https://doi.org/10.2352/EI.2022.34.4.MWSF-331

NoiseSeg: An Image Splicing Localization Fusion CNN With Noise Extraction And Error Level Analysis Branches

Karol Gotkowski; Deutsches Krebsforschungszentrum; Heidelberg, Germany
Huajian Liu; Fraunhofer SIT / ATHENE; Darmstadt, Germany
Martin Steinebach; Fraunhofer SIT / ATHENE; Darmstadt, Germany

Abstract

To counter the ever increasing flood of image forgeries in the form of spliced images in social media and the web in general, we propose the novel image splicing localization CNN NoiseSeg. NoiseSeg fuses statistical and CNN-based splicing localization methods in separate branches to leverage the benefits of both. Unique splicing anomalies that can be identified by its coarse noise separation branch, fine-grained noise feature branch and error level analysis branch all get combined in a segmentation fusion head to predict a precise localization of the spliced regions. Experiments on the DSO-1, CASIAv2, DEFACTO, IMD2020 and WildWeb image splicing datasets show that NoiseSeg outperforms most other state-of-the-art methods significantly and even up to a margin of 46.8%.

Introduction

Image forgeries that copy regions from one image into another, while appearing to be authentic, have become increasingly a problem. These types of image forgeries are known as image splicing and are most notably used in fake news to increase the perceived authenticity. Spliced images are often indistinguishable for the bare eye from genuine images or videos with today's photo editing software, increasing the necessity for image splicing detection & localization algorithms.

Existing statistical methods relying on camera filter array abnormalities, noise artifacts and others statistics may work well for specific types of manipulations but lack an overall reliability due to the vast landscape of possible splicing manipulations. Convolutional Neural Networks (CNN) have emerged as an alternative as they can detect & localize image splicing better overall. However, they too fail on more complex manipulations, making the problem of image splicing anything but solved. To this day, digital forensics lacks the ability to reliably detect and localize spliced images.

In this paper propose the novel CNN-based image splicing localization approach NoiseSeg. NoiseSeg combines statistical approaches and CNN approaches as a set of branches and leverages the information gained from all of them in a segmentation head network. In total NoiseSeg employs a noise separation branch to coarsely separate pristine regions from spliced ones, a fine-grained noise feature branch to precisely localize the borders between pristine and spliced regions and an error level analysis branch to detect compression artefacts. The features of all branches are fused together in a segmentation head, which leverages the information of all branches to create a precise localization of the spliced regions. Example results from NoiseSeg are shown in Figure 1.

We evaluate NoiseSeg on multiple datasets against multiple state-of-the-art statistical and CNN-based methods of image splicing localization. The results demonstrate that NoiseSeg outperforms most of them significantly.

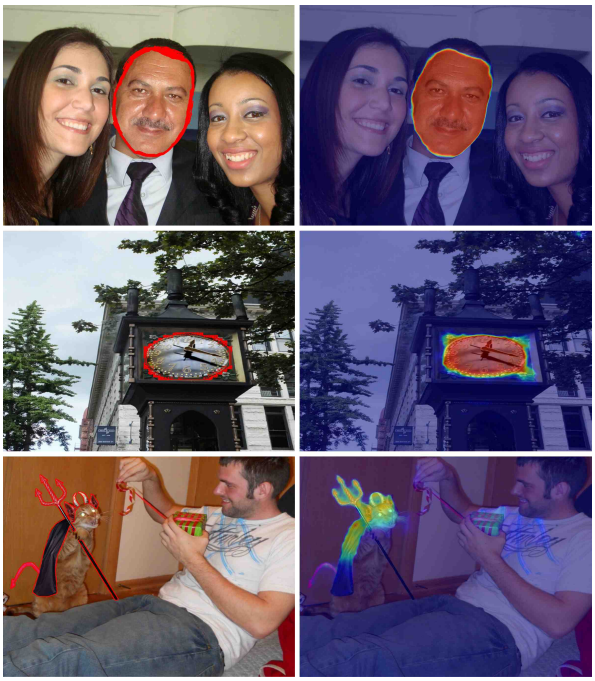

Figure 1. *Example predictions of NoiseSeg on the DSO-1 (top), DEFACTO (middle) and IMD2020 (bottom) datasets. Left side shows the spliced image with the spliced region being outlined by a red contour and the right side depicts the NoiseSeg prediction as a heatmap.*

Related Work

Methods for the detection and localization of spliced image regions can be separated into statistical methods and convolutional neural network (CNN) based methods. Popular statistical methods analyze the color filter array for abnormalities (CFA) [1] [2], noise artifacts (ELA) [3] or the discrete cosine transform (DCT) [4]. Further methods include BLK [5], CAGI [6], NOI [7] [8] and Splicebuster [9], which make use of other statistical features. These methods usually only detect one specific type of manipulation using handcrafted features.

Therefore, in recent years the focus shifted more towards CNNs as they seem to outclass statistical methods in many ways. One popular CNN method from recent years is ManTra-Net [10], which

employs a manipulation-trace feature extractor network and a local anomaly detection network for the detection and localization of 385 different types of image manipulations. Both parts of the networks are trained separately on carefully crafted synthetic datasets. Even though ManTra-Net outclasses most statistical methods, it still performs only mediocre on current image splicing datasets.

Another CNN method named Noisprint [11] enhances the different noise patterns in a spliced image such that they are often visible with the bare eye. The different noise patterns are subsequently segmented via expectation-maximization (EM). The architecture consists of a DnCNN noise extraction network, which is trained in siamese fashion with patches from different pristine images in order to learn to enhance the noise patterns. Like ManTra-Net Noiseprint outperforms most statistical methods, but suffers from mediocre performance on current datasets.

Methodology

In this section, we first give an overview of our model architecture and then give a more detailed explanation of our design decisions.

Model Architecture

Statistical methods can often detect artefacts or statistical anomalies, which can currently not be reproduces by CNNs. However, these statistical methods mostly only detect one type of manipulation. On the other hand, CNNs can learn to detect image anomalies that cannot be found with handcrafted statistical methods. We propose the model NoiseSeg that combines statistical approaches and CNN approaches as a set of branches and leverages the information gained from all of them in a segmentation head network. An overview of our model architecture is depicted in Figure 2. NoiseSeg has in total three branches. First, a Noiseprint branch that employs the Noiseprint [11] model to coarsely separate pristine regions from spliced ones. Second, an image denoising branch that uses the DnCNN [12] model to extract raw and fine-grained noise features. Third, an error level analysis (ELA) [3] branch to detect compression artefacts, which cannot be detected by the other two branches. The information of all the branches is fused together in the segmentation head by concatenating the resulting feature maps of each branch along their channel dimension and forwarding them to the segmentation head. This head is a Feature Pyramid Network (FPN) [13], which learns during training to utilize the information of every branch and therefore can predict fine-grained segmentation masks.

Noise feature learning

Many approaches use high-pass filtering to extract noise features and then perform some statistical measurements to detect splicing. However, by now, better alternatives exist in the form of denoising CNNs. They were first used as an alternative to high-pass filtering in Noiseprint. Noiseprint works well at separating different noise sources, but uses Expectation Maximization (EM) to detect image splicing based on image patches. This limits the performance of the approach and further results in bad localization at the borders of spliced regions.

To solve these problems, we use the pretrained Noiseprint model as a branch for NoiseSeg and replace the Expectation Maximization head with the FPN head to remove the need for patch-based classification and to make the approach more flexible. This further enables us to fuse information within the FPN by adding more branches. To mitigate Noiseprint's problem of bad localization at the borders of spliced regions, we further add a DnCNN model as a separate branch of NoiseSeg to extract raw noise features. By fusing the Noiseprint features and the DnCNN noise features within the FPN head during training, we are able to drastically increase the precision in the border regions by utilizing the coarse class separation of Noiseprint together with the fine-grained noise information at the borders.

Error level analysis

The error level analysis (ELA) detects compression artefacts of images that have been compressed multiple times. These types of artefacts are crucial to analyze in order to detect traces of image splicing. However, denoising CNNs such as DnCNN are not trained to reproduce these types of artefacts in their noise features. The same is true for Noiseprint, which is based on DnCNN. Therefore, we add a third branch to NoiseSeg, which includes this error level analysis. This information is then also fused together with the information of the other branches in the FPN head.

Experimental Setup

In this section, we discuss our experimental setup, which includes the used datasets, the implementation details, the used baselines for comparison and the evaluation measures.

Datasets

In total, we choose five public datasets for training and evaluation and one synthetic dataset exclusively for training.

CASIA v2 [14] is a popular splicing dataset, consisting of images from various sources and depicting a wide range of real-world scenarios. As CASIA v2 has no ground truth masks, we use generated ones from [15] which approximate the ground truth quite well. However, due to some naming ambiguities, not all images could be used. In total, we used 3240 images of which 2592 images are used for training and 648 for testing.

DEFACTO [16] is a image forgery dataset consisting of seven different sets. Each set consists of multiple thousand spliced images with the same type of donor object. E.g. the first set of images has different kind of planes spliced into the images, while the second set has different kind of clocks spliced into the images. Due to the high correlation of same objects, we choose to only use the first set for training while using the second set for testing. The dataset was created synthetically from the MSCOCO [17] dataset with always the same type of splicing operations, thus the other five sets do not add value to the model performance and would only increase the training time significantly. Therefore, we ignore the other sets. In total, we use 10765 images from the first set for training and 2000 images from the second set for testing.

The **DSO-1** [18] dataset is an older popular dataset consisting of 100 spliced images. It focuses entirely on splicing different people into group photos. We use 80 images for training and 20 images for testing.

The **IMD2020** [19] dataset includes 2010 spliced images. This dataset is one of the most diverse, as the spliced images were collected from the internet and reflect the types of manipulations used in the wild. Of the 2010 images we use 1608 images for training and 402 for testing.

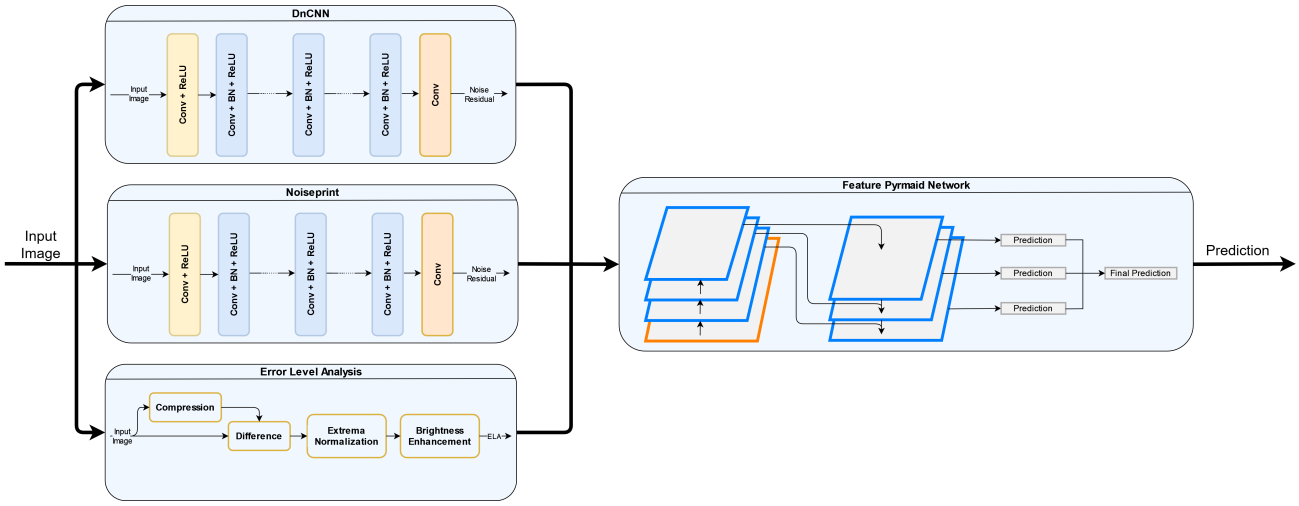

Figure 2. *Overview of the NoiseSeg model architecture. Each of the three branches provide unique information that is fused in the segmentation head. The three branches consist of a noise separation branch employing Noiseprint, a DnCNN denoising branch to extract raw fine-grained noise features and an error level analysis branch to detect compression artefacts. All features are fused in the FPN segmentation head.*

We also use the **WildWeb** [20] dataset, which is also the most difficult dataset. Like the IMD2020 dataset, it consists of spliced images and even its pristine probes collected from the internet. However, almost all the images are of very low resolution and compressed multiple times, making it the most difficult splicing dataset. Each pristine probe has multiple spliced variations with different donor images. In total, 9.665 spliced images are usable for us, of which we use 7538 for training and 2127 for testing.

The last dataset used by us is a synthetically created splicing dataset consisting of 10.000 images. Both the probe and donor images come from the VISION [21] dataset, while the donor image is a randomly chosen region in the form of a random b-spline contour with blurred edges. This dataset is exclusively used for training.

As most of the datasets have only a few thousand images, which is very small in terms of training data for CNNs, we combine the training sets of all datasets to a single one and use it for training our NoiseSeg model. This NoiseSeg model is then evaluated separately on the test set of each dataset.

Implementation Details

We do not resize or normalize the train images as resizing the images could lead to a loss of anomaly information and normalization is not necessary as DnCNN, Noiseprint and FPN employ batch normalization layers. All models are initialized with their pretrained weights, and the parameters of DnCNN and Noiseprint are fixed. The implementation of NoiseSeg is done in PyTorch [22] and the segmentation_models.pytorch [23] library is used for the FPN head. The Adam optimizer is used for training with a learning rate of 0.001, binary cross entropy loss, gradient clipping at 0.1, a batch size of 8 and a training time of 50 epochs. To make use of all the available training data, the NoiseSeg model is trained with 5-fold cross-validation.

Moreover, due to the extreme imbalance of the different datasets each dataset is repeated multiple times to match the size of the largest dataset, which improves the results significantly. Random flipping and rotation is used for data augmentation.

Finally, a technique called threshold-moving is employed after training. Threshold-moving calibrates the model based on the train set by optimizing the pixel-classification threshold with grid-search once after training. This improves the model's performance on new unseen images.

Baselines

We choose a number of different methods as baselines for a comprehensive evaluation of our proposed approach. For most statistical baselines, we use the MKLab-ITI framework [24], which provides the following baselines: ELA, BLK, CAGI, CFA1, CFA2, DCT, NOI1, NOI2, NOI4, NOI5. Another statistical baseline we compare against is Splicebuster. Further, we compare against the CNN-based baselines of ManTra-Net and Noiseprint. For fairness, we also employ threshold-moving based on all combined train sets, as we did for NoiseSeg to optimize the pixel-classification threshold and calibrate each model.

Evaluation Measures

There is a high class imbalance in terms of spliced and pristine regions. Therefore, we opt to use evaluation measures suited for problems with class imbalance. Further, we want to increase comparability with the evaluations of other works and therefore choose to report three popular measures. The first used measure is the Matthews Correlation Coefficient (MCC), which is very robust against class imbalance and therefore the most important measure. It is defined as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (1)$$

Here, TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives, respectively. Another popular measure is the F1 score, which is also known as the Dice score. It is defined as:

$$F1 = \frac{2TP}{2TP+FN+FP} \quad (2)$$

Our third measure is Average Precision (AP), which is defined as:

$$AP = \sum_n (R_n - R_{n-1}) \times P_n \tag{3}$$

$$P = \frac{TP}{TP+FP} \quad R = \frac{TP}{TP+FN} \tag{4}$$

A problem that needs to be addressed during the evaluation is that the splicing and pristine class are interchangeable. It is sometimes only possible to separate both classes in an image, but not to determine which is which, as there is no distinction between the two. Therefore, we always compute the measures on both the original and the inverted prediction and choose the better one during the evaluation.

Results

We first compared NoiseSeg against the baselines on each dataset. Followed by this, we performed an ablation study of the model branches used in NoiseSeg. This is then concluded with a second ablation study by comparing different segmentation heads.

Splicing localization

In this section we compared NoiseSeg against Noiseprint, ManTra-Net, Splicebuster and the other baselines. We report the results for the measures MCC, F1 and AP. However, as MCC is the most important measure, we focus our discussion on the MCC results and show the results for the F1 and AP measure in the appendix.

The MCC evaluation results are shown as table below, but also in Figure 3 and Figure 4. The results show that NoiseSeg performs the best over all datasets for all evaluation metrics. Especially on the DSO-1, IMD2020 and CASIAv2 datasets, NoiseSeg outperforms the second-best method by 46.8%, 26.3% and 41.3%, respectively. The CNN-based methods ManTra-Net and Noiseprint perform on average better than most statistical methods, but also perform overall significantly worse than NoiseSeg. The statistical methods nearly all achieve at maximum a MCC score of 0.2 with DCT and CAGI being the exception. In fact, DCT and CAGI are the only statistical methods that come close to matching the performance of ManTra-Net and Noiseprint. The only dataset on which all methods achieve only low MCC results is the WildWeb dataset. However, this was to be expected as most images in this dataset are of low resolution and are compressed multiple times.

	MCC				
	DSO-1	CASIAv2	DEFACTO	IMD2020	WildWeb
ELA	0.233	0.160	0.061	0.140	0.082
BLK	0.051	0.119	0.056	0.112	0.089
CAGI	0.362	0.108	0.082	0.195	0.109
CFA1	0.052	0.059	0.034	0.089	0.066
CFA2	0.063	0.054	0.033	0.070	0.068
DCT	0.169	0.425	0.116	0.232	0.104
NOI1	0.173	0.028	0.042	0.060	0.021
NOI2	0.021	0.047	0.071	0.071	0.099
NOI4	0.093	0.021	0.029	0.032	0.010
NOI5	0.171	0.038	0.055	0.062	0.043
Splicebuster	0.332	0.079	0.103	0.145	0.127
ManTraNet	0.336	0.171	0.388	0.205	0.026
Noiseprint	0.494	0.077	0.109	0.212	0.093
NoiseSeg	**0.962**	**0.584**	**0.404**	**0.475**	**0.126**

The MCC test set results of NoiseSeg and the baselines over all dataset.

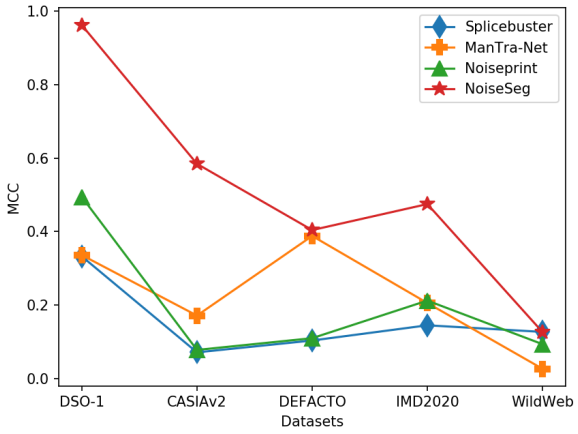

Figure 3. *Comparison of the MCC evaluation results of NoiseSeg and the baselines ManTra-Net, Noiseprint and Splicebuster over all datasets. NoiseSeg outperforms the other methods on multiple datasets significantly.*

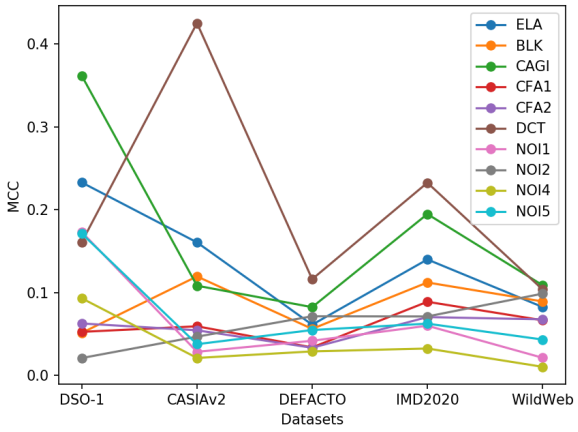

Figure 4. *Comparison of the MCC evaluation results of the statistical baselines over all datasets. Almost all baselines reach only MCC scores below 0.2. This shows the clear advantage of CNN-based methods over statistical ones.*

Qualitative comparison

In Figure 5 a qualitative comparison of the best performing methods ManTra-Net, Noiseprint and Noiseg is shown. ManTra-Net identifies on each image falsely many small regions as spliced. Even though, it is occasionally correct with its prediction, this tendency makes the method difficult to use in practice and unreliable. By contrast, Noiseprint detects often very large regions as spliced. This is a drawback of the used Expectation Maximization method, as it always tries to separate the noise into two classes. However, due to varying image quality also the pristine part of an image can have different noise pattern, which are often larger than the spliced region. This leads the EM algorithm to separate the pristine regions instead. Moreover, the results of Noiseprint are always coarse on lower resolution images, as the EM classifies patches and not pixels. By contrast, NoiseSeg pre-

Figure 5. *Examples from all datasets for NoiseSeg and the relevant baseline methods. From left to right: Spliced image, the corresponding ground truth mask, heatmaps of the predictions results overlayed on the spliced image for ManTra-Net, Noiseprint and NoiseSeg. On the heatmaps the color red indicates a high confidence that the region is spliced while blue indicates that it is pristine.*

cisely localizes the spliced regions on most images. The FPN segmentation head enables localization of spliced regions of any size, while the information fusion from the different branches allows NoiseSeg to learn more splicing anomalies in order to detect them.

Branch ablation

We conducted an ablation study to show the effectiveness of employing multiple splicing localization methods as branches in NoiseSeg in combination with a FPN head. Training and evaluation was done on the CASIAv2 train set. The train set was split into a sub train set and a validation set with an 80/20 ratio. The MCC results for each evaluated branch combination is shown in the table below:

Branches	MCC
Noiseprint	0.3791
DnCNN	0.6352
ELA	0.6832
DnCNN+Noiseprint	0.6937
DnCNN+Noiseprint+ELA	0.7687

The MCC results for all tested branch combinations on the CASIAv2 validation set.

Only using the Noiseprint branch leads to a relatively low MCC score of 0.3791 as the resulting NoiseSeg model would have similar problems of coarse noise separation as the original model. Using only the DnCNN or ELA branch, by contrast, leads to significantly improved performance of a MCC score of 0.6352 and 0.6832, respectively. Both branches enable a much more refined splicing localization as they are not based on patch-based training like Noiseprint. However, these results can even be improved by leveraging the information of two or more branches and fusing them in the FPN head. When both the DnCNN and Noiseprint branch are employed, a MCC score of 0.6937 is achieved. This is the result of the Noiseprint branch being able to detect different types of splicing anomalies than the DnCNN branch, while the fine-grained noise features from DnCNN enable a much more precise localization of the detected Noiseprint anomalies. The results can be improved even more by further adding the ELA branch, which results in the best MCC score of 0.7687. It can be concluded that employing multiple branches to detect different types of splicing anomalies is highly effective.

Head ablation

We also conducted a head ablation study to determine the best performing head segmentation model. In total, we evaluated the segmentation models PAN, LinkNet, PSPNet, DeepLabV3, U-Net and FPN from the segmentation_models.pytorch framework. Again, the evaluation was done on the train set of the CASIAv2

dataset, which was split into a sub train set and a validation set with an 80/20 ratio. The MCC results for each evaluated head is shown in the table below:

Head	MCC
PAN	0.6849
LinkNet	0.7198
PSPNet	0.6425
DeepLabV3	0.7139
U-Net	0.7289
FPN	0.8129

The MCC results for all tested heads on the CASIAv2 validation set.

The PAN model and the PSPNet both have a considerably lower MCC score than the rest of the segmentation models with a MCC score of 0.6849 and 0.6425, respectively. The LinkNet, DeepLabV3 and the U-Net perform about equally good with a MCC score of 0.7198, 0.7139 and 0.7289, respectively. However, the model that clearly outperforms the other models is the FPN model with a MCC score of 0.8129. This is a MCC score difference of 0.084, which is significant. We hypothesise that this could be an effect of the FPN making predictions at all scales of its feature pyramid decoder. Image anomaly features that could potentially be lost in LinkNet, DeepLabV3 and the U-Net, while going through the decoder part of the network, could be recovered in the FPN decoder by making a prediction at every scale.

Conclusion

We introduce the novel image splicing localization model NoiseSeg that fuses statistical and CNN-based splicing localization methods alike to leverage the benefits of both. Unique splicing anomalies that can be identified by the Noiseprint, DnCNN and ELA branch all get fused in an FPN segmentation head to predict a precise localization of the spliced regions. A comparative evaluation of NoiseSeg against a number of state-of-the-art baselines such as ManTra-Net, Noiseprint, Splicebuster and multiple popular statistical methods demonstrate that NoiseSeg outperforms most state-of-the-art methods significantly.

References

[1] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Image forgery localization via fine-grained analysis of cfa artifacts. *IEEE Transactions on Information Forensics and Security*, 7(5):1566–1577, 2012.

[2] Ahmet Emir Dirik and Nasir Memon. Image tamper detection based on demosaicing artifacts. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 1497–1500. IEEE, 2009.

[3] Neal Krawetz and Hacker Factor Solutions. A picture's worth. *Hacker Factor Solutions*, 6(2):2, 2007.

[4] Shuiming Ye, Qibin Sun, and Ee-Chien Chang. Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In *2007 IEEE International Conference on Multimedia and Expo*, pages 12–15. Ieee, 2007.

[5] Weihai Li, Yuan Yuan, and Nenghai Yu. Passive detection of doctored jpeg image via block artifact grid extraction. 2009.

[6] Chryssanthi Iakovidou, Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. Content-aware detection of jpeg grid inconsistencies for intuitive image forensics. 2018.

[7] Babak Mahdian and Stanislav Saic. Using noise inconsistencies for blind image forensics. 2009.

[8] Siwei Lyu, Xunyu Pan, and Xing Zhang. Exposing region splicing forgeries with blind local noise estimation. 2014.

[9] 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. IEEE, 2015.

[10] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. 2019.

[11] Chuadhry Mujeeb Ahmed, Martin Ochoa, Jianying Zhou, Aditya P Mathur, Rizwan Qadeer, Carlos Murguia, and Justin Ruths. Noiseprint: Attack detection using sensor and process noise fingerprint in cyber physical systems. 2018.

[12] 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.

[13] Alexander Kirillov, Kaiming He, Ross Girshick, and Piotr Dollár. A unified architecture for instance and semantic segmentation, 2017.

[14] Jing Dong, Wei Wang, and Tieniu Tan. Casia image tampering detection evaluation database. IEEE, 2013.

[15] Nam Thanh Pham, Jong-Weon Lee, Goo-Rak Kwon, and Chun-Su Park. Hybrid image-retrieval method for image-splicing validation. *Symmetry*, 11(1):83, 2019.

[16] Gaël Mahfoudi, Badr Tajini, Florent Retraint, Frederic Morain-Nicolier, Jean Luc Dugelay, and PIC Marc. Defacto: Image and face manipulation dataset. IEEE, 2019.

[17] 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. Springer, 2014.

[18] Tiago José De Carvalho, Christian Riess, Elli Angelopoulou, Helio Pedrini, and Anderson de Rezende Rocha. Exposing digital image forgeries by illumination color classification. 2013.

[19] Adam Novozamsky, Babak Mahdian, and Stanislav Saic. Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. 2020.

[20] Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. Detecting image splicing in the wild (web). In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2015.

[21] 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):1–16, 2017.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[23] Pavel Yakubovskiy. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch,

2020.

[24] Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. A large-scale evaluation of splicing localization algorithms for web images. *Multimedia Tools and Applications*, page Accepted for publication.

Appendix - Results

This section includes the splicing localization results for the F1 an AP measure.

F1					
	DSO-1	CASIAv2	DEFACTO	IMD2020	WildWeb
ELA	0.192	0.141	0.054	0.197	0.149
BLK	0.168	0.112	0.053	0.161	0.128
CAGI	0.437	0.093	0.060	0.221	0.158
CFA1	0.175	0.069	0.039	0.143	0.138
CFA2	0.173	0.067	0.040	0.127	0.139
DCT	0.286	0.392	0.085	0.254	0.137
NOI1	0.246	0.027	0.040	0.082	0.025
NOI2	0.149	0.056	0.053	0.126	0.133
NOI4	0.190	0.014	0.023	0.066	0.017
NOI5	0.213	0.036	0.047	0.092	0.058
Splicebuster	0.332	0.074	0.094	0.181	**0.169**
ManTraNet	0.367	0.175	0.378	0.228	0.093
Noiseprint	0.488	0.076	0.097	0.243	0.151
NoiseSeg	**0.966**	**0.580**	**0.389**	**0.486**	0.148

The F1 test set results of NoiseSeg and the baselines over all dataset.

AP					
	DSO-1	CASIAv2	DEFACTO	IMD2020	WildWeb
ELA	0.188	0.153	0.053	0.180	0.112
BLK	0.197	0.107	0.045	0.174	0.144
CAGI	0.485	0.095	0.070	0.238	0.121
CFA1	0.243	0.054	0.042	0.128	0.102
CFA2	0.204	0.048	0.036	0.122	0.093
DCT	0.260	0.491	0.113	0.293	0.132
NOI1	0.436	0.071	0.065	0.215	0.176
NOI2	0.171	0.058	0.062	0.131	0.134
NOI4	0.210	0.044	0.029	0.113	0.109
NOI5	0.359	0.061	0.061	0.156	0.181
Splicebuster	0.655	0.061	0.112	0.215	0.1511
ManTraNet	0.497	0.188	0.415	0.279	0.093
Noiseprint	0.820	0.078	0.109	0.289	0.149
NoiseSeg	**0.987**	**0.652**	**0.517**	**0.586**	**0.246**

The AP test set results of NoiseSeg and the baselines over all dataset.

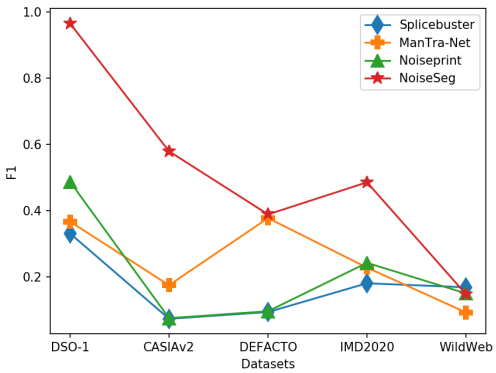

Figure 6. *Comparison of the F1 evaluation results of NoiseSeg and the baselines ManTra-Net, Noiseprint and Splicebuster over all datasets.*

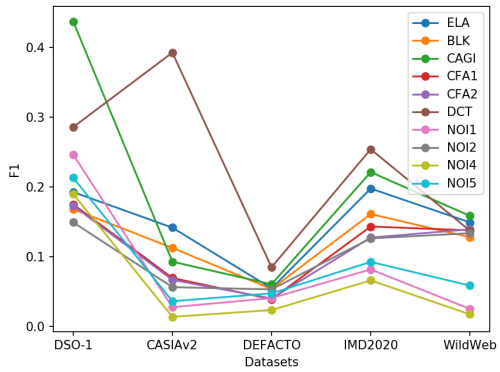

Figure 7. *Comparison of the F1 evaluation results of the statistical baselines over all datasets.*

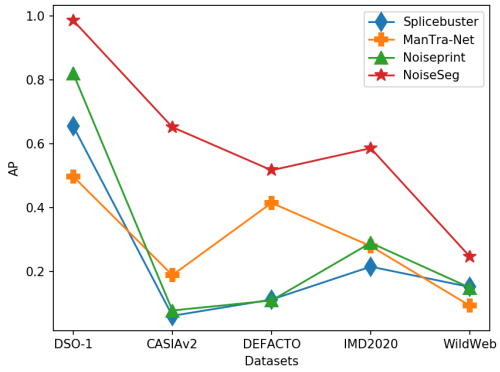

Figure 8. *Comparison of the AP evaluation results of NoiseSeg and the baselines ManTra-Net, Noiseprint and Splicebuster over all datasets.*

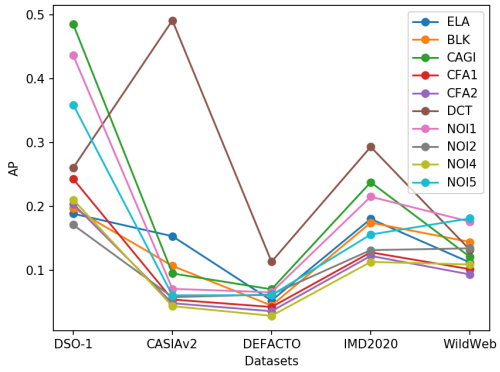

Figure 9. *Comparison of the AP evaluation results of the statistical baselines over all datasets.*