highlighted in the zoomed regions where $+$ indicates a JPEG grid boundary.

In Fig. 4 we show two examples of Noiseprint++ compared to noiseprint and some standard spatial domain residuals (SRM filters), while in Fig. 5 we show a manipulated image where we can notice a JPEG grid misalignment in correspondence to the forged area.

3.2. Anomaly localization map

We treat the forgery localization task as a supervised binary segmentation problem and combine the Noiseprint++ information with the high-level features from the RGB image. To this end, we adopt the CMX architecture [28], a cross-modal fusion framework originally designed for multi-modal semantic segmentation, but easily generalizable to other tasks. Features from input image and Noiseprint++ are extracted on two parallel branches which have a shared encoder architecture from a semantic segmentation method. In particular, we rely on SegFormer [43], a hierarchical network based on a Transformer encoder. Interaction is carried out between each stage using a Cross-Modal Feature Rectification Module, which calibrates the information coming from one modality using features extracted from the other modality. The calibration helps to filter out noisy information of a modality using the knowledge of the other modality. The rectified features of both modalities are provided as input to the Feature Fusion Module, which uses a cross-attention mechanism to merge them into a single feature map. The fused feature maps of all stages represent the input of the decoder, which is used to generate the final anomaly map. For the decoder, we keep the lightweight multilayer perceptron used in SegFormer [43]. Details are provided in the Supplementary.

During phase 2 training, the loss function is a combination of the weighted cross-entropy and the dice loss [32]:

$$\mathcal{L}_2 = \lambda_{ce}\mathcal{L}_{ce} + (1-\lambda_{ce})\mathcal{L}_{dice} \tag{2}$$

with λ_{ce} set experimentally to 0.3. The weighted cross-entropy loss is defined as

$$\mathcal{L}_{ce} = -\frac{1}{N}\sum_i \gamma_0(1-g_i)\log(1-a_i) + \gamma_1 g_i \log a_i \tag{3}$$

with g_i and a_i the i-th pixel of the ground truth and estimated anomaly maps respectively, and N the number of pixels in the image. The weights γ_0 and γ_1, are set to 0.5 and 2.5 to take into account the imbalance between pristine and fake pixels in the training-set.

3.3. Confidence map and integrity score

Many SoTA methods perform localization first, and then use some global statistics of the localization map to perform detection. We also need global statistics about anomalies, but the anomaly map cannot be blindly trusted, as it highlights both manipulated areas and pristine areas with unusual statistics. Hence we propose a method to compute a per-pixel confidence estimate of the predicted anomaly map, which is used to compute robust global statistics for detection. In the pooling block we compute four *weighted* statistics of the anomaly map, maximum, minimum, average, and mean square, where the weights are drawn from the confidence map and help de-emphasize pristine anomalous areas of the image. In formulas

$$a_{\text{avg}} = \sum_i \acute{c}_i\, a_i; \qquad a_{\text{max}} = \log \textstyle\sum_i \acute{c}_i\, e^{a_i} \tag{4}$$

$$a_{\text{msq}} = \sum_i \acute{c}_i\, a_i^2; \qquad a_{\text{min}} = -\log \textstyle\sum_i \acute{c}_i\, e^{-a_i} \tag{5}$$

where a_i and $\acute{c}_i$ are the values of the anomaly and confidence maps at pixel i, respectively, the latter normalized to unit sum, and we adopt a smooth approximation of the minimum and maximum functions. To these features we add the four corresponding features extracted from the confidence map $c_{\text{avg}}, c_{\text{msq}}, c_{\text{max}}, c_{\text{min}}$ obtaining eventually a 8-component feature vector, h, which is used to predict the integrity score y.

The confidence and anomaly maps are generated in parallel, by decoding the same input features with two decoders having the same architecture, as done in [9]. However, while the anomaly values point out statistical outliers, confidence values have to recognize which anomaly values can be trusted. Hence, the confidence decoder must be trained with suitable *ad hoc* reference data. To this end, we use another map, t, the true class probability map [9]:

$$t_i = (1-g_i)\,(1-a_i) + g_i\, a_i \tag{6}$$

where g_i and a_i are the pixel values of the localization ground truth and of the anomaly map. The ground truth values, g_i, are 1 for manipulated pixels and 0 for pristine ones. Therefore, the true class probability map is close to 1 when large anomaly values occur for manipulated pixels or small anomaly values occur for pristine pixels. Instead, it is close to 0 when manipulated pixels are not seen as anomalous or anomalies are detected in pristine data. This latter case is especially important as it may easily lead to false alarms. The confidence decoder must learn to identify and discard these wrong pieces of information. Hence, the confidence loss, $\mathcal{L}_{conf}$, is defined as the mean squared error between the predicted confidence map c and its reference t.

Finally, to maximize the system reliability, the confidence decoder is trained jointly with the final binary classifier. Therefore we train this phase using a weighted sum of confidence loss and detection loss

$$\mathcal{L}_3 = \mathcal{L}_{conf} + \lambda_{det}\mathcal{L}_{det} \tag{7}$$

where $\mathcal{L}_{det}$ is the balanced cross-entropy on the predicted image-level integrity score y and λ_{det} is set to 0.5.

4. Results

4.1. Experimental Setup

Training. Our approach includes three separate training steps. First, we train the Noiseprint++ extractor using a large dataset of pristine images publicly available on two popular photo-sharing websites: Flickr (www.flickr.com) and DPReview (www.dpreview.com). The whole dataset contains 24,757 images acquired from 1,475 different camera models (8 to 92 images per model) of 43 brands. Then, we train encoder and decoder of the anomaly localization network using the same datasets as proposed in CAT-Net v2 [24], comprising pristine and fake images with the corresponding ground truths. Finally, using this same dataset, we train the confidence map decoder and the forgery detector. More details on these datasets can be found in the supplementary.

Testing. We benchmarked our model on seven publicly available datasets and one more dataset of local manipulations created by us using diffusion models. More specifically, we use CASIA v1 [15], Coverage [40], Columbia

Method	CASIAv1+		Coverage		Columbia		NIST16		DSO-1		VIPP		OpenFor.		CocoGlide		AVG	
	best	fixed	best	fixed	best	fixed	best	fixed	best	fixed	best	fixed	best	fixed	best	fixed	best	fixed
ADQ [6]	.494	.302	.167	.165	.401	.401	.238	.146	.483	.421	.549	.457	.644	.414	.302	.300	.410	.326
Splicebuster [10]	.252	.143	.321	.192	.811	.565	.312	.174	.662	.372	.432	.260	.459	.340	.434	.332	.460	.297
EXIF-SC [22]	.255	.106	.332	.164	.880	.798	.298	.227	.577	.442	.424	.215	.318	.175	.424	.293	.437	.303
CR-CNN [44]	.538	.481	.487	.391	.779	.631	.363	.300	.377	.289	.355	.282	.143	.110	.577	.447	.452	.366
RRU-Net [5]	.498	.408	.339	.279	.629	.575	.218	.154	.360	.312	.336	.272	.206	.157	.504	.416	.386	.322
ManTraNet [42]	.320	.180	.486	.317	.650	.508	.225	.172	.537	.412	.373	.255	.661	.551	.673	.516	.491	.364
SPAN [21]	.169	.112	.428	.235	.873	.759	.363	.228	.390	.233	.375	.223	.176	.089	.350	.298	.391	.272
AdaCFA [2]	.158	.128	.215	.183	.587	.403	.124	.106	.262	.235	.210	.184	.115	.098	.357	.314	.254	.206
CAT-Net v2 [24]	**.852**	**.752**	.582	.381	**.923**	**.859**	.417	.308	.673	.584	.672	.590	**.947**	**.899**	.603	.434	.709	.601
IF-OSN [41]	.676	.553	.472	.304	.836	.753	.449	.330	.621	.470	.508	.403	.204	.123	.589	.428	.544	.421
MVSS-Net [8]	.650	.528	.659	.514	.781	.729	.372	.320	.459	.358	.485	.389	.225	.117	.642	.486	.534	.430
PSCC-Net [29]	.670	.520	.615	.473	.760	.604	.210	.113	.733	.458	.309	.183	.353	.105	.685	.515	.542	.371
Noiseprint [12]	.205	.137	.342	.229	.835	.513	.345	.196	.811	.439	.546	.382	.675	.420	.405	.318	.521	.329
TruFor (ours)	.822	.737	**.735**	**.600**	.914	**.859**	**.470**	**.399**	**.973**	**.930**	**.746**	**.693**	.901	.827	**.720**	**.523**	**.785**	**.696**

Table 1. Pixel-level F1 performance of image forgery localization. Results are shown for the metric computed using the best threshold per image and using a fixed threshold (0.5). First and second rankings are shown in bold and underlined respectively. For the fixed threshold, Splicebuster and Noiseprint have been evaluated after a Normalization between 0 and 1, since they provide maps in arbitrary ranges.

[20], NIST16 [19], DSO-1 [13], and VIPP [7], which are extensively used in the literature and include cheapfakes manipulations, like splicing, copy-move and inpainting. Overall these datasets comprise a total of 1530 fake images and 1412 real ones. Then, we added OpenForensics [26] a large dataset of face manipulations generated using GAN models, from which we sampled 2000 images, and CocoGlide, including 512 images we generated from the COCO 2017 validation set [27] using the GLIDE diffusion model [33].

Metrics. As in most of the previous works, we measure pixel-level performance in terms of F1, and report results using both the best threshold and the default 0.5 threshold. Instead, for image-level analysis we use AUC, which does not require setting a decision threshold, and balanced accuracy, which takes into account both false alarms and missed detection, in which case the threshold is set again to 0.5.

4.2. State-of-the-art comparison

To ensure a fair comparison we considered only methods with code and/or pre-trained models publicly available on-line and run them on the selected testing datasets. Moreover, to avoid biases, we included only the approaches trained on datasets disjoint from the test datasets. Eventually, we included two model-based methods: ADQ [6] that relies on JPEG artifacts, Splicebuster [10] that exploits noise artifacts; and 11 deep learning-based methods: EXIF SelfConsistency [22], Constrained R-CNN [44], RRU-Net [5], ManTraNet [42], SPAN [21], AdaCFA [2], E2E [31], CAT-Net v2 [24], IF-OSN [41], MVSS [8], PSCC-Net [29], Noiseprint [12]. A brief summary of these methods is provided in Tab. 3.

Localization results. In Tab. 1 we show the pixel-level localization performance. Our method provides the best F1 performance, on average, and is the best or second best on all datasets, which testifies of a remarkable generalization ability across manipulations. In fact, it performs well also on OpenForensics (GAN-based local manipulations), where most other methods fail catastrophically, except CAT-Net v2, as well as CocoGlide (diffusion-based local manipulations). Thanks to the use of Noiseprint++, with its digital history-based training, our method keeps working well on all the datasets.

Detection results. Detection results are shown in Tab. 2. Note that we also consider methods that were not explicitly designed for this task, in which case we use the maximum of the localization map as the detection statistic, as it works better than the mean value. TruFor is the best performer on most datasets, and has the best average performance both in terms of AUC and Accuracy. On the contrary, many methods exhibit a very poor performance, close to random guessing (0.5). This phenomenon is especially acute for accuracy, which is highly sensitive to the choice of threshold (see supplementary). Indeed, lacking a suitable calibration dataset, setting the right threshold is a difficult problem, as also shown in [14]. Unlike most competitors, our approach guarantees an accuracy of almost 80% even in this challenging case.

Robustness analysis. In this section we carry out a robustness analysis on images impaired by compression and resizing. To this end, we use three datasets uploaded on Facebook and Whatsapp - two provided in [41] and our CocoGlide. For compactness, in Tab. 4 we compare results only with the top three competitors according to the F1 performance (fixed threshold) of Tab. 1: IF-OSN, CAT-Net v2

Method	CASIAv1+		Coverage		Columbia		NIST16		DSO-1		VIPP		CocoGlide		AVG	
	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc
ADQ [6]	.816	.523	.495	.495	.500	.500	.484	.503	.569	.560	.736	.551	.496	.496	.585	.518
Splicebuster [10]	.406	-	.541	-	.597	-	.610	-	.751	-	.539	-	.529	-	.568	-
EXIF-SC [22]	.490	.500	.498	.500	.976	.506	.504	.500	.764	.500	.617	.500	.526	.500	.625	.501
CR-CNN [44]	.670	.535	.553	.510	.755	.628	.737	.641	.576	.535	.504	.558	.589	.533	.626	.563
RRU-Net [5]	.574	.488	.482	.500	.583	.500	.666	.500	.444	.500	.534	.500	.533	.503	.545	.499
ManTraNet [42]	.644	.500	.760	.500	.810	.500	.624	.500	.874	.500	.530	.500	**.778**	.500	.717	.500
SPAN [21]	.480	.487	.670	.605	**.999**	.951	.632	.597	.669	.510	.580	.572	.475	.491	.644	.602
AdaCFA [2]	.500	.500	.500	.500	.500	.500	.500	.500	.500	.500	.500	.500	.500	.500	.500	.500
CAT-Net v2 [24]	**.942**	**.838**	.680	.635	.977	.803	.750	.597	.747	.525	.813	.565	.667	.580	.797	.649
IF-OSN [41]	.735	.635	.557	.510	.882	.522	.658	.553	.853	.505	.696	.522	.611	.567	.713	.545
MVSS-Net [8]	.932	.808	.733	.545	.984	.667	.579	.538	.552	.485	.629	.522	.654	.536	.723	.586
PSCC-Net [29]	.869	.683	.657	.550	.300	.508	.485	.456	.650	.543	.574	.507	.777	**.661**	.616	.558
E2E [31]	.377	.433	.494	.505	.894	.639	.718	.603	.803	.565	.617	.543	.530	.525	.633	.545
Noiseprint [12]	.494	-	.525	-	.872	-	.618	-	.821	-	.580	-	.520	-	.633	-
TruFor (ours)	.916	.813	**.770**	**.680**	.996	**.984**	**.760**	**.662**	**.984**	**.930**	**.820**	**.761**	.752	.639	**.857**	**.781**

Table 2. Image-level AUC and balanced Accuracy performance of image forgery detection. Splicebuster and Noiseprint cannot be evaluated using a fixed threshold because they provide maps in arbitrary ranges.

and MVSS-Net. TruFor performs consistently better than all competitors, even though IF-OSN was specifically proposed to deal with images transmitted via social networks, while the gap with respect to CAT-Net v2 and MVSS-Net widens significantly.

Qualitative comparisons. In Fig. 6 we also show some visual results in order to gain a better insight into the quality of the image localization maps and corresponding confidence maps. Together with some fakes, we show some real images, for which the localization map can be erroneous. In these cases we show the anomaly map, which often presents some hot spots that could lead to false positives. Such errors are avoided in detection thanks to the additional confidence map. The user may inspect all these pieces of information to carry out further analyses. More qualitative results are shown in Supplementary.

		Input Type		Task	
Acronym [ref]	Artifact	RGB	Other	L	D
ADQ [6]	JPEG	✓	DCT analysis	✓	
Splicebuster [10]	camera-based		fixed HP filter	✓	
EXIF-SC [22]	camera-based	✓	-	✓	✓
AdaCFA [2]	demosaicing	✓	-	✓	
Noiseprint [12]	camera-based	✓	-	✓	
ManTraNet [42]	editing	✓	HP filters	✓	✓
RRU-Net [5]	splicing	✓	-	✓	✓
SPAN [21]	editing	✓	HP filters	✓	
CR-CNN [44]	editing		trainable HP filter	✓	
CAT-Net v2 [24]	JPEG	✓	DCT filter	✓	
MVSS-Net [8]	editing		trainable HP filter	✓	✓
IF-OSN [41]	editing	✓	-	✓	
PSCC-Net [29]	editing	✓	-	✓	✓
E2E [31]	editing	✓	noiseprint		✓

Table 3. Methods used for comparison. We indicate the artifacts they rely on and the input type, if they work only on RGB and/or on noise features extracted through high-pass (HP) filtering. In addition, we also indicate if they have been designed for localization (L), detection (D) or both.

Method	CASIAv1		DSO-1		CocoGlide	
	Fb	Wa	Fb	Wa	Fb	Wa
IF-OSN [41]	.513	.524	.484	.395	.406	.404
CAT-Net v2 [24]	.681	.508	.310	.247	.447	.443
MVSS-Net [8]	.469	.444	.356	.308	.347	.351
TruFor (ours)	**.716**	**.713**	**.685**	**.465**	**.460**	**.461**

Table 4. Pixel-level F1 performance (using fixed threshold) on datasets uploaded on Facebook (Fb) and WhatsApp (Wa).

	Original		Res		Res&Cmp	
Version	F1	AUC	F1	AUC	F1	AUC
Noiseprint	.706	.547	.375	.483	.342	.468
Noiseprint++	.877	.913	.666	.745	.435	.566
SegFormer (NP++)	.974	.967	.925	**.966**	.649	.703
SegFormer (RGB)	.917	.903	.780	.792	.756	**.786**
TruFor (NP++, RGB)	**.982**	**.974**	**.937**	.944	**.765**	.730

Table 5. Ablation results. Pixel-level F1 performance (using best threshold) and image-level AUC on original images, resized (Res) and resized and recompressed (Res&Cmp).

4.3. Ablation study

In order to assess the individual impact of all design choices of our approach, we consider a simple baseline, the noiseprint-based method proposed in [12], and add the new key components one at a time. Experiments are carried out on a dataset of 1000 manipulated images built by downloading pristine images from the web and editing them locally, so as to simulate a realistic scenario. Tab. 5 shows the results (F1 and AUC) for the noiseprint baseline, the version with Noiseprint++, the method which includes transformer-based segmentation (using as inputs only RGB and only NP++), and the proposed method with joint analysis of Noiseprint++ and RGB image. We also perform this analysis after resizing all images and after resizing and compressing them. The case with strongly impaired images is more challenging, but the proposed method keeps providing a good performance. In general, the inclusion of high-level segmentation information seems to provide the largest improvement, justifying our focus on all-round clues.

With Tab. 6 we study the effect of using the cross-entropy loss alone or jointly with the dice loss, with and without online augmentation with compressed and resized images. On the original data, results (F1 with best threshold) remain pretty stable in all cases. With resized and compressed data, instead, the joint use of cross-entropy and dice loss proves important, especially together with augmentation.

Finally, in Tab. 7 we consider image-level detection and compare our method with two simplified versions that rely on a single global feature, the mean or the maximum of the anomaly map. First of all, it is clear that the mean is a poor decision statistic, and much better AUC results can be obtained by just switching to the maximum. However, even the maximum turns out to be almost useless without a calibration process that helps select a good decision threshold. So, in terms of accuracy, the feature vector used in the proposed method provides a large competitive advantage.

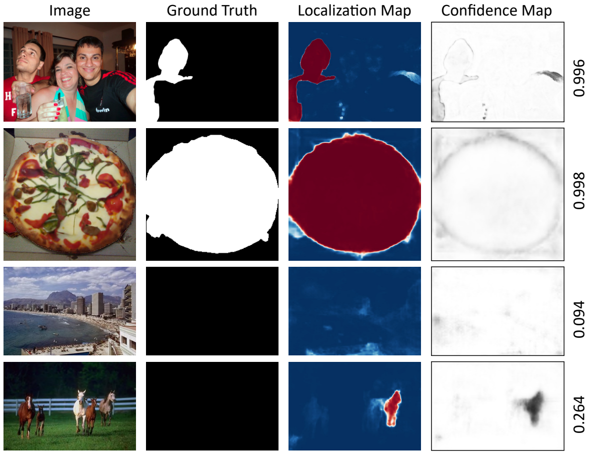

Figure 6. Fake images (top) and pristine images (bottom). In the last row, we show the confidence map that is able to correct the error in the localization map, hence improving the global integrity score (shown on the right).

		Original		**Res**		**Res&Cmp**	
Loss	**Aug**	best	fixed	best	fixed	best	fixed
CE		.980	.926	.912	.824	.583	.397
CE	✓	.981	.929	.885	.781	.575	.575
CE+DL		.973	.949	.865	.767	.655	**.655**
CE+DL	✓	**.982**	**.970**	**.937**	**.902**	**.765**	.627

Table 6. Localization ablation results: Pixel-level F1 performance (using best and fixed threshold).

	Original		**Res**		**Res&Cmp**	
score	AUC	Acc	AUC	Acc	AUC	Acc
mean	.544	.510	.604	.525	.592	.515
max	.974	.505	.944	.515	.730	.500
Ours	**.996**	**.905**	**.949**	**.910**	**.740**	**.675**

Table 7. Detection ablation results: Image-level AUC and Accuracy (best threshold).

5. Conclusions

In this paper we introduce TruFor, a novel framework for reliable image forgery detection and localization. It is built upon the extraction of a learned noise-sensitive fingerprint, that enhances the in-camera and out-camera artifacts even in challenging scenarios, such as circulation on social networks. The model also provides a confidence map that represents an indication of possible false alarms on pristine areas. Our extensive experimental results demonstrate that our approach has a good generalizability and is able to localize even unknown manipulations, such as the recent DNN-based ones. Furthermore, it can provide reliable and robust detection results at image level thanks to the introduction of the confidence map. Our approach has certain limitations. First, it cannot detect fully generated images. Then, we train the anomaly map and detection score in separate phases, requiring full pixel-level supervision. In future work, we would like to explore end-to-end training, allowing partial supervision from only image-level labels. We would also like to evaluate generalization on more recent generative models for local edits [1, 17].

Acknowledgment. We gratefully acknowledge the support of this research by the Defense Advanced Research Projects Agency (DARPA) under agreement number FA8750-20-2-1004. In addition, this work has received funding by the European Union under the Horizon Europe vera.ai project, Grant Agreement number 101070093. It is also supported by a TUM-IAS Hans Fischer Senior Fellowship and by the PREMIER project, funded by the Italian Ministry of Education, University, and Research within the PRIN 2017 program. Finally, we would like to thank Chris Bregler for useful discussions and support.

References

[1] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *arXiv preprint arXiv:2206.02779*, 2022. 1, 8

[2] Quentin Bammey, Rafael Grompone von Gioi, and Jean-Michel Morel. An adaptive neural network for unsupervised mosaic consistency analysis in image forensics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 6, 7

[3] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *ACM Workshop on Information Hiding and Multimedia Security*, pages 5–10, 2016. 2

[4] Christian Riess Benedikt Lorch, Anatol Maier. Reliable JPEG Forensics via Model Uncertainty. In *IEEE Workshop on Information Forensics and Security (WIFS)*, 2020. 3

[5] Xiuli Bi, Yang Wei, Bin Xiao, and Weisheng Li. RRU-Net: The Ringed Residual U-Net for Image Splicing Forgery Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 2, 6, 7

[6] Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Improved DCT coefficient analysis for forgery localization in JPEG images. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2444–2447, 2011. 2, 6, 7

[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. 6

[8] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 2, 6, 7

[9] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2902–2913, 2019. 3, 5

[10] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Splicebuster: A new blind image splicing detector. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2015. 6, 7

[11] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. *Data-Driven Digital Integrity Verification*, pages 281–311. Springer, 2022. 2

[12] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A CNN-Based Camera Model Fingerprint. *IEEE Transactions on Information Forensics and Security*, 15:144–159, 2020. 2, 3, 4, 6, 7, 8

[13] 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. 6

[14] Chengbo Dong, Xinru Chen, Ruohan Hu, Juan Cao, and Xirong Li. MVSS-Net: Multi-View Multi-Scale Supervised Networks for Image Manipulation Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3539–3553, 2023. 6

[15] Jing Dong, Wei Wang, and Tieniu Tan. CASIA image tampering detection evaluation database. In *IEEE China Summit and International Conference on Signal and Information Processing*, pages 422–426, 2013. 5

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

[17] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 8

[18] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342v3*, 2021. 3

[19] H. Guan, M. Kozak, E. Robertson, Y. Lee, A.N. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus. MFC datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *IEEE WACV Workshops*, pages 63–72, 2019. 6

[20] Yu-feng Hsu and Shih-Fu Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 549–552, 2006. 6

[21] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. SPAN: Spatial pyramid attention network for image manipulation localization. In *European Conference on Computer Vision (ECCV)*, pages 312–328. Springer, 2020. 2, 6, 7

[22] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *European Conference on Computer Vision (ECCV)*, September 2018. 2, 6, 7

[23] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:18661–18673, 2020. 4

[24] 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. *International Journal of Computer Vision*, pages 1–21, 2022. 2, 5, 6, 7

[25] 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 *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 375–384, 2021. 2

[26] Trung-Nghia Le, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. OpenForensics: Large-Scale Challenging Dataset for Multi-Face Forgery Detection and Segmentation In-the-Wild. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10117–10127, October 2021. 1, 6

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 6

[28] Huayao Liu, Jiaming Zhang, Kailun Yang, Xinxin Hu, and Rainer Stiefelhagen. CMX: Cross-Modal Fusion for RGB-X Semantic Segmentation with Transformers. *arXiv preprint arXiv:2203.04838*, 2022. 2, 4

[29] Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu. PSCC-Net: Progressive spatio-channel correlation network for image manipulation detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):7505–7517, november 2022. 2, 6, 7

[30] Anatol Maier, Benedikt Lorch, and Christian Riess. Toward reliable models for authenticating multimedia content: detecting resampling artifacts with Bayesian Neural Networks. In *IEEE International Conference on Image Processing (ICIP)*, pages 1251–1255, 2020. 3

[31] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. A Full-Image Full-Resolution End-to-End-Trainable CNN Framework for Image Forgery Detection. *IEEE Access*, 8:133488–133502, 2020. 2, 6, 7

[32] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *Fourth International Conference on 3D Vision (3DV)*, pages 565–571, 2016. 5

[33] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, volume 162, pages 16784–16804, Jul 2022. 1, 6

[34] Jinseok Park, Donghyeon Cho, Wonhyuk Ahn, and Heung-Kyu Lee. Double JPEG Detection in Mixed JPEG Quality Factors using Deep Convolutional Neural Network. In *European Conference on Computer Vision (ECCV)*, September 2018. 2

[35] Yuan Rao and Jiangqun Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2016. 2

[36] Yuan Rao, Jiangqun Ni, and Hao Xie. Multi-semantic CRF-based attention model for image forgery detection and localization. *Signal Processing*, 183:108051, 2021. 2

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

[38] Luisa Verdoliva. Media forensics and deepfakes: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020. 2, 3

[39] Junke Wang, Zuxuan Wu, Jingjing Chen, Xintong Han, Abhinav Shrivastava, Ser-Nam Lim, and Yu-Gang Jiang. ObjectFormer for image manipulation detection and localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 2

[40] Bihan Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. COVERAGE — A novel database for copy-move forgery detection. In *IEEE International Conference on Image Processing (ICIP)*, pages 161–165, 2016. 5

[41] Haiwei Wu, Jiantao Zhou, Jinyu Tian, and Jun Liu. Robust image forgery detection over online social network shared images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 6, 7

[42] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries With Anomalous Features. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 6, 7

[43] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:12077–12090, 2021. 4, 5

[44] Chao Yang, Huizhou Li, Fangting Lin, Bin Jiang, and Hao Zhao. Constrained R-CNN: A General Image Manipulation Detection Model. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2020. 2, 6, 7

[45] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 4

[46] Rongyu Zhang and Jiangqun Ni. A dense U-Net with cross-layer intersection for detection and localization of image forgery. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2982–2986, 2020. 2

[47] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning rich features for image manipulation detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2