TextHydra for Text Tampering Detection

Yifei Wang
Fudan University
22302010062@m.fudan.edu.cn

Yueyuan Huang
Fudan University
21307110014@m.fudan.edu.cn

Yicheng Li
Fudan University
22302010058@m.fudan.edu.cn

Abstract

Detecting tampered text in document images poses unique challenges compared to natural image manipulation. Unlike natural images, text tampering often involves splicing or copy-move operations where background color, font size, and style closely match, making it difficult to rely solely on visual clues. High-frequency domain information can assist in identifying tampered regions, but such cues are often insufficient due to carefully forged text regions obscuring critical details.

To address these issues, we propose TextHydra, a novel framework tailored for text tampering detection and localization. TextHydra incorporates three complementary heads: (1) an RGB head for visual features, (2) a frequency head leveraging Discrete Cosine Transform (DCT) to extract Block Artifact Grids (BAG) inconsistencies, and (3) a noise head to capture residual out-of-distribution artifacts. These features are fused and processed through a transformer block for robust detection and precise localization.

Evaluated on the Tianchi Contest for Text Tampering Detection [34], TextHydra significantly outperforms existing methods, demonstrating its effectiveness in tackling this challenging task.

1. Introduction

1.1. Background

The robust verification of credentials and documents remains a fundamental and enduring concern within financial scenarios, holding significant implications across diverse sectors. Particularly in digital finance, the assessment of non-standard documentation is a common and critical task. To ensure security and foster trust, verification processes must confirm authenticity, reliability, and guarantee tamper-proof and forgery-proof properties. [1]

Yet, advancements in image manipulation techniques, including splicing, copy-move operations, and generative models, make manual detection of document tampering increasingly difficult. Therefore, a broadly applicable and efficient tampering detection algorithm is essential. Such a solution is critical for ensuring document authenticity, providing a reliable foundation for digital finance, and supporting the continued growth and stability of the digital financial industry.

1.2. Task

Our task is to develop a model capable of detecting forgery in an input document image and outputting the region of interest, represented by the upper-left and lower-right points of the rectangular tampered region. Detecting tampered text in document images poses unique challenges compared to manipulation detection in natural scene images. Unlike natural image tampering, text manipulation often involves splicing or copy-move operations where background color, font size, and style closely match, making it difficult to rely solely on visual clues. Additionally, while high-frequency domain information can assist in tampering detection, such cues are often insufficient due to carefully forged regions obscuring critical details. Understanding these distinctions and challenges is crucial, and further discussions on these aspects are provided in Section 2.

1.3. Dataset

The dataset used for this study consists of images of documents, either captured by cameras or taken from screenshots. To ensure robustness, the dataset also includes more diverse examples, such as photographs of shop signs. Among the tampered images, techniques such as copy-move, splicing, or generative models have been used. However, for tampered images, the authentic originals are not provided. Figure 1 illustrates a few examples.

The dataset is composed of a labeled training set and an unlabeled testing set, containing 13,000 and 1,200 images, respectively. The training set has a fairly balanced distribution, with 5,641 tampered images, close to half of the total (6,500). Each labeled image contains at most one tampered region.

[1] For simplicity, tamper and forge will be used interchangeably in this paper.

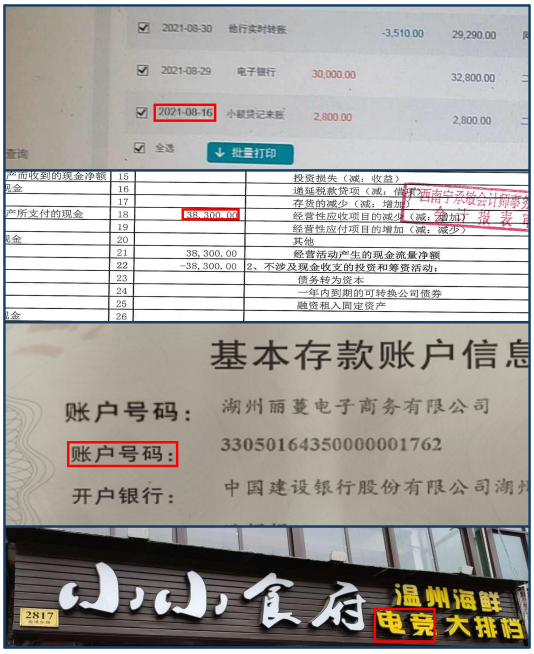

Figure 1. Examples of tampered images. From top to bottom: (1) Splicing in a screenshot, (2) Copy-move forgery in a scanned document, (3) Copy-move forgery in a document photograph, and (4) AI-generated forgery in a shop sign photograph.

2. Related Works

2.1. General Image Manipulation Detection

Early research in image forensics primarily addressed the detection of specific manipulation types, including copy-move [7] [10], splicing [8] [13] [19], and removal [16]. Acknowledging the inherent ambiguity of real-world tampering, researchers shifted focus towards general manipulation detection [17] [22] [20]. For instance, RGB-N [15] proposed a two-stream network, leveraging RGB features to capture visual anomalies and noise features to model inconsistencies between manipulated and authentic regions for localization. SPAN [22] takes a different approach, modeling pixel relationships within image patches at various scales through a pyramid of local self-attention blocks. Further, PSCCNet [23] employs hierarchical feature extraction with top-down and bottom-up pathways to determine image manipulation.

Drawing inspiration from the success of general-purpose object detection models like DETR [21] and subsequent works such as Sun *et al*. [24], a trend has emerged in image manipulation detection that leverages object-level modeling. For instance, ObjectFormer [25] exemplifies this approach by explicitly modeling consistency not only at the patch level but also by utilizing learnable embeddings as object prototypes. However, the direct application of object-level modeling presents a significant challenge in the context of document tampering detection. Unlike natural images with distinct objects, documents typically lack clearly defined semantic objects suitable for such modeling.

2.2. Document Image Tampering Detection

Early investigations into text forgery detection often relied on identifying specific handcrafted features indicative of tampering distortions. Some researchers framed this problem as printer source identification, recognizing external prints as potential forgeries [1] [18]. Others focused on analyzing distortions in elements such as fonts [6], text line orientation [5], geometry [9], image quality [2], DCT coefficients [4], and local texture patterns [11]. While offering high interpretability, these methods often suffer from limited generalization capabilities, particularly when confronted with more sophisticated or concealed manipulations.

Building upon the success of image manipulation detection in natural images, a prevalent trend in recent text forgery detection involves employing dual-stream encoders. These architectures aim to extract complementary information by processing the input through both the RGB domain and a transformed domain. For instance, Xu *et al*. [28] leverage residual filters within the second stream to capture subtle manipulation traces in pixel correlations. Frequency-based approaches are also common, with some methods integrating information from the discrete cosine transform [27] and high-pass filters [26]. Expanding beyond solely visual information, STFL-Net [30] incorporates OCR data alongside RGB input for detecting tampered text in screenshots. Similarly, DTD [29] and ASC-Former [33] propose utilizing JPEG compression traces, among others, coupled with a curriculum learning procedure, to enhance forgery detection. Our approach also adopts this multi-stream strategy, with a particular emphasis on the careful design of network heads to ensure the extraction of unique and non-overlapping information.

3. Method

In this section, we propose TextHydra, an innovative model tailored for document forgery detection. The overall architecture of TextHydra is illustrated in Figure 2. TextHydra comprises three primary components: (1) a Visual Perception Head designed to extract visual features from the original image, (2) a Frequency Perception Head aimed at capturing high-frequency information from the image, and (3) a Noise View module that explores distribution inconsistencies between forged and authentic regions. By synergistically integrating these complementary modalities, TextHydra achieves robust and accurate detection of forged content in document images.

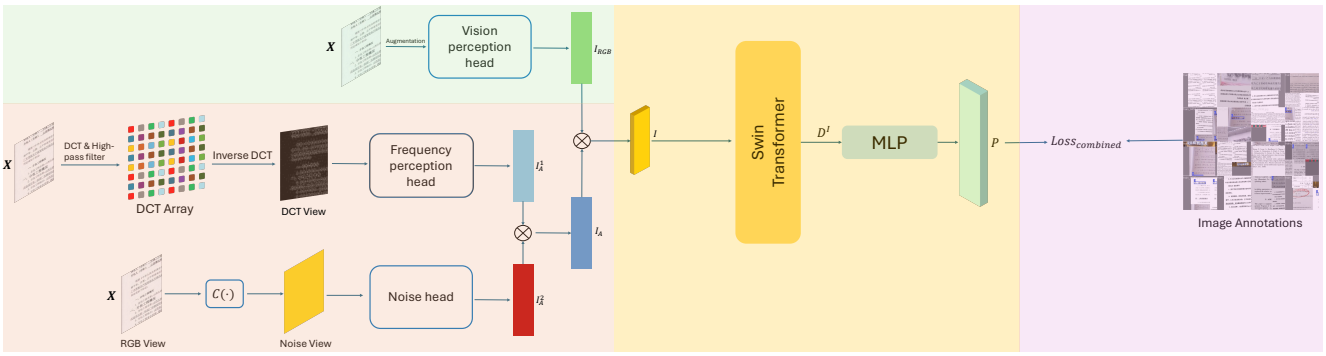

Figure 2. The overall architecture of our model. The model consists of three heads: the Visual Perception Head forms the main branch, while the Frequency Perception Head and Noise View act as auxiliary branches to assist the Visual Perception Head during inference. Different branches are represented using distinct background colors. The feature maps extracted from the three views are fused using the scSE module[14], followed by processing through an encoder-only architecture for detection and localization.

3.1. Visual Perception Head

We employ a pre-trained YOLO11 detection model [31] for our Visual Perception Head, using it as a feature map extractor. Specifically, we extract the output from the model's intermediate layer as the visual feature map, effectively capturing visual characteristics. To improve the robustness and generalization of the Visual Perception Head during training, we incorporate a series of data augmentation strategies, including random rotations, scaling, HSV space transformations, and brightness adjustments. These augmentations are designed to enhance the model's ability to adapt to diverse input variations.

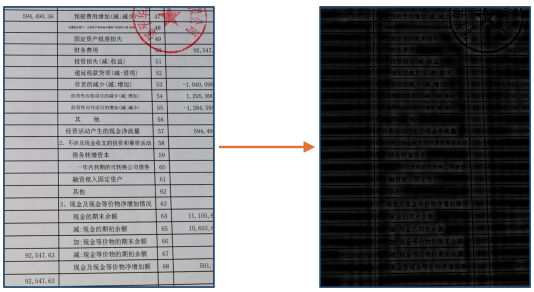

Figure 3. The image on the left represents the original document image, while the image on the right illustrates the result after applying Discrete Cosine Transform (DCT), high-pass filtering, and subsequently performing an Inverse Discrete Cosine Transform (IDCT). The transformation highlights high-frequency components while suppressing low-frequency information.

3.2. Frequency Perception Head

When images are acquired using digital devices such as cameras or smartphones, they are subjected to patching and compression operations, primarily through the quantization of Discrete Cosine Transform (DCT) coefficients. These processes inherently introduce Block Artifact Grids (BAG), a characteristic distortion pattern resulting from block-based compression techniques [3].Human vision primarily acts as a low-pass filter, effectively focusing on low-frequency information while overlooking high-frequency details such as edges and fine textures. To address this limitation and complement the Visual Perception Head, which mimics the role of human vision by extracting low-frequency features, we design a Frequency Perception Head to specifically capture high-frequency edge information critical for identifying tampered regions. This avoids redundant extraction of low-frequency content and ensures the model efficiently utilizes high-frequency details for enhanced inference.

In particular, we employ the Discrete Cosine Transform (DCT) to map the input image X into the frequency domain, where a high-pass filter is applied to isolate the high-frequency components. These components are subsequently transformed back to the spatial domain via the Inverse Discrete Cosine Transform (IDCT), enabling seamless feature interaction while preserving local consistency. Thus, the processed input can be represented as follows:

$$Z = \mathcal{T}_{\text{DCT}}^{-1}\left(\mathcal{F}_h\left(\mathcal{T}_{\text{DCT}}(X), \gamma\right)\right) \tag{1}$$

First, the input image X is transformed into the frequency domain using the Discrete Cosine Transform $\mathcal{T}_{\text{DCT}}(X)$. A high-pass filter $\mathcal{F}_h$ is then applied, parameterized by γ, to selectively extract high-frequency components, such as edges and texture inconsistencies. Unlike human vision, which inherently acts as a low-pass filter and focuses on low-frequency content, the Frequency Perception Head emphasizes high-frequency information. Finally, the extracted high-frequency components are transformed back to the spatial domain via the Inverse Discrete Cosine Transform $\mathcal{T}_{\text{DCT}}^{-1}$, producing Z, a high-frequency representation that

enhances the detection and localization of tampered regions.The processed image is subsequently passed through a second YOLO11 model [31], with its intermediate layer leveraged as the frequency-domain feature. The effect of processing the image using the operations described in Equation 1 is shown in Figure 3. As demonstrated, the transformation highlights high-frequency components, effectively suppressing low-frequency information to facilitate tampering detection.

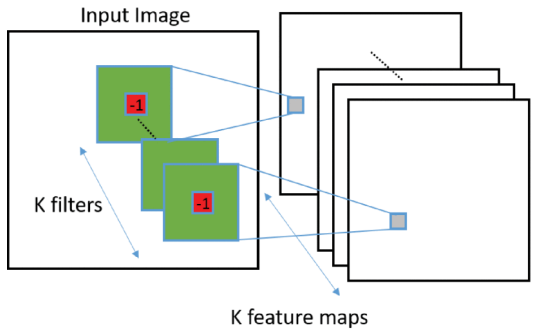

Figure 4. The constrained convolutional layer. The red coefficient is -1 and the coefficients in the green region sum to 1. [12]

3.3. Noise View

We utilize a learnable constrained convolutional layer to extract the noise view. Unlike standard convolutional layers, which primarily capture image features, the constrained convolutional layer is specifically designed for document forgery detection tasks, where the goal is to identify tampering footprints. The convolutional filter is constrained such that the weight at the center of the filter is fixed to -1, while the sum of all other weights is constrained to 1, as shown in Figure 4. This design allows the layer to effectively learn the distributional inconsistencies between the tampered regions and the authentic parts of the document.

The constraints applied to the convolutional weights can be described mathematically as follows:

$$\begin{cases} w_c(0,0) = -1, \\ \sum_{(i,j)\neq(0,0)} w_c(i,j) = 1, \end{cases} \tag{2}$$

where $w_c(i,j)$ represents the weights of the convolutional filter, $w_c(0,0)$ denotes the center weight, and $(i,j) \neq (0,0)$ refers to all non-center weights.

By imposing these constraints, the convolutional layer is guided to focus on capturing discrepancies between the real image and tampered regions, rather than general image features.

The intuition behind this design is further illustrated through the concept of prediction residuals. Specifically, the true pixel value is subtracted from the predicted pixel value, yielding the residual, as shown below:

$$r = \hat{I} - I, \tag{3}$$

where r is the prediction residual, $\hat{I}$ represents the predicted pixel value obtained from the constrained convolutional layer, and I denotes the true pixel value.

This residual highlights the differences between the predicted and actual pixel intensities, enabling the constrained convolutional layer to emphasize tampering footprints that deviate from the expected distribution. Through this mechanism, the constrained layer plays a critical role in capturing noise-based inconsistencies and enhancing the model's ability to detect forgeries.

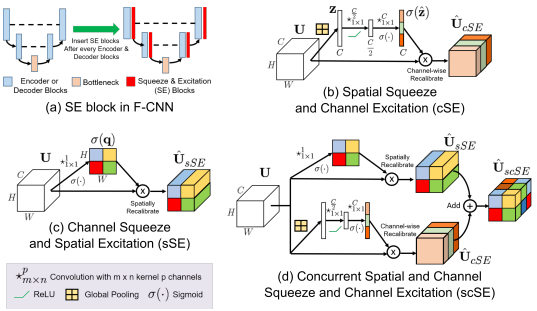

Figure 5. The architectural design of cSE, sSE and scSE blocks. [14]

3.4. Fusion

We utilize the Concurrent Spatial and Channel Squeeze & Excitation (scSE) module to fuse features from different realms. The scSE module combines the strengths of the sSE and cSE modules, allowing feature maps to be recalibrated along both the channel and spatial dimensions[14], as shown in Figure 5. This enhances meaningful features while suppressing weaker ones.

Given the output of the Frequency Perception Head, denoted as I_a^1, and the output of the Noise Map, denoted as I_a^2, the scSE module is applied to concatenate and incorporate these features. Afterward, a 1×1 convolution kernel is employed to reduce the number of channels, producing the feature map I_a. The feature map I_a is then concatenated with the output of the Visual Branch, denoted as I_{RGB}. The combined features are processed through a second scSE module, resulting in the final fused output I.

3.5. Localization

Inspired by [32], the fused output I is processed through an encoder-only transformer architecture, which produces the discriminative query embeddings D^I. These embeddings encode tampering cues from all three realms. To perform manipulation localization, we utilize a multi-layer perceptron (MLP) to process D^I, ultimately generating the mask vector P.

4. Experiment

We evaluate our model on the text tampering detection and localization task in the Tianchi contest [34], with the goal to localize the forgery region within the images or report there is none. Below, we introduce the experimental setup in Section 4.1, and present results in Section 4.2, and finally we perform an ablation study to justify the effectiveness of different components in Section 4.3, and show the visualization results in Section 4.4.

4.1. Experimental Settings

4.1.1. Testing Dataset

We use the dataset offered by the contest [34], which consists of a labeled training set with 13000 images and an unlabeled testing set with 1200 images. [2] To prevent overfitting, we partition the given labeled dataset with ratio $4:1$ to get an evaluation set of size 2600, with the rest being our training set.

4.1.2. Evaluation Metrics

The performance of the proposed method is evaluated by the F1 score. In detail, the region output by the model will be compared with ground truth, and is accepted if the Intersection over Union (IoU) surpasses some threshold. Numbers of TP(True Positive), TN(True Negative), FP(False Positive) and FN(False Negative) will then be calculated, as well as the precision rate $P = \frac{|TP|}{|TP|+|FP|}$ and recall rate $R = \frac{TP}{TP+FN}$. Finally F1 can be acquired by $F1 = \frac{2 \cdot P \cdot R}{P+R}$.

4.1.3. Implementation Details

All images are resized to 224×224. We use SGD for optimizaion with a learning rate of 0.001, which is decayed automatically. We train the complete model for 70 epochs with a batch size of 16, and early stopping is applied. We also leverage self-supervised learning.

4.1.4. Baseline Models

We compare our method with a few baseline models:

- **DTD** [29], which consists of a Frequency Perception Head to compensate the deficiencies caused by the inconspicuous visual features, and a Multi-view Iterative Decoder for fully utilizing the information of features in different scales.
- **ASC-Former** [33], which takes advantage of the complementarity of various transformed domains, with a plugin-play Tampered-Authentic Contrastive Learning module aiming to further increase its discrimination ability.
- **YOLO11** [31], which is the current latest model in the YOLO series, a pre-trained general-purpose architecture designed for both object detection and image segmentation.

[2] Since only one dataset is used, we will simply refer to it as *the dataset*.

For those output a mask instead of a bounding box, we append a function afterwards to do the transformation.

4.2. Experiment Results

We first test different settings of the YOLO11 model based on the Mean Average Precision (MAP) metrics. The criteria of YOLO11 is defined as

$$Loss = \lambda_{box} \cdot Loss_{box} + \lambda_{cls} \cdot Loss_{cls} + \lambda_{dfl} \cdot Loss_{dfl} \quad (4)$$

in which $Loss_{box}$ represents the bounding box regression loss, which measures the accuracy of the predicted bounding box coordinates by penalizing deviations from the ground truth. $Loss_{cls}$ denotes the classification loss, which quantifies the discrepancy between predicted and ground truth class probabilities, ensuring accurate object category predictions. $Loss_{dfl}$ refers to the Distribution Focal Loss, a specialized loss function that models bounding box regression as a discrete probability distribution over predefined bins. It minimizes the divergence between the predicted and target distributions, enabling the model to capture uncertainties and improve localization precision. The weights λ_{box}, λ_{cls}, and λ_{dfl} are hyperparameters that balance the contributions of these loss components during training.

Table 1 shows the results of YOLO11 on the evaluation set. Although the setting (7.5, 2.0, 1.0) maximizes MAP50, we picked (7.5, 0.5, 2.5) as our final setting as it outperforms others on the more comprehensive metric MAP50-90.

λ_{box}	λ_{cls}	λ_{dfl}	MAP50	MAP50-90
7.5	0.5	1.5	0.892	0.679
7	0.5	2	0.897	0.685
7.5	1	1	0.891	0.681
7.7	0.1	1.7	0.844	0.686
7.0	0.2	2.3	0.878	0.672
7.5	0.5	2.5	0.914	**0.698**
9.0	0.5	1.0	0.906	0.683
7.5	2.0	1.0	**0.921**	0.671

Table 1. MAP50 and MAP50-90 of YOLO11 results. The former represents the MAP calculated at an IoU threshold of 0.50, and the latter stands for MAP calculated across multiple thresholds from 0.50 to 0.90 in increments of 0.05.

We list the F1 scores in Table 2, from which we can observe TextHydra markedly outperforms others on the dataset.

It is worth noting that the training input and output formats of DTD and ASC-Former include masks, which are not fully aligned with the requirements of our task. During the process of converting masks to regions, information loss may occur (e.g., determining whether small regions corre-

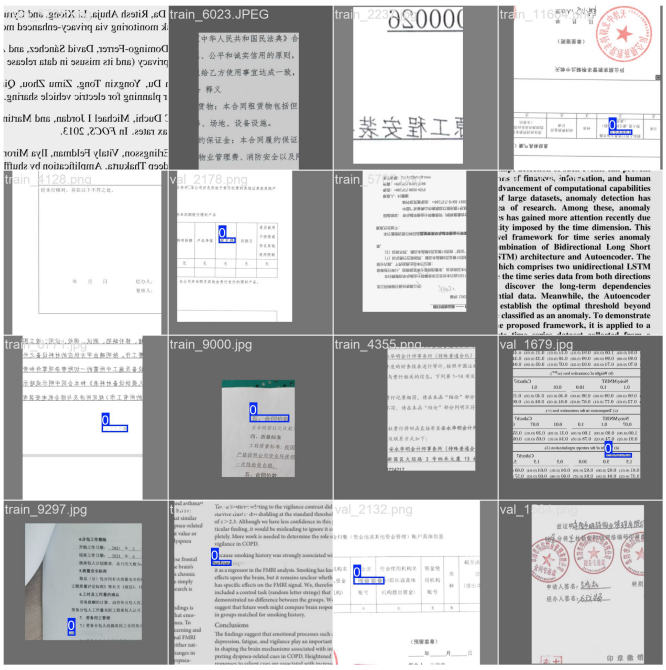

Figure 6. Visualization of the predicted tampering region by our method.

Method	F1
DTD	56.22
ASC-Former	35.84
YOLO11	78.84
Ours	**94.27**

Table 2. Comparisons of F1 scores (%) of different models.

Variants	F1
Baseline (YOLO11)	78.84
Baseline + FH	86.96
Baseline + NH	84.38
Baseline + FH + NH	**87.63**

Table 3. Ablation results on the dataset using different combination of heads.

sponding to mask fragments should be filtered out). Consequently, the reproduced results for these two models show suboptimal performance in our experiments.

4.3. Ablation Study

Firstly, to evaluate the effectiveness of different heads of our method, we test the baseline model with Frequency Perception Head (FH) and the Noise Head (NH) separately on the dataset. Trivial concatenation is used for fusion, and self-supervised learning is not used. The results are shown in Table 3, from which we can see the F1 score increase by 8.12% with FH, by 5.54% with NH, and by 8.79% with both. The progress validates that the use of additional heads FH and NH effectively improves performance of our model.

We then evaluate the effectiveness of the scSE module adopted and the self-supervised learning (SSL) procedure adopted in our method. The results are listed in Table 4. We can observe that the F1 score decrease by 5.52% without scSE, while decrease by 2.25% without SSL, which demonstrates the effectiveness of these two techniques.

Variants	F1
w/o scSE	88.75
w/o SSL	92.02
Ours	**94.27**

Table 4. Ablation results on the dataset using different variants of TextHydra.

4.4. Visualization Results

We provide a few results of TextHydra in Figure 6. The results demonstrate that our method is capable of detecting tampering precisely, even when the forgery is extremely tiny.

References

[1] Christoph H. Lampert, Lin Mei, and Thomas M. Breuel. Printing technique classification for document counterfeit detection. In *2006 International Conference on Computational Intelligence and Security*, volume 1, pages 639–644, 2006. 2

[2] Seung-Jin Ryu, Hae-Yeoun Lee, Il-Weon Cho, and Heung-Kyu Lee. Document forgery detection with svm classifier and image quality measures. In *Advances in Multimedia Information Processing - PCM 2008*, pages 486–495, 12 2008. 2

[3] Weihai Li, Yuan Yuan, and Nenghai Yu. Passive detection of doctored jpeg image via block artifact grid extraction. *Signal Processing*, 89(9):1821–1829, 2009. 3

[4] Zhouchen Lin, Junfeng He, Xiaoou Tang, and Chi-Keung Tang. Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis. *Pattern Recognition*, 42(11):2492–2501, 2009. 2

[5] Joost Beusekom, Faisal Shafait, and Thomas Breuel. Text-line examination for document forgery detection. *International Journal on Document Analysis and Recognition (IJDAR)*, 16:189–207, 06 2012. 2

[6] Romain Bertrand, Oriol Terrades, Petra Gomez-Krämer, Patrick Franco, and Jean-Marc Ogier. A conditional random field model for font forgery detection. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 576–580, 08 2015. 2

[7] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Efficient dense-field copy–move forgery detection. *IEEE Transactions on Information Forensics and Security*, 10(11):2284–2297, 2015. 2

[8] 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. 2

[9] Shize Shang, Xiangwei Kong, and Xingang You. Document forgery detection using distortion mutation of geometric parameters in characters. *Journal of Electronic Imaging*, 24:023008, 03 2015. 2

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

[11] Francisco Cruz, Nicolas Sidere, Mickaël Coustaty, Vincent Poulain d'Andecy, and Jean-Marc Ogier. Local binary patterns for document forgery detection. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 1223–1228, 11 2017. 2

[12] Belhassen Bayar and Matthew C. Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706, 2018. 4

[13] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros. Fighting fake news: Image splice detection via learned self-consistency, 2018. 2

[14] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. Concurrent spatial and channel squeeze & excitation in fully convolutional networks, 2018. 3, 4

[15] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning rich features for image manipulation detection, 2018. 2

[16] Xinshan Zhu, Yongjun Qian, Xianfeng Zhao, Biao Sun, and Ya Sun. A deep learning approach to patch-based image inpainting forensics. *Signal Processing: Image Communication*, 67:90–99, 2018. 2

[17] Jawadul H. Bappy, Cody Simons, Lakshmanan Nataraj, B. S. Manjunath, and Amit K. Roy-Chowdhury. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 28(7):3286–3300, July 2019. 2

[18] Maryam Bibi, Anmol Hamid, Momina Moetesum, and Imran Siddiqi. Document forgery detection using printer source identification—a text-independent approach. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 8, pages 7–12, 2019. 2

[19] 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. Curran Associates, Inc., 2019. 2

[20] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9535–9544, 2019. 2

[21] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020. 2

[22] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. SPAN: spatial pyramid attention network forimage manipulation localization. *CoRR*, abs/2009.00726, 2020. 2

[23] Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu. Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization, 2022. 2

[24] Shuyang Sun, Xiaoyu Yue, Song Bai, and Philip Torr. Visual parser: Representing part-whole hierarchies with transformers, 2022. 2

[25] Junke Wang, Zuxuan Wu, Jingjing Chen, Xintong Han, Abhinav Shrivastava, Ser-Nam Lim, and Yu-Gang Jiang. Objectformer for image manipulation detection and localization, 2022. 2

[26] Yuxin Wang, Hongtao Xie, Mengting Xing, Jing Wang, Shenggao Zhu, and Yongdong Zhang. Detecting tampered scene text in the wild. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, page 215–232, Berlin, Heidelberg, 2022. Springer-Verlag. 2

[27] Yuxin WANG, Boqiang ZHANG, Hongtao XIE, and Yongdong ZHANG. Tampered text detection via rgb and frequency relationship modeling. *Chinese Journal of Network and Information Security*, 2022. 2

[28] Wenbo Xu, Junwei Luo, Chuntao Zhu, Wei Lu, Jinhua Zeng, Shaopei Shi, and Cong Lin. Document images forgery localization using a two-stream network. *International Journal of Intelligent Systems*, 37(8):5272–5289, 2022. 2

[29] Chenfan Qu, Chongyu Liu, Yuliang Liu, Xinhong Chen, Dezhi Peng, Fengjun Guo, and Lianwen Jin. Towards robust tampered text detection in document image: New dataset and new solution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5937–5946, June 2023. 2, 5

[30] Zeqin Yu, Bin Li, Yuzhen Lin, Jinhua Zeng, and Jishen Zeng. Learning to locate the text forgery in smartphone screenshots. In *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 06 2023. 2

[31] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. 3, 4, 5

[32] Shuaibo Li, Wei Ma, Jianwei Guo, Shibiao Xu, Benchong Li, and Xiaopeng Zhang. Unionformer: Unified-learning transformer with multi-view representation for image manipulation detection and localization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12523–12533, 2024. 4

[33] Dongliang Luo, Yuliang Liu, Rui Yang, Xianjin Liu, Jishen Zeng, Yu Zhou, and Xiang Bai. Toward real text manipulation detection: New dataset and new solution, 2024. 2, 5

[34] Tianchi. Contest for text tampering detection. Available at https://tianchi.aliyun.com/competition/entrance/532223, 2024. 1, 5