

TGIF: Text-Guided Inpainting Forgery Dataset

Hannes Mareen*, Dimitrios Karageorgiou†, Glenn Van Wallendael*, Peter Lambert*, Symeon Papadopoulos†

* IDLab, Ghent University – imec, Ghent, Belgium, e-mail: firstname.lastname@ugent.be

† Information Technologies Institute, CERTH, Thessaloniki, Greece, e-mail: dkarageo@iti.gr, papadop@iti.gr

Abstract—Digital image manipulation has become increasingly accessible and realistic with the advent of generative AI technologies. Recent developments allow for text-guided inpainting, making sophisticated image edits possible with minimal effort. This poses new challenges for digital media forensics. For example, diffusion model-based approaches could either splice the inpainted region into the original image, or regenerate the entire image. In the latter case, traditional image forgery localization (IFL) methods typically fail. This paper introduces the Text-Guided Inpainting Forgery (TGIF) dataset, a comprehensive collection of images designed to support the training and evaluation of image forgery localization and synthetic image detection (SID) methods. The TGIF dataset includes approximately 75k forged images, originating from popular open-source and commercial methods, namely SD2, SDXL, and Adobe Firefly. We benchmark several state-of-the-art IFL and SID methods on TGIF. Whereas traditional IFL methods can detect spliced images, they fail to detect regenerated inpainted images. Moreover, traditional SID may detect the regenerated inpainted images to be fake, but cannot localize the inpainted area. Finally, both IFL and SID methods fail when exposed to stronger compression, while they are less robust to modern compression algorithms, such as WEBP. In conclusion, this work demonstrates the inefficiency of state-of-the-art detectors on local manipulations performed by modern generative approaches, and aspires to help with the development of more capable IFL and SID methods. The dataset and code can be downloaded at <https://github.com/IDLabMedia/tgif-dataset>.

Index Terms—Image Forensics, Forgery Detection, Forgery Localization, Synthetic Image Detection.

I. INTRODUCTION

Image manipulation is becoming more accessible with recent image editing tools based on generative AI (GenAI) achieving outstanding levels of photorealism [1], [2]. The advent of open-source and commercial text-guided approaches, such as Stable Diffusion (SD) [3] and Adobe Firefly, introduced the ability to locally manipulate an image using simple text prompts, even to non-experts. However, as high-quality image manipulation is becoming more accessible, there are great risks to be exploited for malicious purposes such as disinformation, fraud, and fake evidence. Therefore, image forensic methods aim to detect and localize manipulations [4], [5]. While inpainting has long been part of the target manipulation types of state-of-the-art image forgery localization (IFL) methods, it is now revived due to the proliferation of text-to-image generative diffusion models [2]. Previously, inpainting

meant the removal of objects [6], [7], whereas text-guided inpainting models enable the addition or adaptation of objects.

Text-guided inpainting using diffusion-based GenAI models works differently than traditional inpainting: an image (crop) is given as input to a model, along with a mask of the inpainted area and a textual prompt. Through a series of diffusion and denoising steps, the model regenerates the entire image, yet only visibly changing the inpainted area [3]. There are two ways of using text-guided inpainting. First, by splicing the inpainted area into the original image, which is most useful when the generative model does not support high resolutions. Second, the fully regenerated image can be used as is. With new models supporting larger resolutions, the latter case is becoming more practical. However, localizing the inpainted region when most forensic cues have been corrupted due to the regeneration of the whole image is a new and hard challenge for IFL algorithms, and untouched by synthetic image detection (SID) methods.

Training and evaluating forensic methods requires large and varied datasets. Although several image manipulation datasets exist, text-guided inpainting is typically not included in them. In fact, to the best of our knowledge, the only dataset including text-guided inpainting is COCO-Glide [8], consisting of 512 manipulated images of 256×256 px, created using a single method, i.e. GLIDE [9]. The dataset lacks extensive metadata, and does not vary in terms of splicing type and mask type. Hence, there is a need for larger text-guided inpainting datasets that can be utilized for both training and evaluation, consisting of high-resolution images.

This paper presents the Text-Guided Inpainting Forgery (TGIF) dataset. The contribution of this work is two-fold. First, we introduce our TGIF dataset in Section III, which can be used for both training and evaluation. The dataset consists of approx. 75k fake images, with resolutions up to 1024×1024 px that we generated using three inpainting models (SD2, SDXL & Adobe Firefly). Additionally, we provide both the manipulated image where the inpainted area is spliced in the original image, as well as the fully-regenerated image, when possible. We provide metadata about the estimated aesthetic quality, as well as image-text matching scores that indicate whether the generated image aligns with the given prompt.

Second, we evaluate our TGIF test set on state-of-the-art IFL and SID methods, in Section IV. Additionally, we study the effects of JPEG and WEBP compression on detection performance. In this way, we provide valuable insights into the text-guided inpainting manipulations, how to detect them in practice, and highlight open challenges.

This work was funded by the Flemish government’s Department of Culture, Youth & Media (under the COM-PRESS project), by IDLab (Ghent University – imec), by Flanders Innovation & Entrepreneurship (VLAIO), by Research Foundation – Flanders (FWO) (V419524N), and by the European Union under the Horizon Europe vera.ai project (grant number 101070093).

II. RELATED WORK

A. Image Forgery Localization (IFL)

To detect local image manipulations, in which a region of an image was forged whereas the rest remains pristine, a variety of methods have been proposed [4]. In general, IFL methods either aim to detect imperceptible inconsistencies in the image, such as the artifacts originating from compression [10] or the noise pattern introduced by camera sensors [11], or they aim to detect visible ones, such as boundaries between the forged and pristine regions [12], while others fuse several forensic cues for increased robustness [13]–[15]. Furthermore, there exist a few IFL methods that specifically target inpainting [16]–[18], which exploit specific changes made by inpainting models, such as in image Laplacian [16], or which automatically learn to extract relevant features [17], [18]. However, they have mostly been superseded by more recent general works [8], [19].

Recent well-performing methods [8], [12], [19]–[23] are mostly based on deep learning, i.e., they train and evaluate on datasets containing diverse manipulation types, such as splicing, copy-move, and “traditional” inpainting/removal, in order to detect a broad range of alterations. However, such methods have not been trained to detect fully regenerated images using text-guided generative models.

B. Synthetic Image Detection (SID)

Early popular synthetic image generation approaches were mostly based on Generative Adversarial Networks (GANs), whereas newer ones typically rely on diffusion models (DMs). Accordingly, several detection methods were trained on images generated by GANs [24]–[30], whereas newer ones typically focus on DMs or a mix of both [31]–[34].

In general, most methods aim to extract features or a fingerprint, and somehow compare them to those typically observed in synthetic images. The type of feature may have a large influence on the model’s generalization performance. For example, some methods were trained on GANs, yet demonstrated the ability to detect images generated by DMs [26], [29], [30]. However, state-of-the-art SID methods were trained and evaluated on real and synthetic images, lacking support for images that were inpainted and hence contain both fake and real parts (albeit regenerated by DMs). In other words, they lack the ability of localizing regions inpainted by DMs.

C. Image Forensics Datasets

A variety of datasets for IFL and SID exist, and have been listed in recent overview papers [35], [36]. Most relevant to this paper, inpainted images can be found in the NIST16 [6] and DEFACTO [7] datasets. However, the inpainting in them is not guided by a textual prompt, but instead aims to simply remove the selected object through splicing.

To the best of our knowledge, the only dataset that includes text-guided inpainting is COCO-Glide [8], which consists of 512 manipulated images with a resolution of 256×256 px, cropped from the MS-COCO dataset and inpainted using GLIDE [9]. This dataset is small and mainly useful

for evaluation. Additionally, the relatively low resolution is already outdated, with new models supporting resolutions up to 1024×1024 px. Moreover, it neither considers the differences between splicing the generated content into the inpainted region of the original image and re-generating the entire picture with the GenAI model, nor does it assess the perceptual quality of the inpainted images. Thus, the forensics community lacks a general large dataset that considers local image manipulations performed by generative models, while including localization masks for images that have been fully regenerated.

III. TGIF DATASET

This section presents our text-guided inpainting forgery dataset, containing both spliced and fully regenerated images; yet only visually different in the inpainted area.

A. Source of Authentic Data

We used the MS-COCO dataset [37] (*val2017*) to collect authentic images licensed under a Creative Commons Attribution 4.0 License, along with captions and object masks (segmentation mask and bounding box). The images contain objects from 80 categories (such as types of animals, food, and furniture). Of those, we excluded *knife* and *frisbee* because these prompts were not allowed by Adobe Firefly.

We edited the provided Flickr URLs in order to collect images with dimensionality of up to 1024 pixels (the largest publicly available resolution) [38], as the original COCO metadata linked to Flickr images with a size limited to 640 pixels on their largest dimension.

From each of the object classes, we selected a maximum of 50 images. We excluded images in cases where the 1024p Flickr URL was not accessible, as well as those with one of their dimensions smaller than 512 pixels. Moreover, the object bounding box resolution should be smaller than 512×512 px, and the product of width and height should be larger than 64^2 .

B. Text-Guided Inpainting Generation

For each authentic image and depicted object, we generated inpainted variants, where the authentic object was replaced by a generated one of the same class. For example, a real cat was replaced by a generated cat.

We utilized three text-guided inpainting methods: a) the open-source SD2 (*stabilityai/stable-diffusion-2-inpainting*), which constitutes a first-generation high-fidelity diffusion model, b) the open-source SDXL (*diffusers/stable-diffusion-xl-1.0-inpainting-0.1*), representing the second generation and supporting images of higher resolution, and c) the commercial Adobe Firefly (in Adobe Photoshop 25.4.0). The open-source models are encountered in popular inpainting pipelines, such as Inpaint-Anything [39], while the commercial approaches are more accessible to non-experts. These three generative approaches constitute a representative set of the available approaches for local manipulation through generative models.

For SD2 and SDXL, we cropped images (centered around the object of interest) to 512×512 px and 1024p, respectively,

as these are their respective native resolutions. Additionally, the dimensions needed to be a multiple of 8. In Adobe Firefly, we did not need to take any crops, although this may happen internally. To perform the inpainting, a mask of the object is given to select the area to be inpainted. For this, we utilized the segmentation mask and bounding box provided by COCO. Moreover, each inpainting operation of each generative model generated was repeated three times in a batch, resulting in three output variations (set with the *num_images_per_prompt* variable, which internally uses a different seed for each variation), which we all included in the dataset. I.e., each image was inpainted using two masks, and each of those inpaintings generated three variations. This process was repeated for each of the three inpainting methods.

As a textual prompt in SD2 and SDXL, we utilized the object category followed by a caption from the image (i.e., describing the full image). During initial experiments, we noticed that including the caption resulted in better results, which was also observed in related work [40], [41]. However, text misalignment between the prompt and the inpainted area could still occur. For Adobe Firefly, we only included the object category as prompt (i.e., only describing the inpainted area), as we noticed that this was sufficient in our initial experiments.

We randomly chose several other parameters. For each image, the number of inference steps is a random integer between 10 and 50, and the guidance scale is a random float between 1 and 10. The seed is randomly chosen. These parameters are saved as metadata.

When inpainting using GenAI, the entire input image is regenerated. Although only the masked area has visible changes, also the non-masked area is (imperceptibly) changed by the GenAI algorithm, since the entire image undergoes diffusion. In Adobe Firefly (Photoshop), the GenAI inpainted image is spliced back into the original image, using the provided mask with an additional small gradient border (i.e., a few pixels larger). This is a similar approach as provided by other inpainting pipelines, such as the one provided in HuggingFace diffusers [42]. For consistency, we spliced the SD2 inpainted image into the original image using the same adapted mask from Firefly, with an additional gradient border. Additionally, we saved the fully regenerated output image as a 512×512 px image. For SDXL, we only saved the fully regenerated output image as a 1024p image. That is because the SDXL inpainting model discolors the entire image, which is extremely visible when splicing into the original (a known issue [43]).

In summary, an authentic image is transformed into $2 \times 3 \times 4 = 24$ manipulated variants. That is, *two* masks are used for its generation (segmentation and bounding box), and each generation creates *three* variations in batch (*num_images_per_prompt*). Using the SD2, SDXL, and PS methods, we saved either the spliced or fully regenerated image, or both. Namely, for SD2, we saved both the spliced and fully regenerated (synthetic) image crop. For SDXL, we saved only the fully regenerated (synthetic) image. For Adobe Firefly (PS), we saved only the spliced image. As a result,

we get *four* sub-datasets SD2-sp, PS-sp, SD2-fr, and SDXL-fr (*sp* for splicing and *fr* for fully regenerated). Fig. 1 shows a high-level diagram of this procedure.

The dataset and code can be downloaded at <https://github.com/IDLabMedia/tgif-dataset>.

C. Dataset Splits

We have separated the full dataset in a training, validation, and test set, in an approximate 80/10/10 setup, based on the authentic images. These sets contain 58,560, 8,184, and 8,232 manipulated images (74,976 total), originating from 2,440, 341, and 343 unique authentic images (3,124 total), respectively. As described in Section III-B, each authentic image is transformed into $2 \times 3 \times 4 = 24$ manipulated images.

We have taken precaution to avoid leakage. First, we split the dataset based on the object category. That is, the same category does not appear in multiple subsets. Second, we deleted images that would appear in multiple subsets. That is because an image can contain multiple objects, and hence appear multiple times in the dataset (under different categories, with different masks and prompts).

D. Aesthetic Quality and Image-Text-Matching Scores

Due to the automatic generation of a large dataset, the quality of the inpainted objects varies significantly. Additionally, the prompt is not always respected by the diffusion model. To objectively measure this, we calculated aesthetic quality scores using the neural image assessment (NIMA) [44] (*epoch-82 & vgg16-397923af*) and generated image quality assessment (GIQA) [45] (*pca95-cat & gmm-cat-pca95-full7*) metrics. Additionally, we calculated image-text-matching (ITM) scores [46] (*Salesforce_blip-itm-base-coco*) to measure whether the prompt matches the outcome. We measured these scores on only the inpainted area (using the provided bounding box), as opposed to on the entire image. We include these values as metadata in the dataset.

IV. BENCHMARK: IFL & SID

A. Experimental Setup

We performed the evaluation on the test set, separating the spliced subsets (SD2-sp and PS-sp) and fully regenerated subsets (SD2-fr and SDXL-fr). Additionally, to make the task more challenging, we evaluated the performance on compressed variants of the real and fake images. That is, we used the JPEG and WEBP compression standards, using quality factors of 80 and 60.

For IFL, we use the mean pixel-level F1 score of the fake images as the performance metric, which is the harmonic mean of the precision and recall. We utilize a threshold of 0.5 to decide whether pixels are real or fake. This threshold assesses the calibration of the models, which is essential when deploying the models in the wild and requiring human interpretation. We selected several recent high-performing AI-based IFL methods, namely the PSCC-Net [19], SPAN [20], ImageForensicsOSN [21], MVSS-Net++ [12], Mantranet [22], CAT-Net (v2) [23], TruFor [8], and MMFusion [13].

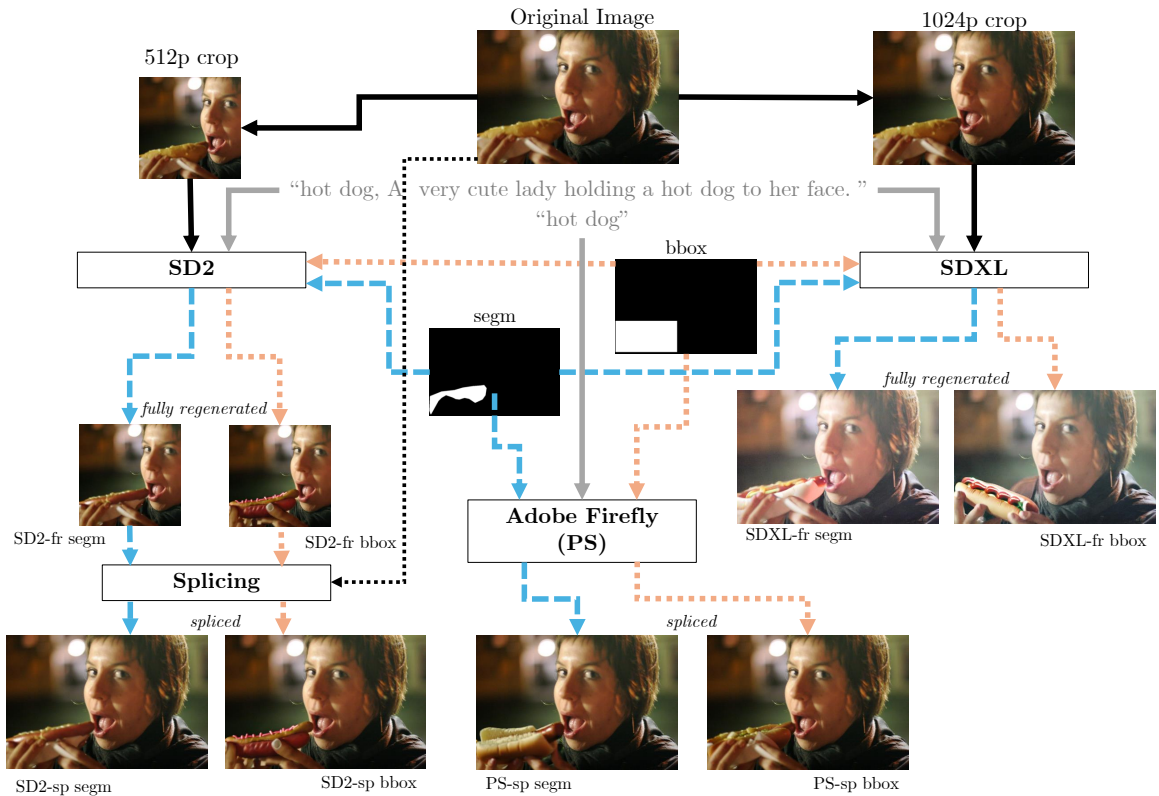


Fig. 1. TGIF dataset creation. The original image is fed to SD2, SDXL, and Adobe Firefly (Photoshop), along with a prompt and a mask (segmentation or bounding box). For SD2, an additional splicing step is performed, which is done automatically in Adobe Firefly, and discarded for SDXL’s output. For simplicity, only one variation is illustrated, rather than all three variations that we generated in batch.

For SID, we use the AUC score as the performance metric. We opted to report the threshold-agnostic AUC rather than threshold-specific image-level F1 score, because several SID methods reported high F1 scores while actually demonstrating bad performance. More specifically, these methods often classified both real and fake images as fake, which, due to the class imbalance, significantly boosted the F1 score. Since the AUC represents the trade-off between the false positive and false negative rate for a range of thresholds, it better reflects detection performance, in this case. We selected several AI-based SID methods available in the SID-Bench framework [47], namely CNDetect [24], DIMD [31], Dire [32], FreqDetect [25], UnivFD [26], Fusing [48], Gram-Net [27], LGrad [28], NPR [29], RINE [33], DeFake [34], and PatchCraft [30]. Some methods provide multiple models, trained on different data. We only show the best performing in this paper. Although some of these methods were trained on GANs, some of them have demonstrated generalization performance to DMs. Hence, also methods trained on GANs were included in our evaluation set.

B. Image Forgery Localization Methods

Table I reports the F1 values for all evaluated IFL methods. We highlight F1 values above 0.7. This threshold was chosen arbitrarily, but allows to quickly grasp which methods demonstrate good detection performance. For the splicing datasets SD2 and PS, we observe that only CAT-Net, TruFor, and MMFusion report high F1 scores. Additionally, Mantranet

only reports a mediocre F1 value for the PS dataset, while it has low performance for SD2. All other methods demonstrate low detection performance.

For the fully regenerated datasets SD2-fr and SDXL-fr, no method is able to detect and localize these manipulations. This is expected, since regenerating the image removes most invisible traces such as compression artifacts and camera noise. In fact, diffusing and regenerating an image is an existing attack against forgery detection methods [49]. This highlights the threat of local image manipulations using diffusion models that fully regenerate images; these cannot be detected by traditional image forgery localization methods. This threat is enlarged when one considers the widespread use of generative models in applications such as super-resolution [50] and image compression [51]. Although these applications can be used with harmless intentions, they may remove traces used by traditional IFL methods.

We only analyzed the influence of compression for the best performing methods in the uncompressed case, and only for the splicing datasets. When performing JPEG compression, F1 scores drop significantly. Moreover, this performance drop is amplified when performing WEBP compression. This demonstrates a need to develop IFL methods that are robust to compression.

C. Synthetic Image Detection Methods

Table II shows the performance (AUC) results for all evaluated SID methods. We highlight AUC values above 0.8. This

TABLE I
EVALUATION OF IMAGE FORGERY LOCALIZATION METHODS ON OUR TGIF DATASET (F1 SCORE). WE ONLY EVALUATED COMPRESSED IMAGES ON IFL METHODS AND SUB-DATASETS EXHIBITING DECENT PERFORMANCE ON UNCOMPRESSED IMAGES.

Method	Uncompressed				JPEG Q80		JPEG Q60		WEBP Q80		WEBP Q60	
	SD2-sp	PS-sp	SD2-fr	SDXL-fr	SD2-sp	PS-sp	SD2-sp	PS-sp	SD2-sp	PS-sp	SD2-sp	PS-sp
PSCC-Net [19]	0.15	0.38	0.05	0.05	-	-	-	-	-	-	-	-
SPAN [20]	0.00	0.00	0.00	0.00	-	-	-	-	-	-	-	-
ImageForensicsOSN [21]	0.23	0.36	0.20	0.18	-	-	-	-	-	-	-	-
MVSS-Net++ [12]	0.07	0.08	0.06	0.09	-	-	-	-	-	-	-	-
Mantranet [22]	0.15	0.56	0.03	0.05	0.08	0.13	0.03	0.04	0.08	0.11	0.09	0.11
CAT-Net [23]	0.87	0.85	0.04	0.03	0.05	0.07	0.09	0.08	0.13	0.13	0.07	0.07
TruFor [8]	0.83	0.79	0.19	0.18	0.32	0.44	0.25	0.32	0.24	0.29	0.20	0.23
MMFusion [13]	0.75	0.74	0.15	0.18	0.35	0.49	0.22	0.32	0.22	0.33	0.19	0.28

TABLE II
EVALUATION OF SYNTHETIC IMAGE DETECTION METHODS ON OUR TGIF DATASET (AUC SCORE).

Method	Trained on	Uncompressed		JPEG Q80		JPEG Q60		WEBP Q80		WEBP Q60	
		SD2-fr	SDXL-fr	SD2-fr	SDXL-fr	SD2-fr	SDXL-fr	SD2-fr	SDXL-fr	SD2-fr	SDXL-fr
CNNDetect [24]	ProGAN	0.57	0.61	0.57	0.62	0.56	0.61	0.55	0.60	0.55	0.60
DIMD [31]	LDM	1.00	0.94	1.00	0.95	0.99	0.92	0.96	0.71	0.90	0.61
Dire [32]	ADM	0.49	0.62	0.49	0.61	0.49	0.60	0.50	0.60	0.48	0.60
FreqDetect [25]	GANs	0.50	0.40	0.72	0.71	0.71	0.72	0.66	0.61	0.62	0.62
UnivFD [26]	ProGAN	0.82	0.8	0.73	0.78	0.70	0.73	0.71	0.74	0.69	0.73
Fusing [48]	ProGAN	0.55	0.47	0.52	0.59	0.53	0.61	0.50	0.53	0.50	0.54
GramNet [27]	GANs	0.82	0.55	0.48	0.56	0.47	0.54	0.49	0.54	0.50	0.53
LGrad [28]	ProGAN	0.85	0.83	0.51	0.57	0.51	0.64	0.47	0.42	0.47	0.41
NPR [29]	ProGAN	0.51	0.67	0.50	0.30	0.51	0.68	0.49	0.69	0.49	0.31
RINE [33]	ProGAN	0.89	0.89	0.79	0.88	0.75	0.85	0.77	0.85	0.76	0.87
	LDM	0.97	0.93	0.84	0.89	0.80	0.82	0.75	0.76	0.72	0.77
DeFake [34]	Diffusion	0.65	0.61	0.54	0.45	0.54	0.43	0.54	0.45	0.53	0.49
PatchCraft [30]	ProGAN	0.98	0.95	0.66	0.67	0.60	0.58	0.60	0.57	0.59	0.61

threshold was chosen arbitrarily, but allows to quickly grasp which methods demonstrate decent detection performance. Note that we do not include the performance results for the spliced datasets (SD2-sp and PS-sp), since we cannot expect SID to detect these types of local manipulation, as they were not designed for this purpose. We have verified that, indeed, no SID method is able to detect the synthetic spliced regions.

We observe that 6 out of 12 methods are able to detect (fully regenerated) synthetic images in uncompressed form. Surprisingly, some methods trained on GANs (such as UnivFD, LGrad and RINE) demonstrate better performance than methods trained on DMs (such as DeFake). Additionally, only two of those methods survive JPEG compression, and only one survives WEBP compression. Moreover, some methods (such as DIMD) are more robust in detecting SD2 than SDXL, whereas others (such as RINE) are more robust in detecting SDXL than SD2. No single method is robust for both SD2 and SDXL on all evaluated compression levels. This demonstrates that these methods are complementary.

In summary, we demonstrated that several state-of-the-art SID methods are able to detect fully regenerated inpainted images by SD2 and SDXL. However, they are not able to localize the inpainted area, but rather produce a SID score for the whole image. This is expected as they were not designed to detect the spliced inpainted images (even though the spliced regions were generated synthetically). Finally, their performance decreases when exposed to compression, with a notable difference between WEBP and JPEG compression.

V. DISCUSSION & CONCLUSION

The proposed TGIF dataset represents a significant advancement in the resources available for training and evaluating forensic methods. By including high-resolution images and utilizing multiple inpainting methods, as well as providing both spliced and fully regenerated versions, our TGIF dataset addresses the limitations of existing forensic methods and datasets. As such, it is expected to contribute to the development of more effective forensic tools.

Our benchmark analysis shows that some of the existing IFL methods are able to detect and localize spliced images, whereas they fail to localize the inpainted area in fully regenerated images. In contrast, some of the existing SID methods are able to detect fully regenerated images, yet lack the ability to localize the synthetic inpainted area. These limitations highlight the need for new forensic methods, leveraging elements from both IFL and SID methods. Furthermore, we showed that forensic methods suffer from a notable decrease in performance when exposed to compression, particularly with WEBP. This highlights the need to include compressed images during training.

Future work could extend and incorporate the TGIF dataset for the training of new detection methods to localize manipulations in fully regenerated images, with attention to robustness against compression.

REFERENCES

- [1] C. Zhang, C. Zhang, M. Zhang, and I. S. Kweon, "Text-to-image diffusion models in generative AI: A survey," *arXiv preprint arXiv:2303.07909*, 2023.

- [2] F. Zhan *et al.*, “Multimodal image synthesis and editing: The generative AI era,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 45, no. 12, pp. 15 098–15 119, 2023.
- [3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proc. IEEE/CVF Conf. Computer Vision Pattern Recogn.*, 2022.
- [4] F. Z. Mehrjardi, A. M. Latif, M. S. Zarchi, and R. Sheikhpour, “A survey on deep learning-based image forgery detection,” *Pattern Recognition*, p. 109778, 2023.
- [5] D. Tariang, R. Corvi, D. Cozzolino, G. Poggi, K. Nagano, and L. Verdoliva, “Synthetic image verification in the era of generative ai: What works and what isn’t there yet,” *arXiv preprint arXiv:2405.00196*, 2024.
- [6] H. Guan *et al.*, “MFC datasets: Large-scale benchmark datasets for media forensic challenge evaluation,” in *Proc. IEEE Winter App. Computer Vision Workshops (WACVW)*. IEEE, 2019, pp. 63–72.
- [7] G. Mahfoudi, B. Tajini, F. Reirant, F. Morain-Nicolier, J. L. Dugelay, and P. Marc, “Defacto: Image and face manipulation dataset,” in *Europ. Signal Process. Conf. (EUSIPCO)*. IEEE, 2019, pp. 1–5.
- [8] F. Guillaro *et al.*, “TruFor: Leveraging all-round clues for trustworthy image forgery detection and localization,” in *Proc. IEEE/CVF Conf. Computer Vision Pattern Recogn. (CVPR)*, 2023, pp. 20 606–20 615.
- [9] A. Nichol *et al.*, “GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models,” in *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022, pp. 16 784–16 804.
- [10] H. Mareen *et al.*, “Comprint: Image forgery detection and localization using compression fingerprints,” in *Proc. Int. Conf. Pattern Recogn. Workshops*, 2023, pp. 281–299.
- [11] D. Cozzolino and L. Verdoliva, “Noiseprint: A CNN-based camera model fingerprint,” *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 144–159, 2020.
- [12] C. Dong, X. Chen, R. Hu, J. Cao, and X. Li, “MVSS-Net: Multi-view multi-scale supervised networks for image manipulation detection,” *IEEE Trans. Pattern Analysis and Machine Intel.*, vol. 45, no. 3, pp. 3539–3553, 2022.
- [13] K. Triaridis and V. Mezaris, “Exploring multi-modal fusion for image manipulation detection and localization,” in *Int. Conf. Multimedia Model*. Springer, 2024, pp. 198–211.
- [14] D. Karageorgiou, G. Kordopatis-Zilos, and S. Papadopoulos, “Fusion transformer with object mask guidance for image forgery analysis,” in *Proc. IEEE/CVF Conf. Computer Vision Pattern Recogn.*, 2024, pp. 4345–4355.
- [15] H. Mareen, L. De Neve, P. Lambert, and G. Van Wallendaal, “Harmonizing image forgery detection & localization: Fusion of complementary approaches,” *J. Imaging*, vol. 10, no. 1, p. 4, 2023.
- [16] H. Li, W. Luo, and J. Huang, “Localization of diffusion-based inpainting in digital images,” *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 12, pp. 3050–3064, 2017.
- [17] H. Wu and J. Zhou, “Iid-net: Image inpainting detection network via neural architecture search and attention,” *IEEE Trans. Circuits Systems Video Technol.*, vol. 32, no. 3, pp. 1172–1185, 2021.
- [18] A. Li, Q. Ke, X. Ma, H. Weng, Z. Zong, F. Xue, and R. Zhang, “Noise doesn’t lie: Towards universal detection of deep inpainting,” in *Proc. Int. Conf. Artificial Intel. IJCAI, Z.-H. Zhou, Ed.*, 2021, pp. 786–792.
- [19] X. Liu *et al.*, “PSCC-Net: Progressive spatio-channel correlation network for image manipulation detection and localization,” *IEEE Trans. Circuits Systems Video Technol.*, vol. 32, no. 11, pp. 7505–7517, 2022.
- [20] X. Hu *et al.*, “SPAN: Spatial pyramid attention network for image manipulation localization,” in *Proc. Europ. Computer Vision Conf*. Springer, 2020, pp. 312–328.
- [21] H. Wu, J. Zhou, J. Tian, J. Liu, and Y. Qiao, “Robust image forgery detection against transmission over online social networks,” *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 443–456, 2022.
- [22] Y. Wu, W. AbdAlmageed, and P. Natarajan, “ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features,” in *Proc. IEEE/CVF Conf. Computer Vision Pattern Recogn.*, 2019, pp. 9543–9552.
- [23] M.-J. Kwon *et al.*, “Learning JPEG compression artifacts for image manipulation detection and localization,” *Int. J. Computer Vision*, vol. 130, no. 8, pp. 1875–1895, 2022.
- [24] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, “Cnn-generated images are surprisingly easy to spot...for now,” in *Proc. Int. Conf. Pattern Recogn. (CVPR)*, 2020.
- [25] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, “Leveraging frequency analysis for deep fake image recognition,” in *Int. Conf. Machine Learning*. PMLR, 2020, pp. 3247–3258.
- [26] U. Ojha, Y. Li, and Y. J. Lee, “Towards universal fake image detectors that generalize across generative models,” in *Proc. IEEE/CVF Conf. Computer Vision Pattern Recogn.*, 2023, pp. 24 480–24 489.
- [27] Z. Liu, X. Qi, and P. H. Torr, “Global texture enhancement for fake face detection in the wild,” in *Proc. IEEE/CVF Conf. Computer Vision Pattern Recogn.*, 2020, pp. 8060–8069.
- [28] C. Tan *et al.*, “Learning on gradients: Generalized artifacts representation for gan-generated images detection,” in *Proc. IEEE/CVF Conf. Computer Vision Pattern Recogn.*, 2023, pp. 12 105–12 114.
- [29] C. Tan, Y. Zhao, S. Wei, G. Gu, P. Liu, and Y. Wei, “Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection,” in *Proc. IEEE/CVF Conf. Computer Vision Pattern Recogn.*, 2024, pp. 28 130–28 139.
- [30] N. Zhong, Y. Xu, Z. Qian, and X. Zhang, “PatchCraft: Exploring texture patch for efficient ai-generated image detection,” *arXiv preprint arXiv:2311.12397*, 2023.
- [31] R. Corvi *et al.*, “On the detection of synthetic images generated by diffusion models,” in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.
- [32] Z. Wang *et al.*, “DIRE for diffusion-generated image detection,” in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2023, pp. 22 445–22 455.
- [33] C. Koutlis and S. Papadopoulos, “Leveraging representations from intermediate encoder-blocks for synthetic image detection,” in *Proc. Europ. Computer Vision Conf. (ECCV)*, 2024.
- [34] Z. Sha, Z. Li, N. Yu, and Y. Zhang, “DE-FAKE: Detection and attribution of fake images generated by text-to-image generation models,” in *Proc. ACM SIGSAC Conf. Computer Comm. Security*, 2023, pp. 3418–3432.
- [35] L. Verdoliva, “Media forensics and deepfakes: an overview,” *IEEE J Selected Topics Signal Process.*, vol. 14, no. 5, pp. 910–932, 2020.
- [36] L. Lin *et al.*, “Detecting multimedia generated by large AI models: A survey,” *arXiv preprint arXiv:2402.00045*, 2024.
- [37] T.-Y. Lin *et al.*, “Microsoft COCO: Common objects in context,” in *Proc. Europ. Computer Vision Conf*. Springer, 2014, pp. 740–755.
- [38] “Flickr services photo image urls,” <https://www.flickr.com/services/api/misc.urls.html>, accessed: 2024-07-05.
- [39] T. Yu *et al.*, “Inpaint anything: Segment anything meets image inpainting,” *arXiv preprint arXiv:2304.06790*, 2023.
- [40] S. Xie *et al.*, “SmartBrush: Text and shape guided object inpainting with diffusion model,” in *Proc. IEEE/CVF Conf. Computer Vision Pattern Recogn.*, 2023, pp. 22 428–22 437.
- [41] S. Wang *et al.*, “Imagen editor and EditBench: Advancing and evaluating text-guided image inpainting,” in *Proc. IEEE/CVF Conf. Computer Vision Pattern Recogn.*, 2023, pp. 18 359–18 369.
- [42] “Diffusers: State-of-the-art diffusion models for image and audio generation in pytorch and flax.” <https://github.com/huggingface/diffusers>, accessed: 2024-07-05.
- [43] “Diffusers github issue - inpainting produces results that are uneven with input image,” <https://github.com/huggingface/diffusers/issues/5808>, accessed: 2024-07-05.
- [44] H. Talebi and P. Milanfar, “NIMA: Neural image assessment,” *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3998–4011, 2018.
- [45] S. Gu, J. Bao, D. Chen, and F. Wen, “GIQA: Generated image quality assessment,” in *Proc. Europ. Computer Vision Conf. (ECCV)*. Springer, 2020, pp. 369–385.
- [46] J. Li, D. Li, C. Xiong, and S. Hoi, “BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation,” in *Int. Conf. Machine Learning*. PMLR, 2022, pp. 12 888–12 900.
- [47] M. Schinas and S. Papadopoulos, “SIDBench: A python framework for reliably assessing synthetic image detection methods,” in *MAD ’24: Proc. ACM Int. W. Multimedia AI ag. Disinform.*, 2024, pp. 55–64.
- [48] Y. Ju, S. Jia, L. Ke, H. Xue, K. Nagano, and S. Lyu, “Fusing global and local features for generalized ai-synthesized image detection,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*. IEEE, 2022, pp. 3465–3469.
- [49] M. Tailanián, M. Gardella, A. Pardo, and P. Musé, “Diffusion models meet image counter-forensics,” in *Proc. IEEE/CVF Winter Conf. App. Computer Vision (WACV)*, 2024, pp. 3925–3935.
- [50] H. Li *et al.*, “SRDiff: Single image super-resolution with diffusion probabilistic models,” *Neurocomputing*, vol. 479, pp. 47–59, 2022.
- [51] R. Yang and S. Mandt, “Lossy image compression with conditional diffusion models,” in *Advances in Neural Information Processing Systems*, vol. 36. Curran Associates, Inc., 2023, pp. 64 971–64 995.