DinoLizer: Separating VAE and Diffusion Artifacts in Generative Inpainting Localization

DOI Minh Thong, Vincent ITIER
UMR 9189 CRIStAL, CERI SN
IMT Nord Europe
F-59000 Lille, France
minh-thong.doi@imt-nord-europe.fr
vincent.itier@imt-nord-europe.fr

Jan BUTORA, Jérémie BOULANGER, Patrick BAS
Univ. Lille, CNRS, Centrale Lille
UMR 9189 CRIStAL
F-59000 Lille, France
jeremie.boulanger@univ-lille.fr
{jan.butora,patrick.bas}@cnrs.fr

***Abstract*—We introduce DinoLizer, a DINOv2-based localizer of manipulated areas in generative inpainting. The model is trained to focus on semantically altered regions by treating reconstructed areas outside the inpainted mask as a separate class, which yields significant improvements w.r.t. the conventional approach. We train the model with LORA on the Query and Value of the transformer blocks and simply add 1 linear layer on top of the backbone to predict manipulations on a 14×14 patch resolution. Because DINOv2 only accepts fixed-sized images, we use a sliding window approach to aggregate the predictions on larger images. Empirical results show that DinoLizer outperforms state-of-the-art methods on our proposed dataset and SOTA inpainting datasets. Furthermore, it is very robust to JPEG (double) compression. On average, DinoLizer achieves a 20% higher Intersection over Union score compared to the second best model. The code is publicly available here: https://github.com/anonyme610/dinolizer.**

I. Introduction

Generative Artificial Intelligence (Gen-AI) poses a significant threat to democracy by facilitating the creation of highly realistic disinformation. In particular, generative inpainting allows for the easy local removal or insertion of subjects by seamlessly blending authentic image components with synthetic backgrounds or objects. These hybrid manipulations are notoriously difficult to detect because the high fidelity of modern generative models makes it nearly impossible for human observers to differentiate between authentic and fabricated content.

Many existing inpainting methods are based either on diffusion models such as Stable-Diffusion [1], Glide [2], or the commercial model from Adobe FireFly, but also other methods such as Fourier convolution-based methods and residual networks like LaMa [3] for object removal. On diffusion models, the inpainting methods such as BrushNet [4], PowerPaint [5], or HD-painter [6] come from models that are fine-tuned or adapted using extra input channels. They are used either to infer a masked part of the image during the backward diffusion process (for object insertion) or to replace an object with another. To obtain a better inpainting quality, the alignment between masked regions and semantic objects can be enforced by using BLIP embeddings [7]. InpaintAnything [8] also proposes to mix the inpainting process with segmentation tools to perform high-quality inpainting.

Identify applicable funding agency here. If none, delete this.

Image forgery localization (IFL) methods have received increased attention due to rapidly evolving sophisticated algorithms for local image tampering. TruFor [9] uses a Transformer-based fusion of high-level image information and a noise signature extracted by the Noiseprint++ feature extractor to predict local discrepancies. SAFIRE [10] segments an image into regions having the same properties, and localizes manipulations as outlier regions. ManTra-Net [11] uses a fully convolutional network without any subsampling to predict local anomalies, and CAT-Net v2 [12] is another CNN that focuses on distortion in JPEG compression artifacts due to forgeries. In [13], the authors aim to find non-semantic features useful for IFL, and modify SparseViT [14] to extract context-irrelevant features by splitting the image into lattices of non-neighboring patches. However, most current methods have only been evaluated on traditional forgeries, such as copy-move and splicing, and not many works deal with AI-based inpainting with, e.g., diffusion models.

Our methodology for training DinoLizer is characterized by two core contributions. First, we prioritize computational efficiency by employing Low-Rank Adaptation (LoRA) [15] to fine-tune the DINOv2 [16] backbone, which significantly reduces the number of trainable parameters from 86M to 2.4M while preserving high performance. Second, we distinguish our approach by introducing a dedicated class for VAE artifacts; this enables the model to learn a more nuanced representation of generative traces rather than simply grouping them with the "real" class, thereby improving the distinction between authentic and manipulated regions.

II. Related Work

A. Localization Methods Using Transformers

Vision Transformers (ViTs) utilize self-attention mechanisms to capture long-range dependencies, making them highly effective for tasks such as object segmentation. Following the approach proposed by Segmenter [17], a common strategy involves an encoder-decoder architecture where the transformer serves as the encoder, and the decoder consists of a linear mapping between the predicted masks and patch embeddings,

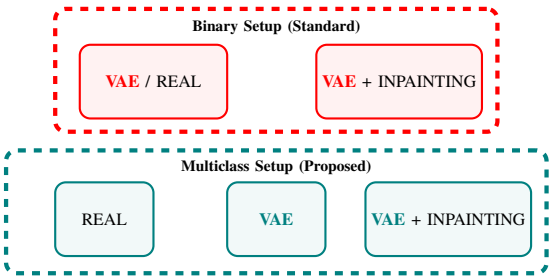


Fig. 1. In a binary setup, the learning mechanism can be confused by the VAE fingerprint, which is potentially present in both classes. On the contrary, in a multiclass setup, the VAE fingerprint is learned from one specific class, which eases the distinction with respect to inpainting.

followed by upsampling. This methodology was similarly adopted by the DINOv2 ViT [16] for image segmentation.

Regarding image forgery localization, SparseViT [13] and DINOv3-IML [18] are, to the best of our knowledge, the only works that utilize transformers for this purpose. SparseViT [13] is trained on disjoint subsampled subsets of patches to enforce sparse self-attention maps; this prevents the model from focusing solely on semantic content, as many generation traces (e.g., VAE artifacts) are independent of image semantics.

B. Semantic and VAE bias removal

Recently, different contributions have shown both excellent in-distribution and out-of-distribution detection (*i.e.* testing on unseen generators) performance by adopting training strategies designed to remove the semantic and/or VAE biases (artifacts introduced by the VAE encoder and decoder of generative models). The semantic bias usually comes from the fact that images are generated with prompts that are too short to represent a realistic description of a natural image. Another bias is due to the heterogeneity of VAEs that are used to decode the generated image from the latent to the pixel space. The training proposed in B-Free [19] used only natural images going to the VAE or inpainted images with StableDiffusion2 to produce a training database of AI-processed images treated as "generated" images. The original images come from the COCO [20] database. For video detection, the frames of the generated class are also encoded by a VAE, and to improve generalization, the high-pass subbands of the original frames are injected in the encoded frames [21]. A similar strategy [22] has recently been proposed by using JPEG compression as a calibration between original images and auto-encoded images, the calibrated auto-encoded images forming the class of generated images during training. Note that all three methods [21], [19] and [22] finetune DINOv2 to train the classifier.

III. METHODOLOGY

A. Benchmark construction

As the generative inpainting landscape evolves rapidly, existing datasets such as B-Free [19] are becoming less representative of modern synthetic techniques. To provide a robust foundation for both training and evaluation, we require a dataset that captures the diversity of contemporary models while specifically challenging the ability of detectors to identify modifications across various scales.

To increase diversity in source images and generative models, we introduce **FOSSIL (Foundation dataset for Occluded Structure Synthesis and Inpainting Localization)**, which pairs real images from the COCO dataset with tampered images generated by various SOTA inpainting models (e.g., DreamShaper8, SD1.5, SD2, SDXL, SD3, Flux Kontex, and Qwen Edit Image). The data construction pipeline involves two primary stages: (1) **Segmentation and Masking**, where we employ YOLOv11 to identify objects and select a mask whose area ratio r is closest to a uniform sample $r \sim U(0, 1)$ to ensure variety in mask sizes; and (2) **Inpainting and Synthesis**, where objects are either removed or replaced with semantically similar counterparts. To maintain high-quality background consistency, 50% of the images retain their original pristine backgrounds. The resulting dataset comprises 12,704 images (∼2,000 per model), with qualitative examples shown in Figure 2. Note that while the proposed dataset is based on COCO, the methodology can be applied to any dataset at hand. The dataset is publicly available here: FOSSIL .

B. Multiclass classification

Most contemporary inpainting models employ a three-stage pipeline: first, the input image is encoded into a latent representation via a VAE encoder; second, the inpainting operation is performed within this latent space; and finally, the modified representation is decoded back into the pixel space using a VAE decoder. Consequently, when an image is generated by these models, the inpainted regions exhibit a superposition of both generative synthesis artifacts and reconstruction-based artifacts (VAE artifacts). In contrast, the regions outside the inpainted mask – which undergo only reconstruction – retain only VAE-specific artifacts.

Because the reconstructed regions maintain semantic content similar to the original image, they are typically treated as "real" in standard evaluation protocols. This leads to a conventional training setup involving only two classes: inpainted vs. not inpainted (semantically real). However, this configuration is suboptimal; by grouping VAE reconstructions with "Real" regions, we introduce conflicting supervisory signals. The model is forced to treat VAE-specific signatures as "authentic" in one class while simultaneously encountering those same signatures within the "Inpainted" class. To resolve this conflict, we adopt a three-class training strategy: Real, VAE, and Inpainted, as shown in Figure 1. This approach compels the model to learn a more granular and discriminative feature representation.

C. Training Strategy

We evaluate two distinct training paradigms in this work. The first employs the pre-trained DINOv2-B [19] model optimized for deepfake detection as a frozen feature extractor.



Fig. 2. Qualitative examples of the proposed dataset. For each example, we show the source image, the generated mask used for manipulation, and the final inpainted result (from left to right).

In this configuration, we append either a simple linear layer (769 trainable parameters) or a sequence of two transformer blocks (14.2M trainable parameters) to the backbone.

The second approach leverages LoRA to achieve parameter-efficient fine-tuning. Specifically, we apply LoRA to the Query (Q) and Value (V) projections in each block of the DINOv2 architecture with a rank of $r = 64$ and an alpha of $\alpha = 64$, followed by a linear layer for patch-level classification. By keeping the majority of the model weights frozen, this method reduces the total number of trainable parameters to 2.4M.

Our primary focus is the second approach, which yields significantly better performance than the two-block configuration while remaining more parameter-efficient. Furthermore, we consider two types of backbones: **DINOv2** pretrained on LVD-142M [16] using self-supervised learning, and **DINOv2-B** [19] fine-tuned on the B-Free dataset for deepfake detection. We demonstrate that the DINOv2-B [19] backbone yields improvements in localization.

D. ViT for Localization

Since the `[CLS]` token aggregates information from the entire image, it is unsuitable for pixel-level localization. Instead, we focus on the patch embeddings (patch tokens) $[z_1^{(L)}; \ldots; z_N^{(L)}]$, each of which encodes the visual content of its corresponding patch. To obtain a per-patch modification probability p_i, we attach a lightweight linear classification head to each embedding and apply the softmax:

$$p_i = \mathit{softmax}(W z_i^{(L)} + b), \quad i = 1, \ldots, N, \tag{1}$$

where $W z_i^{(L)} + b$ is the i-th patch logit.

Beyond the standard binary classification (real vs. inpainted), we introduce a third category: the **VAE class**. This class represents regions that were processed by the generative model but lie outside the primary mask, meaning they undergo regeneration without altering the underlying content. By explicitly classifying these generative traces as a separate category, we encourage the model to distinguish between general generative artifacts and the specific forged content of the inpainted object (see also Figure 1).

The model is trained with Weighted Cross Entropy loss, with the class weights computed as: $w_0 = 1$, $w_1 = \min(N_0/N_1, \tau)$, $w_2 = \min(N_0/N_2, \tau)$, where N_0, N_1, and N_2 denote the number of pixels for class 0 (real), class 1 (inpainted), and class 2 (VAE), respectively. The stability constant τ is set to 50 in our experiments.

TABLE I

OVERVIEW OF EVALUATION DATASETS WITH THEIR REAL IMAGE SOURCES, INPAINTING GENERATORS (SD = STABLE DIFFUSION, BTB: BEYOND THE BRUSH, COCOG: COCOGLIDE, MB = MAGIC BRUSH, MD = MY DATASET). THE PRESENCE OF AN ORIGINAL BACKGROUND (ORIG BG): ✓ OR A BACKGROUND THAT IS AUTO-ENCODED: × IN THE FORGED IMAGES IS ALSO INDICATED.

Dataset	**Real Source**	**Inpainting Schemes**	Orig Bg	Fake Images
BtB [23]	Flickr30k	Fooocus [24], SDXL	✓	7.4k
COCOG [9]	MS-COCO [20]	GLIDE [25]	✓	512
TGIF [26]	MS-COCO [20]	SD2, SDXL, Firefly	✓	4.1k
SAGI-SP [27]	MS-COCO [20] RAISE [28]	BrushNet, PowerPaint, HD-Painter, ControlNet	✓	2k
SAGI-FR [27]	OpenImages [29]	Inpaint-Anything	×	1.3k
SIDA [30]	MS-COCO [20]	Latent Diffusion	×	20k
MB [31]	MS-COCO [20]	Magic Brush	×	465
FOSSIL-SP	MS-COCO [20]	SD1.5, SD2, SD3, SDXL, Flux Kontex,	✓	12.7k
FOSSIL-FR		Qwen Edit Image	×	

Sliding-window Inference: To localize inpainted regions in images of arbitrary resolution, we avoid down-sampling to preserve discriminative forensic cues. Instead, we apply a sliding window of size $w \times w = 504 \times 504$ with stride s to the input image, producing a set of overlapping patches. For each patch, we compute logits via a linear classification head and sum them over all windows that cover a given pixel. Because the decision threshold remains fixed at 0 (or 0.5 in probability space), this summation is mathematically equivalent to averaging, as linear scaling does not alter the final classification. Grid search reveals that $s = 128$ yields performance comparable to $s = 64$ while doubling inference speed; thus, we adopt $s = 128$ for all subsequent experiments. The window size w matches the spatial resolution of our training set.

We observe that multiple overlapping windows improve prediction quality. Therefore, we upscale small test images to at least 1016×1016 pixels using bicubic interpolation to guarantee a 5×5 grid of windows. This ensures a diverse set of crops prior to averaging, further enhancing the accuracy of the localization map; the resulting masks are then resized to the original image dimensions.

While our detector is built upon the DINOv2-B [19] backbone, it departs fundamentally from its conventional use. Standard DINOv2-B averages probabilities from five global crops (four corners and the center) to produce a single binary decision. This strategy often fails when the inpainted region is small, as the global image statistics remain largely authentic. In contrast, our sliding-window approach yields a dense per-pixel prediction map that can reveal localized forgeries even when global statistics are unchanged.

IV. RESULTS

A. Experimental Setup

Training: We fine-tune the detector on the B-Free dataset, which contains 51k real and 309k synthetic images and a subset of the TGIF dataset , which contains 118k synthetic images. The real samples are drawn from MS-COCO, whereas the synthetic images from the B-Free dataset are reconstructed

TABLE II

AVERAGE PIXEL-LEVEL IOU AND F1 SCORE COMPARISON ON IMAGE FORGERY LOCALIZATION. THE BEST RESULTS ARE SHOWN IN BOLD, THE SECOND BEST ARE UNDERLINED.

Model	Byd the Brush		CocoGlide		TGIF		SAGI-FR		SAGI-SP		SIDA		MB		FOSSIL-SP		FOSSIL-FR		Average	
	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1	IOU	F1
DinoLizer (ours)	**0.38**	**0.50**	**0.74**	**0.82**	0.60	0.68	**0.29**	**0.40**	**0.56**	**0.64**	**0.60**	**0.69**	**0.65**	**0.76**	**0.66**	**0.72**	**0.45**	**0.53**	**0.55**	**0.64**
DINOv3-IML [18]	0.16	0.21	0.48	0.56	0.23	0.31	0.23	0.28	0.32	0.39	0.23	0.29	0.25	0.35	0.33	0.39	0.16	0.21	0.27	0.33
TruFor [9]	0.22	0.30	0.29	0.36	**0.70**	**0.80**	0.16	0.20	0.48	0.55	0.18	0.25	0.28	0.38	0.63	0.71	0.11	0.16	0.34	0.41
SAFIRE [10]	0.12	0.17	0.40	0.47	0.21	0.28	0.21	0.28	0.27	0.33	0.21	0.28	0.21	0.29	0.29	0.36	0.15	0.21	0.24	0.30
ManTra-Net [11]	0.06	0.10	0.26	0.35	0.26	0.35	0.02	0.03	0.27	0.35	0.03	0.05	0.04	0.07	0.11	0.17	0.02	0.04	0.12	0.17
CAT-Net v2 [12]	0.13	0.20	0.17	0.24	0.14	0.23	0.03	0.05	0.11	0.18	0.08	0.13	0.02	0.04	0.07	0.12	0.01	0.02	0.09	0.14
SparseViT [14]	0.05	0.08	0.32	0.39	0.11	0.15	0.09	0.11	0.12	0.15	0.12	0.16	0.15	0.22	0.20	0.24	0.06	0.08	0.13	0.16

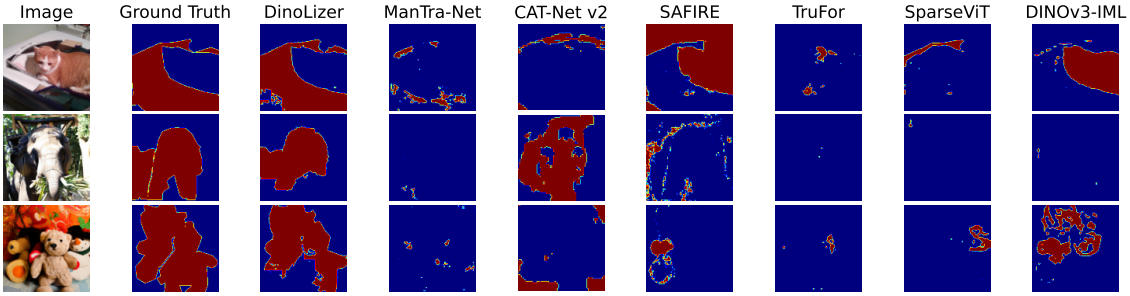


Fig. 3. Comparison with existing methods on the CoCoGlide dataset.

and inpainted variants of the real images produced with Stable Diffusion 2.1. The synthetic images from the TGIF dataset are generated using Stable Diffusion 2.1, XL, Flux 1 Dev, Flux 1 Schnell, and Flux 1 Fill.

DinoLizer is built upon a ViT-B/14 backbone with registers, refined on the B-Free dataset [19]. We train the model with LORA adaptation and a linear classification head with 3 classes (real, VAE, and Inpainted) using the `AdamW` optimizer with a batch size of 64, an initial learning rate of 10^{-3}, and a weight decay of 10^{-3}. The learning rate is managed by a `lrdrop` scheduler, which halves the rate whenever the validation loss fails to improve for four consecutive epochs. Training is conducted for a maximum of 100 epochs on a single NVIDIA A6000 GPU (48GB), with early stopping triggered once the learning rate falls below 10^{-7}. To enhance robustness against common image processing operations, we employ a diverse suite of augmentations via the Albumentations library on 504×504 images using Lanczos interpolation. These include random horizontal flips ($p = 0.5$), 90° rotations ($p = 1.0$), and a stochastic selection (each $p = 0.25$) of `RandomResizedCrop`, `RandomCrop`, direct resizing, or upscaling by 200%–300% followed by `RandomCrop`. Furthermore, we incorporate uniform box blur ($p = 0.1$), Gaussian noise ($\sigma \in [51, 112], p = 0.1$), color jitter ($p = 0.1$), and a two-stage JPEG compression process where the first stage uses a quality factor $QF \in \{40, \ldots, 100\}$ ($p = 0.5$) and an optional second stage ($p = 0.1$). During validation, we accelerate the evaluation process by applying only a center crop followed by an optional JPEG compression ($p = 0.5$) with a randomly sampled quality factor.

Testing: We evaluate our method on 5 public datasets of generative inpainting. In particular, we use the inpainted images from Beyond the Brush [23], CocoGlide [9], TGIF [26], SAGI-SP [27], SAGI-FR [27], SIDA [30], Magic Brush [31], our benchmark FOSSIL-FR and FOSSIL-SP. SP (splicing) and FR (fully regenerated) come from the same dataset, but we separate them in 2 situations. SP is the case where the region outside the mask is kept from the original image, while FR is the case where the region outside the mask is also regenerated by the inpainting model. Across all datasets we consider roughly 48k images, see the details in Table I.

Evaluation Metrics: We use IOU (Intersection over Union) and F1 score to evaluate the localization performance. A fixed threshold of 0.5 is used to binarize the predicted probability map. We consider IOU equal to 1 when both the predicted and ground truth masks are empty (i.e., no inpainted pixels), which is necessary during training, as there are also fully pristine images and reconstructed images.

B. State-of-the-art Comparison

Qualitative Results: We evaluate DinoLizer against a representative set of state-of-the-art inpainting-forgery detectors, employing only the publicly released weights. The competing methods are TruFor [9], SAFIRE [10], ManTra-Net [11], CAT-Net v2 [12], and SparseViT [14]. All models are used off-the-shelf without any additional re-training.

Table II summarizes the quantitative results. DinoLizer attains the highest F1 score across all datasets, with the sole exception of the TGIF benchmark where TruFor slightly surpasses it. On average, our approach outperforms the other detectors by at least 20% in F1. We can also notice that although TruFor performs well on FOSSIL-SP, its accuracy drops significantly on FOSSIL-FR (with regenerated background), while our method is more robust to both situations.

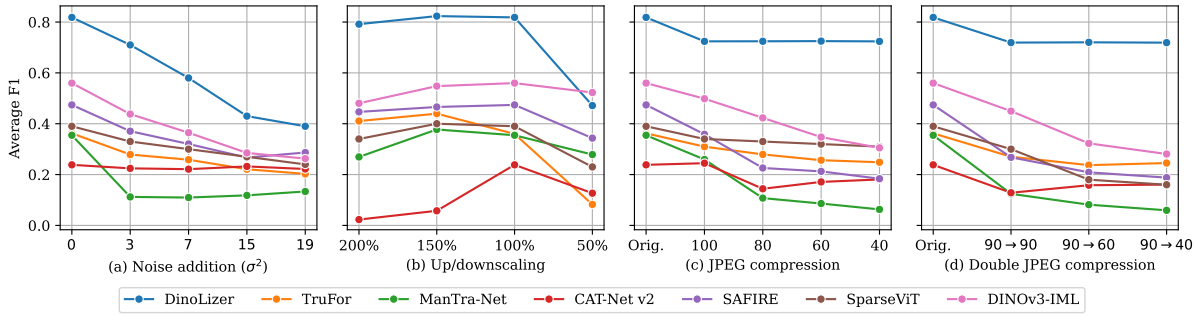


Fig. 4. F1 score performance under different types of perturbations on the CocoGlide dataset: (a) gaussian noise, (b) resizing, (c) JPEG compression, and (d) double JPEG compression.

TABLE III
ABLATION STUDY ON THE IMPACT OF LORA ADAPTATION, PRETRAIN, AND MULTICLASS ON THE MODEL'S PERFORMANCE

Pretrain	Setup	# Classes	IOU	F1
BFree [19]	1 linear layer	2	0.31	0.41
	2 transformer blocks	2	0.40	0.50
		3	0.49	0.58
	LoRA Adaptation	2	0.51	0.60
		3	**0.55**	**0.64**
LVD-142M [16]	2 transformer blocks	3	0.44	0.54
	LoRA Adaptation	3	0.54	0.62

Figure 3 presents a qualitative comparison of DinoLizer against the SOTA on the CoCoGlide dataset, demonstrating that our method can localize well the inpainted regions.

Robustness Evaluation: We evaluate the robustness of the detectors against four common post-processing operations: **JPEG compression** with quality factors $QF \in \{100, 80, 60, 40\}$; **double JPEG compression** (first pass $QF_1 = 90$, second pass $QF_2 \in \{90, 60, 40\}$); **bicubic resizing** to 50%, 150%, and 200% of the original resolution; and **additive Gaussian noise** with variance $\sigma^2 \in \{3, 7, 15, 19\}$.

Figure 4 illustrates the robustness of DinoLizer across various perturbation types. Across nearly all categories, DinoLizer significantly outperforms the baseline methods. Notably, our model exhibits remarkable stability under JPEG (double) compression (Figure 4c, d), maintaining a consistent F1 score from QF=100 to QF=40. While DinoLizer shows sensitivity to Gaussian noise (Figure 4a), it remains the top-performing model across all noise levels; however, due to the notable drops in performance, we identify this sensitivity as an important area for further investigation in future work. Furthermore, in the up/downsampling task (Figure 4b), DinoLizer maintains a substantial lead over most baselines, with the sole exception of the 50% downsampling case, where it slightly underperforms compared to DINOv3-IML.

V. Ablation Studies

A. LORA Adaptation

We evaluate the effectiveness of LoRA in our model by comparing its performance with a frozen model where we train with a 2-transformer-block head on top of the frozen model. The results, in Table III, indicate that LoRA adaptation significantly enhances the model's ability to learn discriminative features for inpainting localization, leading to improved performance across all evaluation metrics while reducing the number of trainable parameters (6% in IOU for the 3-class setting and 11% in IOU for the 2-class setting).

B. 3 classes vs 2 classes

We conduct an ablation study to evaluate the impact of our proposed 3-class training strategy (Real, VAE, and Inpainted) compared to the conventional 2-class setup (Real and Inpainted). Specifically, we investigate two primary configurations: 2-class and 3-class settings, both of which utilize LoRA adaptation or two additional transformer blocks on top of the frozen backbone. Furthermore, we establish a baseline by evaluating a model with a frozen backbone and a simple linear head for the 2-class classification task. For the 2-class setup, we use the Weighted Binary Cross-Entropy loss. The three-class setup follows that proposed in Section III-D.

Our results in Table III demonstrate that the 3-class configuration significantly enhances the model's ability to differentiate between real/VAE and inpainted regions, leading to superior localization performance across all evaluation metrics.

C. Pretrained vs Not pretrained

We investigate the impact of fine-tuning DINOv2 on the B-Free dataset by comparing our model's performance with and without this step. As shown in Table III, while this fine-tuning yields almost negligible improvements in overall performance for our primary configuration (LORA), it provides a more noticeable gain when using only two transformer blocks.

VI. Conclusion

We propose a novel framework that leverages the versatile DINO family of Vision Transformers to localize generated image regions, achieving state-of-the-art performance across multiple benchmarks. Additionally, we introduce a new dataset, **FOSSIL**, which incorporates contemporary generative inpainting methods, such as Flux Kontex and Qwen Edit Image, providing a more challenging evaluation scenario for existing detectors.

Our methodology addresses the localization challenge in inpainting forgeries by explicitly modeling auto-encoded regions as a distinct class alongside real and inpainted areas. By employing Low-Rank Adaptation, we are able to fine-tune the model with only 2.4M trainable parameters while maintaining state-of-the-art performance. The proposed solution is highly computationally efficient: by leveraging DINO's patch tokens and a sliding window approach, both training and inference can be performed on a single GPU, even for large images. Furthermore, our comprehensive data augmentation strategy ensures robustness against common image post-processing operations, most notably JPEG (double) compression. This resilience is particularly valuable given the diverse processing pipelines that forgeries often undergo before analysis.

REFERENCES

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2022, pp. 10 684–10 695.

[2] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2112.10741*, 2021.

[3] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2022, pp. 2149–2159.

[4] X. Ju, X. Liu, X. Wang, Y. Bian, Y. Shan, and Q. Xu, "Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion," in *European Conference on Computer Vision*. Springer, 2024, pp. 150–168.

[5] J. Zhuang, Y. Zeng, W. Liu, C. Yuan, and K. Chen, "A task is worth one word: Learning with task prompts for high-quality versatile image inpainting," in *European Conference on Computer Vision*. Springer, 2024, pp. 195–211.

[6] H. Manukyan, A. Sargsyan, B. Atanyan, Z. Wang, S. Navasardyan, and H. Shi, "Hd-painter: high-resolution and prompt-faithful text-guided image inpainting with diffusion models," in *The Thirteenth International Conference on Learning Representations (ICLR)*, 2023.

[7] S. Xie, Z. Zhang, Z. Lin, T. Hinz, and K. Zhang, "Smartbrush: Text and shape guided object inpainting with diffusion model," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 22 428–22 437.

[8] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, 2023, pp. 3836–3847.

[9] F. Guillaro, D. Cozzolino, A. Sud, N. Dufour, and L. Verdoliva, "Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2023, pp. 20 606–20 615.

[10] M.-J. Kwon, W. Lee, S.-H. Nam, M. Son, and C. Kim, "Safire: Segment any forged image region," vol. 39, no. 4, 2025, pp. 4437–4445.

[11] W. A. Yue Wu and P. Natarajan, "Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[12] M.-J. Kwon, S.-H. Nam, I.-J. Yu, H.-K. Lee, and C. Kim, "Learning jpeg compression artifacts for image manipulation detection and localization," *International Journal of Computer Vision*, vol. 130, no. 8, pp. 1875–1895, 2022.

[13] L. Su, X. Ma, X. Zhu, C. Niu, Z. Lei, and J.-Z. Zhou, "Can we get rid of handcrafted feature extractors? sparsevit: Nonsemantics-centered, parameter-efficient image manipulation localization through spare-coding transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 7, 2025, pp. 7024–7032.

[14] X. Chen, Z. Liu, H. Tang, L. Yi, H. Zhao, and S. Han, "Sparsevit: Revisiting activation sparsity for efficient high-resolution vision transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2061–2070.

[15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[16] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *Transactions on Machine Learning Research Journal*, pp. 1–31, 2024.

[17] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, 2021, pp. 7262–7272.

[18] J. Yu, Q. Feng, Z. Wang, and X. Ma, "Dinov3 beats specialized detectors: A simple foundation model baseline for image forensics," 2026. [Online]. Available: https://arxiv.org/abs/2604.16083

[19] F. Guillaro, G. Zingarini, B. Usman, A. Sud, D. Cozzolino, and L. Verdoliva, "A bias-free training paradigm for more general ai-generated image detection," in *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2025, pp. 18 685–18 694.

[20] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015. [Online]. Available: https://arxiv.org/abs/1405.0312

[21] R. Corvi, D. Cozzolino, E. Prashnani, S. De Mello, K. Nagano, and L. Verdoliva, "Seeing what matters: Generalizable ai-generated video detection with forensic-oriented augmentation," *arXiv preprint arXiv:2506.16802*, 2025.

[22] R. Chen, J. Xi, Z. Yan, K.-Y. Zhang, S. Wu, J. Xie, X. Chen, L. Xu, I. Guan, T. Yao *et al.*, "Dual data alignment makes ai-generated image detector easier generalizable," *arXiv preprint arXiv:2505.14359*, 2025.

[23] G. Bertazzini, C. Albisani, D. Baracchi, D. Shullani, and A. Piva, "Beyond the brush: Fully-automated crafting of realistic inpainted images," in *2024 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2024, pp. 1–6.

[24] L. Zhang, "Fooocus," https://github.com/lllyasviel/Fooocus, 2023, online; accessed 2025-11-11.

[25] A. Q. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. Mcgrew, I. Sutskever, and M. Chen, "GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 16 784–16 804. [Online]. Available: https://proceedings.mlr.press/v162/nichol22a.html

[26] H. Mareen, D. Karageorgiou, G. V. Wallendael, P. Lambert, and S. Papadopoulos, "Tgif: Text-guided inpainting forgery dataset," in *2024 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2024, pp. 1–6.

[27] P. Giakoumoglou, D. Karageorgiou, S. Papadopoulos, and P. C. Petrantonakis, "Sagi: Semantically aligned and uncertainty guided ai image inpainting," in *International Conference on Computer Vision (ICCV)*, 2025.

[28] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "Raise: a raw images dataset for digital image forensics," in *Proceedings of the 6th ACM Multimedia Systems Conference*, ser. MMSys '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 219–224. [Online]. Available: https://doi.org/10.1145/2713168.2713194

[29] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International journal of computer vision*, vol. 128, no. 7, pp. 1956–1981, 2020.

[30] Z. Huang, J. Hu, X. Li, Y. He, X. Zhao, B. Peng, B. Wu, X. Huang, and G. Cheng, "Sida: Social media image deepfake detection, localization and explanation with large multimodal model," 2025. [Online]. Available: https://arxiv.org/abs/2412.04292

[31] K. Zhang, L. Mo, W. Chen, H. Sun, and Y. Su, "Magicbrush: A manually annotated dataset for instruction-guided image editing," 2024. [Online]. Available: https://arxiv.org/abs/2306.10012