Text Guided Image Manipulation Detection

Abstract

Identifying any image regions that have been manipulated is a key component of defending against malicious use of image editing tools. Prior work has addressed a task where a model is given an image and the goal is to localize any regions where the image has been altered. However, in many applications of these detectors (*e.g.*, searching through social media), there are important contextual cues, such as image captions, that may provide insight into the objects that are likely to be manipulated. In this paper we propose a new task, Text Guided Image Manipulation Detection, where the goal is to take an image and associated text (*e.g.*, image captions, labels, or tags) as input and predict whether/where images were altered. Note that we consider cases where the contextual text both is and is not related to the manipulation, and expect that a good detector should work well in both settings. To address this task, we introduce Text-Conditioned Manipulation Detector (TCMD), which refines the initial prediction of the image manipulation by reconsidering the image regions associated with the contextual information. We find that our TCMD model that uses text guidance performs better than an image-only model regardless of whether the manipulation is related to associated text or not, demonstrating the potential impact for our work.

1 Introduction

The prevalence of image editing tools and generative AI models [24, 5, 22, 34, 18, 23, 29] has made altering digital images increasingly accessible to everyone. However, while enhancing image quality, these techniques also heighten concerns regarding the propagation of fake news. Multiple models have been proposed for image manipulation detection [40, 2, 27, 8, 26] that can detect and localize various types of manipulations like copy-move, splicing, inpainting and removal. The majority of existing research in the domain of image manipulation detection is limited predominantly to image data as shown in Fig. 1(a). However, in real-world scenarios, we often encounter multi-modal data that encompasses text, audio, and images. Such diverse modalities carry an abundance of valuable information that can significantly enhance the accuracy of manipulated image detection, yet, their integration into image manipulation detection systems remains an underexplored area. For example, in the context of social media platforms, images are frequently accompanied by textual descriptions, comments, and sometimes audio narratives. These accompanying elements offer crucial context, providing additional layers of information that may reveal discrepancies or anomalies indicative of image manipulation.

To address these shortcomings, we introduce "Text Conditioned Manipulation Detection Model (TCMD)" which is inspired by our above ideas; we show the effectiveness of our proposed work by comparing it with the existing unimodal models. Here, we want to have a manipulation detection system leverage text data associated with images to guide the detection process and possibly enhance the accuracy and efficiency of identifying the manipulated regions as shown in Fig. 1(b).

Not only is this task integral to addressing misinformation, it also gives a richer contextual understanding that could reveal inconsistencies or manipulations that might not have been readily evident in the image itself. Our task is challenging because establishing an accurate mapping between text and image can be non-trivial. A caption could be ambiguous, subjective and may not correspond to image content at all. Even in the cases where text is relevant to the image it might not contain information about the manipulated objects. Consider an example where a person (Alex) has manipulated an image of himself in the park, where he added his dog Benjamin standing beside him. Depending on the type

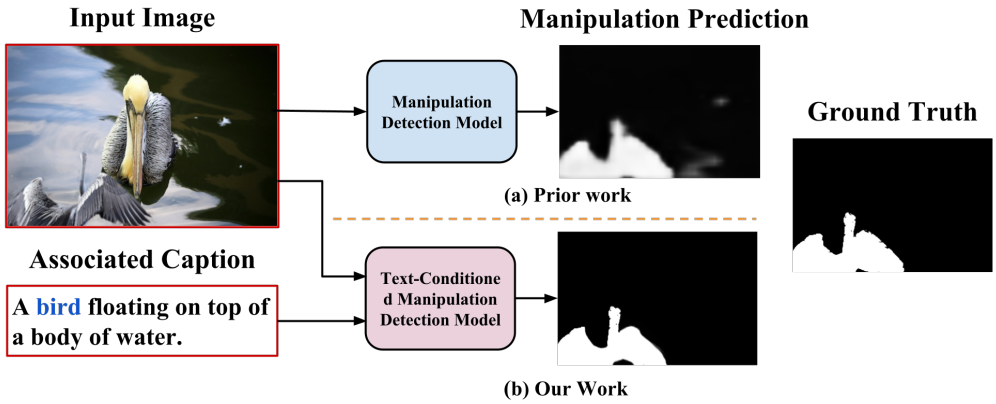

Figure 1: **Comparison of Prior work vs. Our work**. Left side is the input to our task (taken from [30, 25]) which consists of both the manipulated image and the caption associated with it. Right side of the image has two parts. Part (a) is prior work, a generic manipulation detection model; here manipulations can be both deep learning based and non deep learning based. Part (b) is our work, where we introduce the text guided model which uses other modalities of data (in our case text) associated with an image as input and helps in performance improvement over the previous approaches.

of caption he posts there could be various challenges. If the caption reads "Had a nice time with my dog Benjamin, heading to the beach now!", it is trivial to map the objects in the caption to the image and thus guide the manipulation model to detect the tampering. But, if the caption reads something like "Having nice time with Mr. Benjamin!", it is challenging to map Benjamin to Alex's dog in the image. It is even more challenging in the case where the caption is completely irrelevant like "Going to the beach!" and does not provide any information about what is going on in the image.

While numerous existing models, [8, 42, 27, 41] have shown successful applications in image manipulation detection, they have predominantly overlooked the task of incorporating text or other modalities. Consequently, the challenges inherent to this task remain largely unexplored. It is crucial to address this because in many real-world scenarios, such as on social media platforms, images are often accompanied by relevant text which often mentions the manipulated object, so the systems that do not consider this information might not be as effective.

In order to address this challenge we make the following contributions. We introduce a new model for detecting tampering in images guided by the accompanying text. Our model has three main components: The first component is a unimodal manipulation detection model [41, 8, 27, 43, 28], through which we only pass the image input and obtain the prediction mask. The second component is text-conditioned semantic segmentation model, where we obtain a semantic segmentation mapping from an off-the-shelf model and condition the caption on this segmentation mask. This is done by obtaining all the words from the caption and retaining just the objects mentioned in the caption from the semantic segmentation output. Now that we have our prediction from manipulation detection model and output from semantic segmentation model (text-conditioned), we use this data as a two-channel input to train a simple convolutional neural network. Through this straightforward procedure we observe that the performance of a text-based model is slightly better than the image-only model. We also show that our model works better in settings where the caption is relevant to image vs. when caption is not relevant. Finally, we introduce the new GLIDE-Inpainting Dataset, which we use to test all our models.

2 Related Work

Image manipulation detection has been an area of active research for a number of years. Early research in this area primarily focused on detecting a single type of manipulation, such as splicing [7, 13, 16, 3], copy-move [6, 14], or inpainting [45, 15, 41, 19]. In addition to these singular manipulation detection techniques, some efforts have been made to create unified models [46, 20, 38, 36, 1, 47, 44, 31, 2]

that can detect and localize multiple manipulation types, including splicing, copy-move, and removal manipulations.

Inpainting detection in images has been another active area of exploration [15, 45, 21]. For example, Haiwei et al. [41] introduced IID-Net, a model composed of three distinct blocks: an Enhancement block for enhancing inpainting traces, an Extraction block for feature extraction, and a Decision block for inpainted region detection at pixel-level accuracy.

Some approaches aim to identify tampered artifacts without particular concern for the manipulation type [17]. Yet, they are limited when dealing with manipulations like inpainting, which leave minimal traces. Attempts have been made to develop more generalized models capable of detecting various kinds of manipulations that do not rely solely on artifacts in images [42, 8, 33, 11, 12, 27, 37]. One such model is the PSCC-Net [27], a general manipulation detection model that processes images via a two-path procedure. The first path extracts both local and global features (top-down), while the second path determines whether the image has been manipulated and estimates the manipulation masks at multiple scales (bottom-up), with each mask being conditioned on the previous one.

An interesting approach was presented in [8], where the authors propose multi-view feature learning to simultaneously exploit tampering boundary artifacts and the noise view of the input image. They argue that these features are semantics-agnostic and hence, generalizable. To effectively learn from authentic images, they incorporated multi-scale (pixel/edge/image) supervision in their training. TransForensics [11] is an innovative image forgery localization method that utilizes dense self-attention encoders to model the global context and all pairwise interactions between local patches at different scales. The Spatial Pyramid Attention Network (SPAN) [12] is another novel architecture developed for detecting and localizing multiple types of image manipulations. SPAN constructs a pyramid of local self-attention blocks, effectively modeling the relationship between image patches at multiple scales. ObjectFormer [37] captures subtle manipulation traces invisible in the RGB domain, extracts high-frequency features of images and combines them with RGB features as multimodal patch embeddings.

It is worth noting that previous methods primarily focused on the detection of manipulations in the visual domain and did not explore the use of other modalities. Our research, on the other hand, aims to investigate the effectiveness of text guidance in image manipulation detection.

3 Text-Guided Image Manipulation Detection and Localization Network

Given an image and text data associated with it, our objective is to develop a model for image manipulation detection guided by the text data. Manipulations in these images can take various forms, such as copy-move, splicing, removal, or inpainting. We introduce Text-Conditioned Manipulation Detector (TCMD) which is designed to leverage both image and associated text to improve the accuracy of detecting and localizing manipulated regions within an image. Previous methods for image manipulation detection primarily focused on analyzing image data alone, disregarding valuable contextual information. In contrast, our proposed approach utilizes the associated text data (e.g., image captions, labels, or tags) to guide the manipulation detection process. By using the context provided by the text, our model gains insights into the objects that are likely to be manipulated, leading to improved detection performance. Here we design our model to consider both cases where the manipulation is related to the associated text and cases where it is not, ensuring that it can effectively detect manipulations in various scenarios.

Formally we assume that input of our end-to-end model consists of an RGB image $I \in \mathbb{R}^{H \times W \times 3}$ and captions T associated with the image as shown in Fig. 2(a). The final output is a binary mask $B_o \in \mathbb{R}^{H \times W \times 1}$ which has a pixel value 1 if that part of the image was manipulated and a pixel value 0 if it was not. So, our Text-Conditioned Model $\mathbf{G} : \{\mathbb{R}^{H \times W \times 3}, T\} \rightarrow \mathbb{R}^{H \times W \times 1}$ takes I and T as input and returns B_o as output (Fig. 2(c)). On a high level, our TCMD model consists of three parts as shown in Fig. 2(b). Firstly, we have a traditional unimodal Manipulation Detection Model which takes the input image I and returns an image-only predicted mask $\text{Image}_{pred} \in H \times W \times 1$ with each pixel values in [0, 1]. Next, we have our Semantic Segmentation model which takes I as input and returns a segmentation mask S where each element $S(i, j)$ in the classification map of S represents the predicted class label for the corresponding pixel at position (i, j) in the input image. The class labels are typically represented by integer values, so $S(i, j) \in 1, 2, ..., C$, where C is the total number of classes which in our case are the detected objects (which could have potentially been

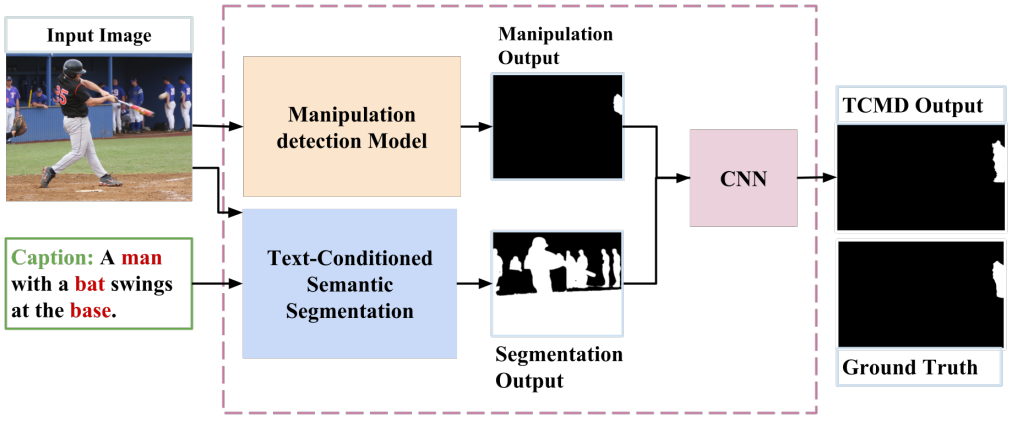

Figure 2: **Text Guided Image Manipulation Detection**. Here we propose an approach where we retrieve a post from Social Media which generally has a caption associated with an image. We then show that this caption may guide a manipulation detection model towards the object that was tampered with. This is shown by increase in performance of our proposed model vs. the unimodal model. Our method consists of a manipulation detection model which takes image as input and an object detection model which takes as the image and text as input. These output masks that are given to a Convolutional Network which then directs the model towards localizing the object that has been manipulated and ignoring any noise predicted by the manipulation model.

manipulated). We convert S into a binary mask $\text{Text}_{pred} \in H \times W \times 1$. Here we condition on T to select just the classes that were mentioned in the captions and combine them to get an output Text_{pred} where each pixel value is 1 if object is present in the caption and 0 otherwise. Finally we pass both Image_{pred} and Text_{pred} into our Text-Modulated Manipulation Detector as inputs and get the final prediction mask B_o.

The intuition behind this is that text/captions that are present along with the images in most cases will have the tampered objects mentioned in the caption. This is especially true in case of articles or posts that are published with the intention to mislead the audience. And in the cases this is true, our model is able to direct the attention towards those objects and in turn reduce the noise associated with the other parts of the image.

3.1 Manipulation Detection Model

In this first step of our task, we take a standard unimodal manipulation detection model, lets call it M. We pass our image input $I \in \mathbb{R}^{H \times W \times 3}$ through this model and get the model's prediction:

$$\mathbf{M}(x_i) = m_i,$$

where y_i is the prediction of this model which is of the form $y \in \mathbb{R}^{H \times W \times 1}$, with each value of $m_i \in [0, 1]$. The next step is text-conditioned semantic segmentation.

3.2 Text-Conditioned Semantic Segmentation (TCSS)

In the second step of our process we pass in the caption and input both into our Text-Conditioned Semantic Segmentation model. Let's denote the segmentation model as S, the input image as $I = x_i$ and one input caption as c_i. We assume the output of the semantic segmentation model to be:

$$\mathbf{S}(x_i) = s_i, \forall s_i(x, y) \in [0, n],$$

where n is the total number of categories of objects present in the image. This output is then conditioned on the caption c_i:

$$\mathbf{D}(s_i \mid c_i) = \begin{cases} 1, & \text{if } s_i(x, y) \in C \\ 0, & \text{otherwise} \end{cases}$$

Here we say that C is a set of all the words (objects) present in the caption and the above equation picks out the objects from the semantic segmentation output s_i which belong to the caption c_i. We define our TCSS as:

$$\textbf{TCSS}(x_i, c_i) = D(S(x_i)) = t_i$$

Here we leverage an off-the-shelf semantic segmentation model and use the words in the input caption to generate masks that represent the objects in the caption to combine them together and get a unified segmentation map representing all the objects present in the text.

In our experiments, we have used MS COCO [25] masks for each image directly instead of using a segmentation model. Each image in MS COCO has a set of ground truth semantic segmentation masks that represent different objects present. This will help us see what will happen in the case when we have perfectly segmented masks. So here, for each image we take all the segmentation masks and given the caption for an image (after processing the caption by lemmatizing, tokenizing and removing stop words), we retrieve all the potential objects that are present in the image. Then when we pick the segmentation masks that represent objects present in the caption. In the end we combine all the masks to get one final mask that segments out all the objects mentioned in the caption.

3.3 Text-conditioned Manipulation Prediction

Now, in this third part of our model, we combine both our outputs from the Manipulation Detection model and from the TCSS. Here we experiment with a simple CNN architecture to recognize whether associated text has any relevance to the manipulated image. We define our CNN architecture to be:

$$\begin{aligned}
Z_1 &= \text{Conv}(tm_i, W_1) + b_1 \\
A_1 &= \text{ReLU}(Z_1) \\
Z_2 &= \text{Conv}(A_1, W_2) + b_2 \\
A_2 &= \text{ReLU}(Z_2) \\
Z_3 &= \text{Conv}(A_2, W_3) + b_3 \\
O &= \text{sigmoid}(Z_3)
\end{aligned}$$

Here $Conv$ represents the convolution operation, $ReLU$ represents the Rectified Linear Unit activation function, $W1, W2, W3, b1, b2, b3$ represents the learnable weights and biases respectively, tm_i is the combined two-channel input, consisting of both the outputs from TCSS and Manipulation detection Model. Overall we can represent our CNN as:

$$O = \text{sigmoid}(\text{Conv}_3(\text{ReLU}(\text{Conv}_2(\text{ReLU}(\text{Conv}_1(tm_i, W_1) + b_1), W_2) + b_2), W_3) + b_3)$$

As mentioned in Fig. 2, O represents our final output mask giving us the information about where the manipulation has happened. The intuition behind this step is to see if a simple CNN-like function can find relations between images and associated text. It might not always be the case that a caption will help the CNN to detect manipulations but a more sophisticated model should be able to understand when the caption will help and when it does not, and we want our paper to be one step towards building that model.

4 Experiments

4.1 Datasets

The problem we tackle here consists of different types of manipulations including splicing, copy-move, removal and inpainting. Thus we use a diverse set of datasets that contain images pertaining to all the mentioned manipulations. We use DEFACTO [30] and CASIAV2 [9] datasets to train our model and baselines to perform manipulation detection on images that have been forged by splicing, removal or copy-move. We also created an inpainting dataset by using the recent GLIDE [32] model on MS COCO dataset [25].

4.2 GLIDE-Inpainting Dataset Generation

As mentioned in the previous section we generated an inpainting dataset using GLIDE [32]. We sampled images from MS COCO [25] dataset (statistics mentioned in Table 3). Then for each sampled

image we took the relevant segmentation masks and captions from MS COCO. We selected one caption and retrieved the object categories for each image. Then we used word embeddings to divide categories into two sets: caption relevant categories and caption non-relevant categories. The specific word embeddings we used were fastText [4], where we used the cosine similarity metric to see how similar our caption was to the MS COCO category. We tokenized, lemmatized and removed stop words from the caption, and computed cosine similarity between each of the words in the caption and the object categories, and took the max of all the similarity values to represent how similar this caption is to each category.

$$\textbf{CosineSimilarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

The intuition behind this is that, if the MS COCO category (like "dog") is present in the caption it will give the cosine similarity close to 1 and the max value of similarity of the preprocessed caption will be 1. Even if the caption does not contain the exact word, it could contain a related word, e.g, "animal". Using this procedure we divide the categories into two parts based on a threshold of 0.9.

After we obtain two different masks for each image representing two different categories (one present in the caption and one not), we use GLIDE to inpaint the image with each of these two masks and their categories as prompts. The idea behind having two types of images is to see how our model performs when the text is relevant to the manipulated object vs. when it is not.

Table 1: Quantitative comparisons of baseline models by using pixel-level AUC as the critic. The models mentioned below are trained on DEACTO[30] and tested on GLIDE[32], CASIAV2[9] and TEST-set of DEFACTO. Thiss table gives an overaall performance of these models and also statistics of how the model performed on images that have relevent captions (denoted by r-caption) and non-relevant captions (denoted by nr-caption).

Models		MVSS	MVSS + Text	IID	IID + Text
GLIDE	Overall	0.389	0.573	0.316	0.553
	r-Caption	0.320	0.594	0.244	0.630
	nr-Caption	0.415	0.324	0.341	0.517
CASIAV2	Overall	0.610	0.536	0.628	0.497
	r-Caption	0.605	0.568	0.625	0.492
	nr-Caption	0.607	0.504	0.629	0.499
DEFACTO-TEST	Overall	0.788	0.632	0.681	0.738
	r-Caption	0.771	0.762	0.663	0.762
	nr-Caption	0.784	0.579	0.694	0.699

Table 2: Quantitative comparisons of baseline models by using pixel-level AUC as the critic. The models mentioned below are trained on DEACTO[30] and tested on GLIDE[32], CASIAV2[9] and TEST-set of DEFACTO. This table gives an overaall performance of these models and also statistics of how the model performed on different manipulation types of the images present in the above mentioned datasets.

Models		MVSS	MVSS + Text	IID	IID + Text
GLIDE	Overall	0.389	0.573	0.316	0.533
CASIAV2	Overall	0.610	0.536	0.628	0.497
	Splicing	0.656	0.566	0.655	0.482
	Copymove	0.586	0.520	0.614	0.511
DEFACTO-TEST	Overall	0.788	0.632	0.681	0.738
	Splicing	0.851	0.710	0.789	0.773
	Copymove	0.740	0.609	0.680	0.720
	Inpainting	0.741	0.720	0.574	0.723

Table 3: Quantitative comparisons of baseline models by using pixel-level AUC as the critic. The models mentioned below are trained on DEACTO[30] and tested on GLIDE[32], CASIAV2[9] and TEST-set of DEFACTO. Thiss table gives an overaall performance of these models and also statistics of how the model performed on images that have relevent captions (denoted by r-caption) and non-relevant captions (denoted by nr-caption).

Models		PSCC	PSCC + Text	UperNet	UperNet + Text
GLIDE	Overall	0.584	0.699	-	-
	r-Caption	0.628	0.906	-	-
	nr-Caption	0.539	0.497	-	-
CASIAV2	Overall	0.798	0.524	-	-
	r-Caption	0.798	0.237	-	-
	nr-Caption	0.798	0.811	-	-
DEFACTO-TEST	Overall	0.853	0.760	-	-
	r-Caption	0.820	0.810	-	-
	nr-Caption	0.760	0.666	-	-

Table 4: Quantitative comparisons of baseline models by using pixel-level AUC as the critic. The models mentioned below are trained on DEACTO[30] and tested on GLIDE[32], CASIAV2[9] and TEST-set of DEFACTO. This table gives an overaall performance of these models and also statistics of how the model performed on different manipulation types of the images present in the above mentioned datasets.

Models		PSCC	PSCC + Text	UperNet	UperNet + Text
GLIDE	Overall	0.584	0.699	-	-
CASIAV2	Overall	0.798	0.524	-	-
	Splicing	0.915	0.526	-	-
	Copymove	0.734	0.523	-	-
DEFACTO-TEST	Overall	0.853	0.760	-	-
	Splicing	0.943	0.884	-	-
	Copymove	0.840	0.798	-	-
	Inpainting	0.777	0.647	-	-

4.3 Metrics

For all our experiments we use The Area Under the receiver operating characteristic Curve (ROC-AUC) [10] in the pixel domain as the evaluation criterion. This was used by most prior work for the task of image manipulation detection [8, 41, 27] and we follow them for better comparison of our model. Here, for each output we obtain, we compute the AUC between it and the ground-truth mask. This is done by plotting the fraction of true positives out of the positives (TPR = true positive rate) vs. the fraction of false positives out of the negatives (FPR = false positive rate), at various threshold settings [39].

4.4 Baselines

In our evaluations, we compare our approach against several state-of-the art manipulation detection models [27, 43, 8, 41] as shown in Tables 2 and 3.

In addition to that we also evaluate our test dataset on MS-COCO masks [25]and RISE masks [35]. In case of COCO masks we compute ROC between the MS COCO category mask and our ground truth. This essentially gives a baseline for how well the model would perform in the case of perfect semantic segmentation with the caption provided in the context. As for RISE masks, we take several random masks (2000 in our case) and for each image we compute AUC on all these 2000 masks and compute the average. We do this for the entire dataset and this essentially gives us a random baseline.

5 Conclusion

In conclusion, this paper introduces a novel task called Text Guided Image Manipulation Detection, where both the image and associated text are used to predict whether and where image manipulations have occurred. Currently, our experiments with various models and datasets still donot show any conclusive evidence of text being directly related to the take of image manipulation. But this research highlights the potential impact of incorporating textual context in image manipulation detection, which can be might be valuable in some applications, and hopefully in teh future can help combat the malicious use of image editing tools and enhancing the reliability of image content across different platforms and media.

References

[1] S. Agrawal, P. Kumar, S. Seth, T. Parag, M. Singh, and V. Babu. Sisl:self-supervised image signature learning for splicing detection & localization. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 22–32, 2022.

[2] Vishal Asnani, Xi Yin, Tal Hassner, Sijia Liu, and Xiaoming Liu. Proactive image manipulation detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15386–15395, June 2022.

[3] Xiuli Bi, Zhipeng Zhang, and Bin Xiao. Reality transform adversarial generators for image splicing forgery detection and localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14294–14303, October 2021.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[5] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11315–11325, June 2022.

[6] 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.

[7] 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.

[8] 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*, pages 1–14, 2022.

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

[10] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ROC Analysis in Pattern Recognition.

[11] Jing Hao, Zhixin Zhang, Shicai Yang, Di Xie, and Shiliang Pu. Transforensics: Image forgery localization with dense self-attention. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15035–15044, 2021.

[12] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. Span: Spatial pyramid attention network for image manipulation localization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 312–328. Springer, 2020.

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

[14] Ashraful Islam, Chengjiang Long, Arslan Basharat, and Anthony Hoogs. Doa-gan: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[15] Xiao Jin, Yuting Su, Liang Zou, Yongwei Wang, Peiguang Jing, and Z. Jane Wang. Sparsity-based image inpainting detection via canonical correlation analysis with low-rank constraints. *IEEE Access*, 6:49967–49978, 2018.

[16] 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.

[17] 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*, 130(8):1875–1895, may 2022.

[18] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7880–7889, 2020.

[19] Haodong Li and Jiwu Huang. Localization of deep inpainting using high-pass fully convolutional network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[20] Haodong Li and Jiwu Huang. Localization of deep inpainting using high-pass fully convolutional network. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 8301–8310, October 2019.

[21] Haodong Li and Jiwu Huang. Localization of deep inpainting using high-pass fully convolutional network. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8300–8309, 2019.

[22] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10758–10768, June 2022.

[23] Xiaoguang Li, Qing Guo, Di Lin, Ping Li, Wei Feng, and Song Wang. Misf: Multi-level interactive siamese filtering for high-fidelity image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1869–1878, June 2022.

[24] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. *CVPR*, 2023.

[25] 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 David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[26] Xun Lin, Shuai Wang, Jiahao Deng, Ying Fu, Xiao Bai, Xinlei Chen, Xiaolei Qu, and Wenzhong Tang. Image manipulation detection by multiple tampering traces and edge artifact enhancement. *Pattern Recognition*, 133:109026, 2023.

[27] 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, 2022.

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[29] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11461–11471, June 2022.

[30] Gaël MAHFOUDI, Badr TAJINI, Florent RETRAINT, Frédéric MORAIN-NICOLIER, Jean Luc DUGELAY, and Marc PIC. Defacto: Image and face manipulation dataset. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5, 2019.

[31] Ghazal Mazaheri, Niluthpol Chowdhury Mithun, Jawadul H. Bappy, and Amit K. Roy-Chowdhury. A skip connection architecture for localization of image manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[32] Alex 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 (ICML)*, 2022.

[33] Fahim Faisal Niloy, Kishor Kumar Bhaumik, and Simon S. Woo. Cfl-net: Image forgery localization using contrastive learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4642–4651, January 2023.

[34] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2085–2094, October 2021.

[35] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[36] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2364–2373, June 2022.

[37] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2364–2373, June 2022.

[38] Sheng-Yu Wang, Oliver Wang, Andrew Owens, Richard Zhang, and Alexei A Efros. Detecting photoshopped faces by scripting photoshop. In *ICCV*, 2019.

[39] Wikipedia. Receiver operating characteristic. `https://en.wikipedia.org/wiki/Receiver_operating_characteristic`.

[40] Haiwei Wu and Jiantao Zhou. Iid-net: Image inpainting detection network via neural architecture search and attention. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3):1172–1185, 2022.

[41] Haiwei Wu and Jiantao Zhou. Iid-net: Image inpainting detection network via neural architecture search and attention. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3):1172–1185, 2022.

[42] 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.

[43] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*. Springer, 2018.

[44] Chao Yang, Huizhou Li, Fangting Lin, Bin Jiang, and Hao Zhao. Constrained r-cnn: A general image manipulation detection model. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2020.

[45] Yulan Zhang, Feng Ding, Sam Kwong, and Guopu Zhu. Feature pyramid network for diffusion-based image inpainting detection. *Inf. Sci.*, 572(C):29–42, sep 2021.

[46] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning rich features for image manipulation detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1053–1061, 2018.

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