DiffSeg: Towards Detecting Diffusion-Based Inpainting Attacks Using Multi-Feature Segmentation

Raphael Antonius Frick, Martin Steinebach
Fraunhofer SIT — ATHENE Center
Rheinstraße 75, Darmstadt, Germany
{raphael.frick, martin.steinebach}@sit.fraunhofer.de

Abstract

With the advancements made in deep learning over the past years, creating convincing media manipulations has become easy and accessible than ever before. In particular, diffusion models such as Stable-Diffusion allow users to synthesize realistic images based on a given text input. Apart from synthesizing entirely new images, diffusion models can also be used to make edits to images using inpainting. To combat the spread of disinformation and illegal content created with diffusion-based inpainting, this paper presents a new detection method based on multi-feature segmentation. Apart from information derived from the raw pixel values, noise, and frequency information are also exploited to detect and localize regions that have been subject to editing. Evaluation results strongly suggest that the proposed method can achieve high mIoU and AUC scores, outperforming state-of-the-art methods, even for syntheses generated by unseen diffusion models, or highly compressed images.

1. Introduction

Nowadays, social media platforms are frequently used to share images and videos. In addition to the traditional social media platforms and messaging applications that support multimedia, such as X (formerly known as Twitter) and Facebook, there are some social media platforms that are particularly focused on multimedia, including Instagram and Tiktok. However, the authenticity of the shared content is not always ensured, as images and videos can be subject to filters provided by the platforms or dedicated editing software. In the past, editing images was time-consuming and required certain expertise. With the advancements that have been made in artificial intelligence (AI) over the years, various AI models have been developed that allow new image content to be synthesized or existing ones to be edited. In addition to deepfakes [17] and generative adversarial networks [7], diffusion models [20] have gained a lot of popularity as of late. Diffusion models such as GPT-4 [15] and Stable-Diffusion [20] allow synthesizing new images conditioned by a text prompt as well as editing existing ones using image-to-image synthesis or inpainting convincingly. The latter can be used to replace parts of an image with new content guided by text input. It can be used to remove unwanted objects, but also enables adding new objects, by which the context of an image is likely to be altered. Inpainting thus has the potential to be used to create and disseminate disinformation on social media. Therefore, it is of great concern to be able to identify images that have been subject to diffusion-based inpainting and that have been shared on social media, especially since sometimes journalists rely on user-created content as it is often the only source available.

In this paper, a novel method for detecting and locating regions in images that have been subject to diffusion-based inpainting is presented. It takes advantage of multi-feature segmentation, namely features based on noise, frequency and raw pixel data analysis. By using a variety of features, the aim is not only to improve detection performance, but also explainability, which is often lacking in fully data-driven recognition models. The main contributions made in this paper are as follows:

- The paper presents a novel method for detecting diffusion-based inpainting based on multi-feature segmentation.
- The model takes advantage of deep-learning-based and hand-crafted features that provide interpretability usually not available in fully deep-learning-based solutions.
- It is able to provide high robustness to low and high JPEG compression and high generalizability to unseen diffusion models.
- In experiments, it was able to outperform state-of-the-art methods by a large margin.

The remainder of this paper is structured as follows: in Section 2 an introduction to diffusion-based inpainting is

given. Section 3 presents related work used to identify generated and altered image content. The proposed approach to detect images subject to diffusion-based inpainting is presented in Section 4. The results achieved by the method on the test set and its robustness and generalizability is showcased in Section 5. The paper then concludes in Section 6 with an outlook at future work.

2. Diffusion-based Image Editing

Over the years different methods for synthesizing and editing images have been proposed. While generative adversarial networks [3], are able to synthesize images of high quality, they often lack detailed control over the synthesized content. Further, edited pre-existing images often have degraded image quality due to the optimization process involved in the GAN-inversion process [23].

Diffusion models try to solve this issue by providing control over the synthesized content using e.g., a text-prompt [20]. During training, noise is gradually added to the training data first and then the model learns to reverse the noise addition process to recover the original data. For this, a noise prediction model is used to determine the noise pattern to be removed, by receiving additional information in the form of a given text prompt. This enables the model to steer the denoising process in a direction so that the final image content matches the description in the text prompt. Since performing the diffusion process in the pixel space of the image would be computationally expensive, latent diffusion models are performing it in a latent space instead. In this case, a variational autoencoder is often used to transform an image into the latent space and to transform a diffused latent embedding back into an image. Recent diffusion models are not limited to image data, and can also be used e.g. for the synthesis of audio [9] and 3D mesh objects [18].

Besides being able to synthesize entire images using a text-prompt, existing images can be edited using inpainting. Inpainting allows re-filling a self-defined region inside an image with new content. This requires a 2D mask as input in addition to the image to be edited, in order to specify which areas should be retained and which should be modified. The selected area is then synthesized with the help of a given text prompt and the chosen diffusion model.

3. Related Work

In the past, several methods for detecting diffusion models have been proposed. Most of them try to detect and annotate different models using a frequency analysis [2, 16, 19]. They found that certain models introduce specific patterns in the spectrum of the synthesized images. In other words because different diffusion models have learned to synthesize images with different weights, it results in certain frequencies being less represented or overrepresented depending on the model used. For the analysis, the spectrum is calculated for the whole image. This is however not directly applicable to images that have been subject to inpainting attacks and which therefore consist of authentic and synthetically created parts. Since the related works base their decision frequency analysis, compression, e.g. JPEG compression, may have an impact on the detection capabilities.

Detection techniques, that consider inpainting attacks based on traditional image processing, are either based on the analysis of handcrafted features, e.g., on a CPA-analysis [6] or double JPEG-compression analysis [12], or of features gained by a deep learning model [5, 21].

ManTraNet [21] is an image forgery detection model that has been trained on several post-processing operations, including scaling, noise and compression.

TruFor [5] combines the analysis of raw RGB pixel data with the analysis of noise residuals obtained by a denoising deep learning based model. During training, it was also trained on diffusion-based inpainted images synthesized using GLIDE [14].

However, deep-learning-based methods often do not offer explanatory capabilities, since the automatically extracted features are too abstract to interpret. Providing explanations is a prerequisite for the use of detection systems for some use cases, e.g., authenticity reports in court proceedings or in journalism.

Thus, in this paper, a combination of handcrafted and deep-learning-based features is used to detect diffusion-based inpainting attacks as they can provide high performance while in addition providing interpretability of the obtained results.

4. Proposed Approach

In this section, the proposed approach for detecting diffusion-based inpainting using multi-feature segmentation is presented. The overall architecture is shown in Figure 1 and consists of three types of modules that are interconnected and form the classification system: *feature extraction modules*, *segmentation modules* and a *fusion module*.

4.1. Feature Extraction Module

For the detection, a total set of three features are explored: *RGB-based*, *frequency-based*, and *noise-based* features. As aforementioned, the aim is to provide high detection accuracies while maintaining interpretability and generalizability using handcrafted features. The feature extraction module is hereby used to extract the necessary features.

RGB-based features are derived directly from the deep-learning-based classification model and do not require explicit feature extraction. The raw RGB data is therefore

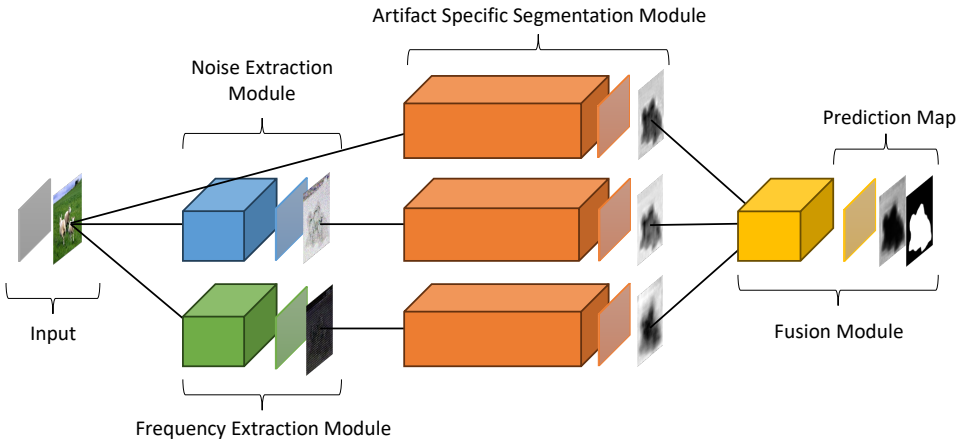


Figure 1. Outline of the architecture used in the proposed detection method

passed straight to the segmentation module after performing normalization (Figure 2b).

Previous work has found that certain frequencies in the synthesized images are either overrepresented or underrepresented. However, since related work perform the analysis on the entire image, they are unsuitable for detecting inpainting attacks, where parts of the image might be authentic. Therefore, in this work, the frequency analysis is performed using non-overlapping windows instead (Figure 2c). Using a window-based approach, deviations in the frequency representation of authentic and non-authentic image regions are attempted to be found.

Images shot with a camera often exhibit a certain noise pattern introduced by the camera sensor. Analysis of this noise has been successfully used in image forensics for camera identification [13]. Since the regions affected by the inpainting attack are denoised as part of the reverse diffusion process, it is unlikely that the sensor noise will be preserved and that the exhibited noise will differ in forged areas of the image. For feature extraction, a denoiser is used to predict the noise. By subtracting the noise from the input image, the noise residual is obtained (Figure 2d), which is used as a feature for the subsequent segmentation module.

4.2. Segmentation Module

For each of the features (RGB, frequency, and noise), a specific segmentation module is created. They consist of segmentation networks that take the extracted feature map as input and provide a probability map as output (Figure 2e), indicating areas that have likely been modified.

4.3. Fusion Module

The fusion module then takes the predicted probability maps obtained from each of the three artifact-specific segmentation modules and fuses them into one single prediction map (Figure 2f). For this purpose, each of the probability maps are first weighted by the validation loss the respective segmentation model used in the segmentation module. As a result, models with good performance on the validation set have a greater impact on the classification result than models with less good performance. The results are then merged into a single one by forming their linear combination. Here, we specifically did not take advantage of ensemble learning, such as stacked classification, for multiple reasons. First, by this method, we are able to improve efficiency with regard to the processing time, and second, we can mitigate possible effects of overfitting the meta-classifier to a distribution found within the images of the train set. To obtain a binary prediction map, the output is binarized with a threshold. The prediction map is to reconstruct the mask (Figure 2g) used to perform the inpainting attack.

5. Evaluation

In this section, the implementation of the classification architecture is described in detail and evaluation results are presented for images created with different diffusion models and additionally for inpainted images subjected to JPEG compression.

5.1. Datasets

Images from the Common Objects in Context (COCO) dataset [11] were used for training and evaluation. The dataset consists of a train, validation, and test split and features annotations such as image descriptions and segmentation masks.

Here, solely the 5,000 images of the validation set were used. The images are diverse in terms of their resolution, quality and the content they depict. From each image, its associated annotation for the segmentations is used to build an inpainting mask. For this purpose, the segmentation mask

for randomly selected objects within the list of annotations is used. Then, the area is increased using a dilation operation of size 15x15 applied for five iterations. Subsequently, a text description of the source image is derived using BLIP2 [10], specifically using the pre-trained *blip2-opt-2.7b* model.

Once the inpainting masks and the textual description have been obtained, images are synthesized using Automatic1111 [1] a front-end for diffusion models. In total, a set of four models were used during synthesis: Stable-Diffusion 1.5[1], Stable-Diffusion 1.5 Inpainting[2], Stable-Diffusion 2 Inpainting[3] and Anything v4.5 Inpainting[4]. All these models were sourced from HuggingFace. For better reproducibility, a seed of 104 was used in conjunction with a cfg-scale of 5.0 and a denoising strength of 0.7. The fill mode within Automatic1111 was set to original, as it provides the best results in terms of realism.

The resulting N*5000 images (N referring to the N models considered) were then split into three partitions, a train set consisting of 3000 images, a validation set consisting of 500 images and a test set consisting of the remaining images. For training and validation, solely the images synthesized by the Stable-Diffusion 1.5 and Stable-Diffusion 1.5 Inpainting models were used. The images derived from the other models were solely used for generalizability tests during evaluation.

5.2. Implementation

The core of the detection method is based on SegFormer-based segmentation networks [22]. In particular, the *mit-b1* model was fine-tuned to receive feature maps of size 512x512 pixels as input and to return a probability map of 112x112 pixels in size. For this, the images were first resized and then the features were extracted, except for noise features where resizing took place after feature extraction.

For the frequency analysis, a DCT-transform was applied on non-overlapping windows of size 16x16 pixels. Their coefficients were then used as features. Regarding the noise analysis, the application of various denoising techniques is conceivable. Here, denoising was done using a median filter of kernel size 3x3. A small kernel size was used, as any larger kernel size would also affect altering the structure of images, e.g., edges, thus, leading to worse results.

For better robustness against compression, which not only affect the distribution of DCT coefficients, but also the amount of noise present in the image, the input images were augmented using JPEG-compression. Hereby, a quality factor between 75 and 100 was chosen at random.

The segmentation models based on SegFomer have been trained on an NVIDIA A100 for 100 epochs with a batch size of 32. Moreover, Adam [8] was used as optimizer with an initial learning rate of $6 * 10^{-5}$ in conjunction with a binary cross-entropy loss. By using model checkpoints, only the best performing models are retained in the validation set, preventing overfitting.

[1] runwayml/stable-diffusion-v1-5
[2] runwayml/stable-diffusion-inpainting
[3] stabilityai/stable-diffusion-2-inpainting
[4] Koolchh/anything-v4.5-inpainting

	Stable-Diffusion 1.5		Stable-Diffusion 1.5 Inpainting	
	Quality Factor		Quality Factor	
	50	*100*	*50*	*100*
RGB	0.946	0.943	0.943	0.991
Noise	0.871	0.894	0.882	0.895
DCT	0.893	0.990	0.926	0.991
Final	0.979	0.995	0.964	0.995

Table 1. Results of the ablation study. Combining multiple feature types can increase the AUC-scores achieved on the test datasets.

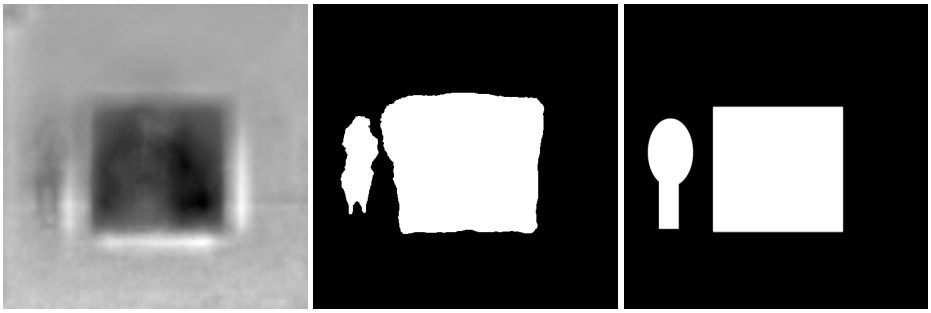
During inference, the probability maps gained from each of the segmentation modules get fused in the fusion module. Hereby, the three probability maps are weighted by the validation loss obtained during training. Here, the RGB-model performed best (validation loss = 0.03771), followed by the Noise-based model (validation loss = 0.10183) and then the DCT-based model (validation loss = 0.12554). It should, however, be noted that the DCT-based model is bound to provide oversized masks than, e.g., the RGB and NOISE-based model, due to utilizing non-overlapping windows of 16x16 pixels in size. However, in experiments investigating the benefits of each feature type, it was revealed that the DCT-based approach worked well for compressed images (Table 1). Once the unified probability map is obtained, it is upsampled to the dimensions of the input image first. Then, it is binarized using a threshold. By estimating the cutoff point on the validation split, a threshold value of 148 was identified and was used in the subsequent tests of the evaluation.

5.3. General Performance Test

For the initial performance test, the model was evaluated solely on images created by the Stable-Diffusion 1.5 and Stable-Diffusion 1.5 Inpainting model. The overall performances with regard to the achieved mIoU score is displayed in Table 2. AUC scores are showcased in Table 3. Hereby, mIoU was used as metric as it is a common metric for evaluating segmentation tasks. However, the main disadvantage is that the results are bound to a specific threshold value. Since the segmentation task presented in this paper only involve two labels, the AUC score can be used in addition to provide a metric independent of a selected threshold value.

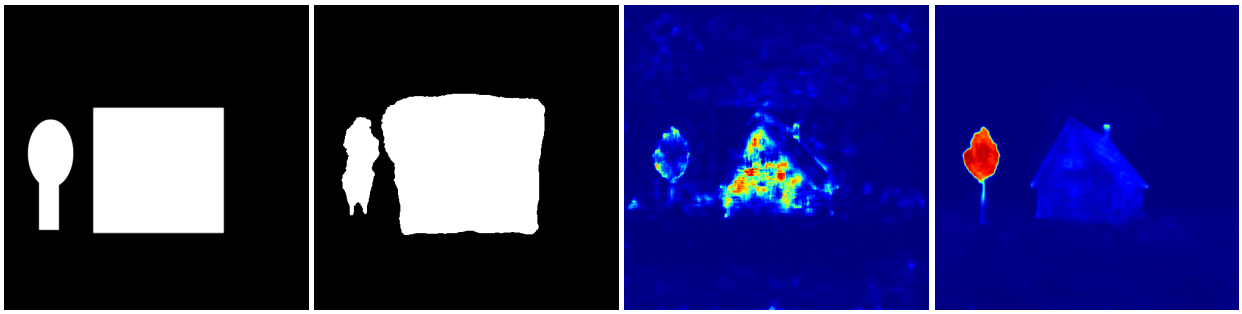
As it can be seen, the overall performance with regard to the mIoU and AUC scores obtained for images compressed using a quality factor of 100 (near lossless), range between

(a) Resized unmodified image (b) Resized input image (c) DCT feature map (d) Noise feature map (inverted)



(e) Probability map (f) Prediction map (g) Ground truth map

Figure 2. Example classification of a stock-image inpainted using an unseen diffusion model (Adobe Photoshop v25.1 Beta).



(a) Ground truth map (b) Ours (c) MantraNet [21] (d) TruFor [4]

Figure 3. Comparison of heatmaps obtained by various methods.

0.921 and 0.932 with respect to the mIoU and 0.995 for the AUC score. This indicates that the model is able to provide high discriminative capabilities that can be used to identify edited regions.

5.4. Robustness Test

Since images are nowadays often uploaded onto social media, it is of great importance to analyze whether these models are capable of synthesizing images, that have undergone compression. Often JPEG compression is used; common quality factors include 80 (for images uploaded to Instagram) and 84 (for images uploaded to X / Twitter) and are thus analyzed in the following robustness test. Adversaries could leverage from JPEG compression to reduce the artifacts to hide a forgery. Thus, compression factors as low as 50 are examined as well. The resulting scores are displayed in Table 3 and 2.

Although compression factors between 50 and 70 were not used for augmenting the data during training, the AUC-scores and mIoU scores achieved on images compressed with those quality factors are still high. This demonstrates the robustness of the system to unseen high compression rates ranging from strong to low compressions.

5.5. Generalizability Test

In many cases, it is not possible to create a distinct detection model for each diffusion model available. As such, the detection methods are required to generalize to images edited by models that have not been seen during training. As part of the evaluation, the test splits featuring images from

	Quality Factor										
	50	*55*	*60*	*65*	*70*	*75*	*80*	*85*	*90*	*95*	*100*
Stable-Diffusion 1.5	0.843	0.849	0.865	0.872	0.878	0.879	0.910	0.927	0.911	0.937	0.932
Stable-Diffusion 1.5 Inpainting	0.800	0.804	0.833	0.837	0.848	0.848	0.891	0.914	0.893	0.929	0.921
Stable-Diffusion 2 Inpainting	0.767	0.775	0.804	0.813	0.818	0.815	0.866	0.895	0.865	0.906	0.899
Anything v4.5 Inpainting	0.829	0.832	0.852	0.863	0.867	0.871	0.903	0.924	0.910	0.934	0.922

Table 2. MIoU values obtained by the proposed approach on the test split. Shaded values refer to settings not seen during training either in terms of the used diffusion model or the quality factor.

	Quality Factor										
	50	*55*	*60*	*65*	*70*	*75*	*80*	*85*	*90*	*95*	*100*
Stable-Diffusion 1.5	0.979	0.980	0.985	0.986	0.988	0.988	0.993	0.996	0.994	0.997	0.995
Stable-Diffusion 1.5 Inpainting	0.964	0.967	0.975	0.976	0.979	0.980	0.990	0.994	0.991	0.996	0.995
Stable-Diffusion 2 Inpainting	0.949	0.951	0.963	0.966	0.968	0.968	0.981	0.989	0.985	0.992	0.991
Anything v4.5 Inpainting	0.968	0.972	0.979	0.981	0.984	0.984	0.991	0.994	0.994	0.997	0.994

Table 3. AUC scores obtained by the proposed approach on the test split. Shaded values refer to settings not seen during training either in terms of the used diffusion model or the quality factor.

Stable-Diffusion 2 Inpainting and Anything v4.5 Inpainting have been considered. While the performance slightly decreases, the AUC (Table 3) and mIoU scores (Table 2) are still high, indicating the model's generalization capabilities.

Further, tests have been made for detecting inpainted images, that have been created using diffusion models in commercial applications, such as Adobe Photoshop v25.1 Beta. However, since the EULA of Adobe Photoshop does not permit batch processing for their generative models, no distinct dataset has been established for evaluation. An example classification is, however, showcased in Figure 2. When comparing the results with the heatmaps obtained from other image-forgery detection methods (3), such as MantraNet [21] and TruFor [4], it showcases, that the proposed model can not only identify the used mask better, the model also did not overfit during training to detect specific objects. This is especially true for TruFor (3d), a data-driven detection method which was only able to identify silhouettes of the inserted tree and of the house, but did was not able to identify the mask used for the forgery.

5.6. Comparison with State-of-the-Art Detection Methods

To obtain a better understanding of the results achieved by our model in contrast to current state-of-the-art detection methods, we also analyzed the performance of both ManTraNet [21] and TruFor [4]. While the latter was already trained on the COCO-validation dataset, ManTraNet was trained on various other datasets. In order to provide a fair comparison between the models and our proposed approach, we evaluated the model's performance on the images derived from the Stable-Diffusion 2.1 Inpainting diffusion model, i.e. images that were not yet seen during training of our model. In addition, we measure the performance using the AUC-score to measure the performance independent of a given threshold. As indicated by Table 4, ManTraNet is unable to correctly identify the inpainting masks. Interestingly, although TruFor was also trained to identify images that were subject to diffusion-based inpainting, the AUC scores are significantly lower than those achieved by our model. This is especially the case when trying to classify images that are compressed with a low-quality factor of 50.

	Quality Factor	
	50	*100*
Ours	0.949	0.991
ManTraNet	0.606	0.602
TruFor	0.622	0.728

Table 4. Comparison of the AUC-scores achieved on the Stable-Diffusion 2 Inpainting images.

6. Conclusion & Future Work

In this paper, a novel method for detecting diffusion-based inpainting attacks is presented. By making use of handcrafted and deep-learning-based features, the method is able to provide good performance, generalization and interpretability capabilities. The model was able to achieve high AUC values between 0.96 and 0.99 and high mIoU values between 0.76 and 0.94 in the dataset with images of seen and unseen synthesis models. The evaluation experiments and the shown example (Figure 2) also indicates that the model can detect inpainted images created with unknown diffusion models without additional training. In ad-

dition, noise- and frequency-based features provided explanations that are often missing in fully deep-learning-driven approaches, and it did not overfit to identifying specific objects within the images.

In the future, the method could be extended to include assessing videos that have been modified using diffusion-based models such as SORA from OpenAI [5].

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research (BMBF) and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of "ATHENE – DREAM" and "Lernlabor Cybersicherheit" (LLCS).

References

[1] AUTOMATIC1111. Stable Diffusion Web UI, 2022. 4

[2] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. 2

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 2

[4] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nick Dufour, and Luisa Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20606–20615, 2022. 5, 6

[5] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa 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)*, pages 20606–20615, 2023. 2

[6] Xiao Jin, Yuting Su, Yongwei Wang, and Z. Jane Wang. Image inpainting detection based on a modified formulation of canonical correlation analysis. In *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–5, 2018. 2

[7] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020. 1

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 4

[9] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *ArXiv*, abs/2009.09761, 2020. 2

[10] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ArXiv*, abs/2301.12597, 2023. 4

[11] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 3

[12] Qingzhong Liu, Andrew H. Sung, Bing Zhou, and Mengyu Qiao. Exposing inpainting forgery in jpeg images under recompression attacks. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 164–169, 2016. 2

[13] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, 2006. 3

[14] 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, 2022. 2

[15] OpenAI. Gpt-4 technical report, 2024. 1

[16] Lorenzo Papa, Lorenzo Faiella, Luca Corvitto, Luca Maiano, and Irene Amerini. On the use of stable diffusion for creating realistic faces: from generation to detection. In *2023 11th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6, 2023. 2

[17] Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr. Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, Sheng Zhang, Pingyu Wu, Bo Zhou, and Weiming Zhang. Deepfacelab: Integrated, flexible and extensible face-swapping framework, 2021. 1

[18] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *ArXiv*, abs/2209.14988, 2022. 2

[19] Jonas Ricker, Simon Damm, Thorsten Holz, and Asja Fischer. Towards the detection of diffusion model deepfakes. *ArXiv*, abs/2210.14571, 2022. 2

[20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and BjÃ¶rn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2

[21] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9535–9544, 2019. 2, 5, 6

[22] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, José Manuel Álvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *ArXiv*, abs/2105.15203, 2021. 4

[23] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. *ArXiv*, abs/2004.00049, 2020. 2

[5] https://openai.com/research/video-generation-models-as-world-simulators