Localizing AI Image Manipulations by Learning Generative Noise Signatures using Diffusion Based Noise Priors

Elijah Shannon
Department of Computer Science
University of Alabama in Huntsville
Huntsville, AL, United States
elijahgshannon@gmail.com

Chaity Banerjee
Department of Computer Science
University of Alabama in Huntsville
Huntsville, AL, United States
chaity.banerjee.mukherjee@uah.edu

***Abstract*—This work presents a two-stage segmentation framework for detecting AI-generated alterations in images. The proposed method first extracts Diffusion Noise Features (DNFs) through diffusion inversion to capture statistical differences between authentic and synthesized regions. These DNFs are concatenated with RGB channels and input into a Swin U-Net for pixel-level localization of altered areas. To ensure fair evaluation, we reconstruct portions of the OpenImages, ADE20k, and CelebA-HQ datasets from the Semi-Truths Evalset by recomposing fake content into original images using official masks. Experiments demonstrate strong generalization across generative models and datasets, achieving robust performance under varied test conditions. The approach integrates a combined Dice–Cross Entropy loss and low distortion augmentations for stable training. In addition, we introduce a reproducible recomposition protocol and highlight the network's ability to jointly learn from RGB and DNF representations.**

***Index Terms*—Forgery localization, Diffusion models, Noise features, Transformer segmentation, Trustworthy AI**

I. Introduction

The rapid growth of generative artificial intelligence has created both opportunities and risks, particularly in the trustworthiness of visual media. Modern latent diffusion-based generators such as *Stable Diffusion*, *OpenJourney*, and *Kandinsky* [1]–[3] can generate incredibly realistic content. Although the detection of fully AI-generated images is now a well-established research area, the identification of partially manipulated images, through inpainting, splicing, or object replacement, remains less explored. Semi-synthetic content poses huge risk to journalism, forensics, and online moderation due to the lack of availability of detectors and filters for small generative alterations in the original images.

This is the author's accepted manuscript of a paper that has been accepted for publication in the *IEEE BigData 2025 AI-PublicSafety Workshop*. When published, the version of record will be available in IEEE Xplore at: https://ieeexplore.ieee.org/.

Recent research highlights the limitations of existing approaches. For example, the Semi-Truths [4] study demonstrates that global detectors often succeed in classifying entire images as real or synthetic but fail to reliably capture small edited regions in their classification. In practice, manipulations are not always "all or nothing." Tools such as Adobe Photoshop and text-guided diffusion editing allow users to alter specific regions of otherwise authentic photographs, creating forgeries that bypass detectors tuned for global judgments. These developments highlight the need for a segmentation-level detection method, rather than relying on image-level decisions.

Although there is a vast array of methodologies for detecting generated images, this paper aims to introduce a novel dataset and pipeline for segmentation of diffusion-based synthetic content on an otherwise real image. In this work, we first create a new composite AI-manipulated dataset of partially AI-altered images and then propose a novel segmentation pipeline that leverages a physically motivated prior, the noise distribution as obtained through DNF [5]. By re-noising an image through diffusion inversion, we reveal residual noise statistics that differ between real and synthetic regions. These DNFs, combined with RGB input, are processed by a Swin U-Net to generate pixel-level segmentation masks that highlight manipulated areas. The major contributions of this work are as follows:

1) We created a new dataset of composite AI-manipulated images in which a real image is altered by substituting a part of the original image with AI-generated content.
2) We propose a novel semantic segmentation pipeline using Swin U-Net that generates the segmentation mask for AI-manipulated regions using a RGB rendering of the AI-manipulated image and a noise prior of the same in the form of a DNF map.
3) We use a novel weighted loss function designed to ensure pixel level accuracy and alignment of the segmentation mask with the ground truth.
4) We conducted experiments in varied settings and show

results that demonstrate the efficacy of the proposed method and also clearly demonstrate the value of using a noise-based prior for the task of recognizing AI-generated content intermingled with original image content.

The rest of the paper is organized as follows: in Section II we discuss the previous work and introduce relevant ideas, in Section III we discuss the dataset generation, in Section IV we introduce our method and discuss the segmentation pipeline, in Section V we describe our experimental setup and finally in Section VI we describe our results and discuss the observations. We conclude the work in Section VII.

II. Related Work

In this section we discuss the current state of the art in the detection of AI-generated/manipulated images. We start with the detection of images generated using AI, the so called global AI-generated images.

A. Global AI-Generated Image Detection

Most existing approaches to AI-generated image detection operate at the global level, classifying entire images as real or synthetic. Diffusion Reconstruction Error (DIRE) [6] exploits diffusion inversion, showing that real images incur a higher reconstruction error than diffusion-generated ones. Spectral Reconstruction Similarity (SRS) [7] instead models the frequency spectrum of natural images and identifies generated content as out-of-distribution by masked spectral learning. Few-Shot Classification for AIGI Detection [8] frames the task as a few-shot problem, improving generalization across unseen generators. The original DNF paper was intended for global generation detection, and it performed incredibly well. Although effective for whole-image classification, these methods have not been adapted to segment partial edits, underscoring the need for segmentation-based approaches such as ours.

B. Forgery Localization and Manipulation Detection

Beyond global classification, a substantial body of work addresses pixel-level forgery localization. Convolutional neural networks (CNNs) have been widely applied to detect copy-move and splicing manipulations, with Li et al. [9] showing that local inconsistencies in noise and illumination provide strong signals for tampering. Complementary approaches combine hand-crafted preprocessing with learned features, such as Error Level Analysis (ELA) paired with CNN classifiers, which enhance sensitivity to subtle compression and editing artifacts [10]. Historically, methods such as PRNU [11] and Noiseprint [12] were utilized to detect camera fingerprints to verify the integrity of an image. Although effective for traditional manipulations, many of these methods struggle to generalize to modern diffusion-based edits, motivating the development of pipelines tailored to generative forgeries.

C. Transformer-Based Segmentation Backbones

Transformers [13] have become increasingly prominent in computer vision after the introduction of the Vision Transformer (ViT) [14], with several architectures adapted for dense prediction tasks such as segmentation. The Swin Transformer introduced a hierarchical design with shifted-window attention, enabling scalability to high-resolution images while maintaining efficiency [15]. Building on this foundation, Swin U-Net extends the architecture with an encoder–decoder structure, closely following the principles of classic U-Net [16], but replacing convolutional blocks with Transformer modules. This design preserves multi-scale context while providing precise localization, making it well suited for pixel-level forensics.

Other variants, such as TransUNet [17], have also demonstrated strong performance in medical and natural image segmentation, underscoring the versatility of attention-based backbones. In this work, we adopt Swin U-Net [18] as our segmentation backbone, modifying its input layer to accommodate the additional DNF channel and output binary masks for authentic versus manipulated regions.

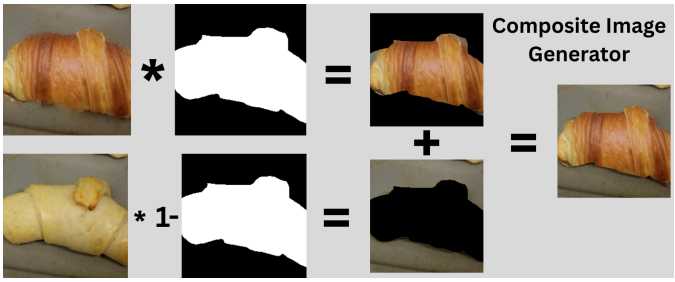

Fig. 1. The composite AI-manipulated dataset is created by splicing the area from an AI-generated image via a Semi-Truths mask and applying the spliced area to the original image.

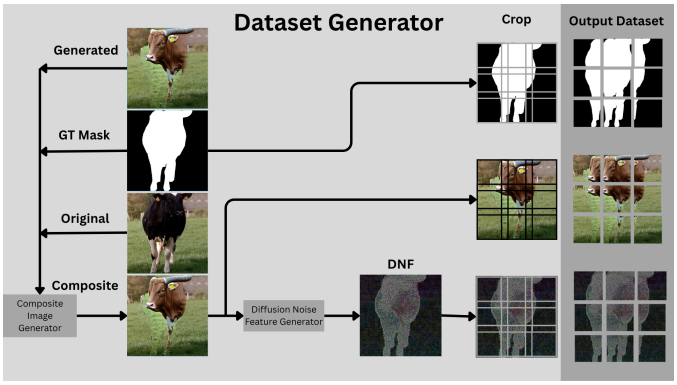

Fig. 2. The composite AI-manipulated dataset generation pipeline: combine original and generated images based on the ground truth mask into a composite image, process the composite image to produce the 224 individual samples. Each image patch is given a unique ID and treated as an independent sample.

III. Dataset

To evaluate our approach under realistic diffusion-based manipulations, we adapt the Semi-Truths Evalset, a benchmark

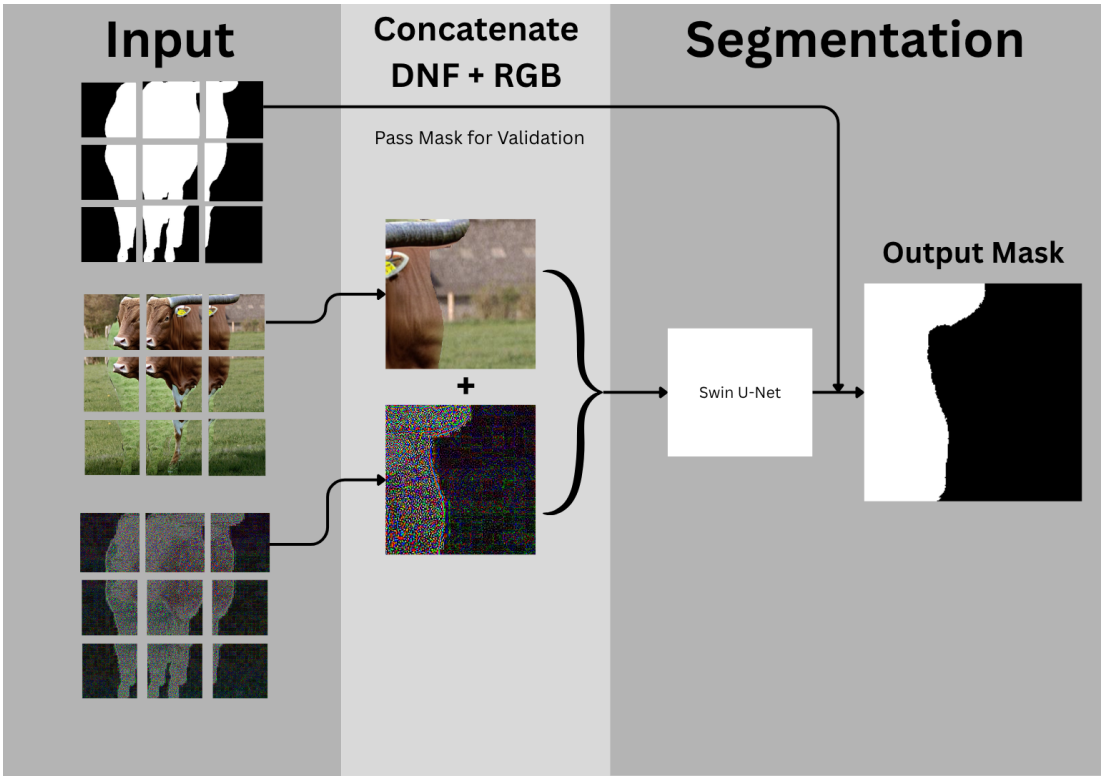

Fig. 3. The inference pipeline: the input is the ground truth mask, composite image, and DNF map. The pipeline uses the four channel image with a Swin U-Net which generates an output mask that is compared with the ground truth (GT) mask.

originally proposed for forgery detection. Although Semi-Truths promotes five different datasets, for simplicity, we use their processed OpenImages, CelebA-HQ, and ADE20k data [19]–[21]. The dataset contains three elements per sample: a "fake" image produced via diffusion inpainting or splicing, the corresponding real image, and an official binary mask of the altered region.

However, in its raw form, the "fakeimages " are globally processed and often misaligned with their originals. This makes direct pixel-level supervision unreliable, since small shifts or inconsistencies between the real/fake pair lead to noisy masks. To better align with pixel-level segmentation, we reconstruct the dataset by stitching the altered content from the fake image back onto the original image via the mask provided as shown in Fig. 1. The composite image is calculated directly using $G*M+O*(1-M)$ where G is the generated image, M is the mask, and O is the original image. The whole process produces aligned quadruplets consisting of:

1) the original image,
2) the mask defining the edited region,
3) the diffusion-altered content, and
4) the composite (semi-synthetic) image

This re-composition step ensures precise pixel-level difference between authentic and manipulated regions, which is critical for our segmentation process. The pipeline is implemented with ID matching across the official Semi-Truths archives and parallelized for efficiency, enabling reproducible generation of aligned data. This completed re-composition produces a new dataset we will refer to as Semi-Truths Composite.

Semi-Truths Composite images are a size of 512 $\times$ 512, which is incompatible with our pre-trained Swin Transformer backbone that expects 224 $\times$ 224 inputs. To resolve this mismatch, and avoid downsampling, we divide each 512 $\times$ 512 recomposed sample into nine overlapping images of size 224 $\times$ 224. Each image is treated as an independent training sample, inheriting the aligned real, recomposed, and mask triplets as shown in Fig. 2.

IV. Methodology

This work introduces a two-stage segmentation pipeline to localize AI-altered (manipulated) regions in images as visualized in Fig. 3. First, DNFs are computed via residual noise extraction from reverse diffusion steps to highlight residual noise statistics that differ between authentic and synthesized regions. Second, a Swin U-Net system processes the RGB image and its corresponding DNF together to produce a pixel-level segmentation mask. An overview of each component is provided below.

A. Diffusion Noise Feature

DNFs are residual noise maps extracted using a pre-trained diffusion model that utilizes a reverse diffusion (inversion) process. Because the backbone diffusion model is trained solely on real images, it encodes the statistics of natural sensor and processing noise. When an image is inverted through the

model, authentic regions yield noise consistent with this distribution, while synthesized regions reveal distinctive artifacts left by the generative process. This discrepancy provides a physically motivated cue for localizing manipulated content.

In this work, the DNFs are derived from the reverse diffusion process of a pre-trained denoising diffusion probabilistic model (DDPM) [22]. At each inversion step t, the model predicts the residual noise ϵ_t from the noisy input x_t:

$$\epsilon_t = \epsilon_\theta(x_t, t), \tag{1}$$

where ϵ_θ is the noise-prediction network of the diffusion model. In our implementation, we use the pre-trained diffusion model released by the DNF authors from [22]. Following the first-noise strategy, we select the residual from the earliest inversion step ϵ_1 as the DNF representation to be fed into our backbone along with the RGB representation of the target image. The first-noise strategy is chosen due to the later steps degrading the generative artifacts as it approaches natural noise. Hence, we compute:

$$\epsilon_1 = \epsilon_\theta(x, t = 1), \tag{2}$$

where x is the input mapped to timestep 1.

The DNF patterns are visually identifiable, as shown in (Fig. 4). This clear manifestation of the statistical differences between natural sensor noise and generative artifacts provides a strong prior for a segmentation model which can be exploited to generate accurate segmentation masks.

B. Swin Transformer U-Net

The segmentation backbone is a Swin Transformer U-Net, which extends the hierarchical Swin Transformer into an encoder–decoder framework for segmentation. Building on a pre-trained Swin-T backbone, the network follows the general design of the U-Net architecture, replacing convolutional blocks [23] with transformer stages that capture long-range dependencies through shifting-window self-attention.

The encoder progressively extracts high-dimensional representations, which are passed through a bottleneck composed of two Swin Transformer blocks. The decoder then hierarchically upsamples the feature maps while fusing skip connections from the encoder to recover fine spatial detail. This process is highlighted in Fig. 5.

In our implementation, the input layer is modified to accept a four-channel tensor, comprising the three RGB channels concatenated with the DNF map. The decoder produces a two-class segmentation mask, labeling each pixel as either authentic or altered. This integration enables the network to jointly utilize both semantic appearance cues from RGB and subtle statistical irregularities captured in DNF, improving its ability to localize AI-generated regions.

C. Loss Functions

To address class imbalance between large background regions and small manipulated regions, we employ a weighted combination of cross-entropy and Dice loss:

$$\mathcal{L} = \lambda_{CE}\,\mathcal{L}_{CE} + \lambda_{Dice}\,\mathcal{L}_{Dice}, \tag{3}$$

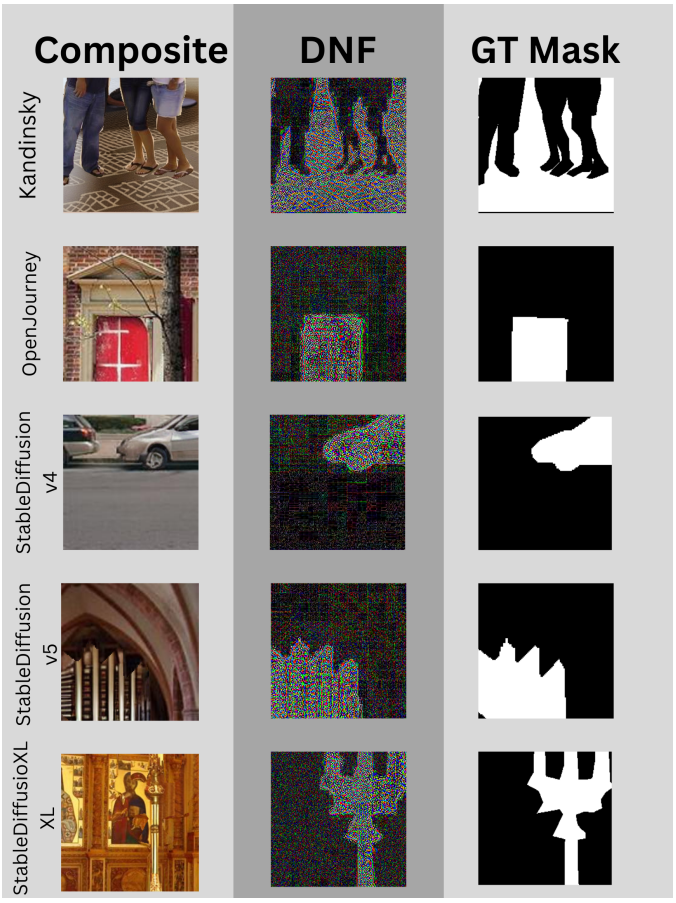

Fig. 4. Examples of ADE20k AI-manipulated composite images with their corresponding Semi-Truths-provided ground truth (GT) mask and their DNF maps.

with $(\lambda_{CE}, \lambda_{Dice}) = (0.3, 0.7)$ in our experiments. Note that in this work the choice of the weights $(\lambda_{CE}, \lambda_{Dice})$ is made empirically. The choice of the weights might be different for different datasets and different methods for generating the manipulated images. It might also be possible to select the weights in a data-driven manner but such an endeavor is left as a future exercise.

Our Cross-Entropy is consistent with the official PyTorch implementations [24]. The cross-entropy term measures pixel-level classification error between the predicted and ground-truth labels:

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log\hat{y}_{i,c}, \tag{4}$$

where N is the total number of pixels, C is the number of classes (here $C = 2$ for authentic vs. altered), $y_{i,c} \in \{0, 1\}$ denotes the one-hot encoded ground-truth class, and $\hat{y}_{i,c} \in [0, 1]$ is the predicted softmax probability for class c at pixel i [25].

The Dice loss [26] complements cross-entropy by optimizing region-level overlap between prediction and ground truth

TABLE I

PERFORMANCE (F1↑) BY DATASET AND GENERATOR. BOLD INDICATES BEST VALUE AMONG RGB BASELINE, RGB+NOISEPRINT, AND RGB+DNF.

Dataset	Method	Kandinsky 2.2	OpenJourney	SD v4	SD v5	SD XL	Mean	Δ vs RGB
OpenImages	RGB	**94.18**	90.97	91.03	94.61	91.32	92.42	—
	RGB+Noiseprint	90.35	92.59	93.05	93.01	90.10	91.82	-0.60
	RGB+DNF	94.11	**93.34**	**94.23**	**94.98**	**92.91**	**93.91**	**+1.49**
ADE20k	RGB	81.66	89.25	85.42	91.51	85.61	86.69	—
	RGB+Noiseprint	89.46	90.47	**93.61**	84.82	87.68	89.21	+2.52
	RGB+DNF	**95.17**	**93.71**	92.66	**93.91**	**94.12**	**93.91**	**+7.22**
CelebA-HQ	RGB	88.52	82.35	82.28	92.81	89.65	87.12	—
	RGB+Noiseprint	89.50	86.47	**89.86**	87.57	85.87	87.85	+0.73
	RGB+DNF	**94.49**	**92.50**	89.27	**94.02**	**93.77**	**92.81**	**+5.69**

TABLE II

PERFORMANCE (IOU↑) BY DATASET AND GENERATOR. BOLD INDICATES BEST VALUE AMONG RGB, RGB+NOISEPRINT, AND RGB+DNF.

Dataset	Method	Kandinsky 2.2	OpenJourney	SD v4	SD v5	SD XL	Mean	Δ vs RGB
OpenImages	RGB	**92.15**	88.42	88.58	92.80	89.11	90.21	—
	RGB+Noiseprint	87.43	90.24	90.90	90.80	87.36	89.35	-0.86
	RGB+DNF	91.62	**91.01**	**92.26**	**92.97**	**90.46**	**91.66**	**+1.45**
ADE20k	RGB	77.88	86.69	81.99	88.77	82.07	83.48	—
	RGB+Noiseprint	86.63	87.73	**91.01**	81.56	84.56	86.30	+2.82
	RGB+DNF	**93.04**	**91.47**	90.34	**91.60**	**91.65**	**91.62**	**+8.14**
CelebA-HQ	RGB	85.02	78.44	79.30	90.75	86.81	84.06	—
	RGB+Noiseprint	86.33	83.55	**86.73**	84.57	85.87	85.41	+1.35
	RGB+DNF	**91.84**	**89.89**	86.01	**91.55**	**91.24**	**90.11**	**+6.04**

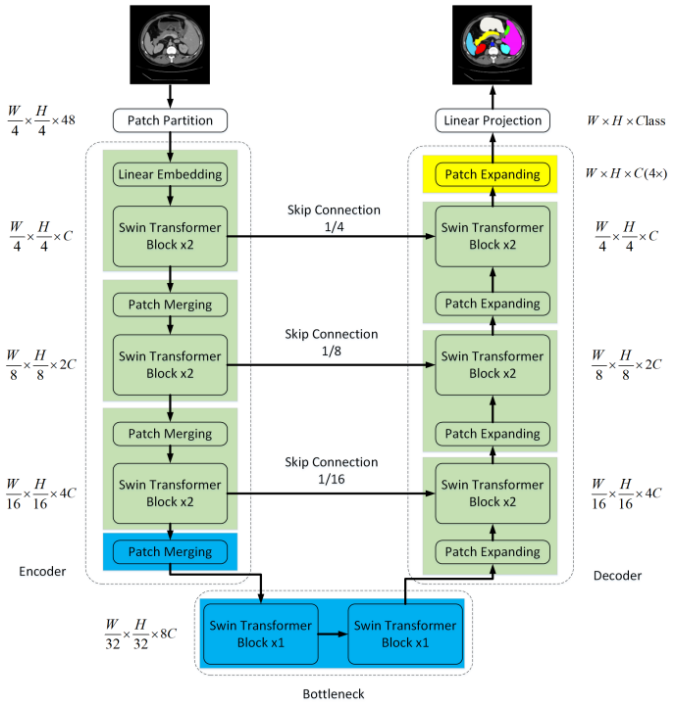

Fig. 5. The Swin U-Net architecture [18] utilized highlights three encoding layers, a bottleneck, and three decoding layers. Each encoder is connected to a decoder via a skip connection at matching shapes.

as proposed in [26]:

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\hat{y}_{i,c} + \epsilon}{\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}^2 + \sum_{i=1}^{N}\sum_{c=1}^{C} \hat{y}_{i,c}^2 + \epsilon}, \quad (5)$$

where ϵ is a small smoothing constant (10^{-5} in implementation) to prevent division by zero.

This composite loss balances global pixel-wise accuracy (via cross-entropy) with region-level consistency (via Dice), making it particularly effective for segmenting small manipulated regions within large backgrounds [27].

D. Training Protocol

All models are trained using the stochastic gradient descent (SGD) [28] with momentum 0.9, optimizer weight decay of 1×10^{-4}, and an initial learning rate set as 0.05. The learning rate is polynomially decayed at each iteration according to $(1 - \frac{iter}{max\_iter})^{0.9}$. Training uses a combined loss function of weighted cross-entropy (0.3) and Dice loss (0.7) to balance pixel-level accuracy with region-level overlap as noted before.

Data augmentation is deliberately kept to a minimal to preserve subtle generative artifacts: random horizontal/vertical flips and rotations are applied, but no blurring or strong color jittering is used. Each 512 × 512 composite image is cropped into overlapping 224 × 224 image patches, providing inputs compatible with the pretrained Swin backbone. Data is drawn from the RGB+DNF Dataset, where each sample includes three RGB channels concatenated with a DNF channel.

V. Experiment

A. Implementation Details

The Swin U-Net with DNF pipeline is implemented in Python 3.12 using PyTorch 2.8. To improve data diversity, we apply standard augmentations including random flips and rotations but avoid more complex methods for reasons noted in Section IV-D. All input images are 224×224, with a channel size of 4 (RGB + DNF). Training is performed on an NVIDIA RTX 5090 GPU with 32 GB VRAM. In the Swin U-Net, the Swin Transformers are initialized with pretrained weights for Swin-T. The DNF channel is initialized with the averaged RGB weights from the pretrained Swin-T. Optimization is carried out using stochastic gradient descent (SGD) using a batch size of 64 throughout the training and parameters as noted in Section IV-D. All experiments were performed over 100 epochs.

B. Metrics

We report a suite of metrics to evaluate segmentation quality. Accuracy, while widely used, can be misleading because real pixels typically outnumber generated pixels by a large number; therefore, we also include F1 [29] and Intersection-over-Union (IoU) [30] to better capture class imbalance and boundary performance. The F1 score balances false positives and false negatives. IoU (also called the Jaccard index) is a stricter overlap measure that penalizes both types of error. Formally, given true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN), these metrics are defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{6}$$

$$\text{F1} = \frac{2TP}{2TP + FP + FN} \tag{7}$$

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{8}$$

In our evaluation, the metrics are computed at the pixel level across the test set.

C. Experiment Results

The primary objective of our experiments is to evaluate whether concatenating DNF with RGB input improves the generative-content segmentation performance of the Swin U-Net compared to an RGB-only baseline. Table I and Table II reports F1 and IoU results respectively across five diffusion generators in our Semi-Truths Composite dataset.

To contextualize the effect of DNFs, we additionally evaluate a noiseprint-based variant of our pipeline. Following the method of [12], we extract a model-generated noiseprint for each image and provide it as a fourth input channel, mirroring the role played by DNFs in our architecture. This controlled substitution enables a direct comparison between our diffusion-derived residual signal and a classical noise-fingerprint baseline.

Across nearly all tested generators, incorporating the DNF channel improves performance relative to the RGB-only baseline and the noiseprint-based variant. For example, within the ADE20k dataset and the *Stable Diffusion XL*, the F1 score rises from 85.61 on the RGB baseline to 94.12 on our method. On another dataset, CelebA-HQ, the *OpenJourney* generator reports an increase from 82.35 to 92.50. Similar gains are observed for IoU and accuracy, indicating consistent benefits from incorporating DNFs over either RGB-only inputs or noiseprint-based inclusion.

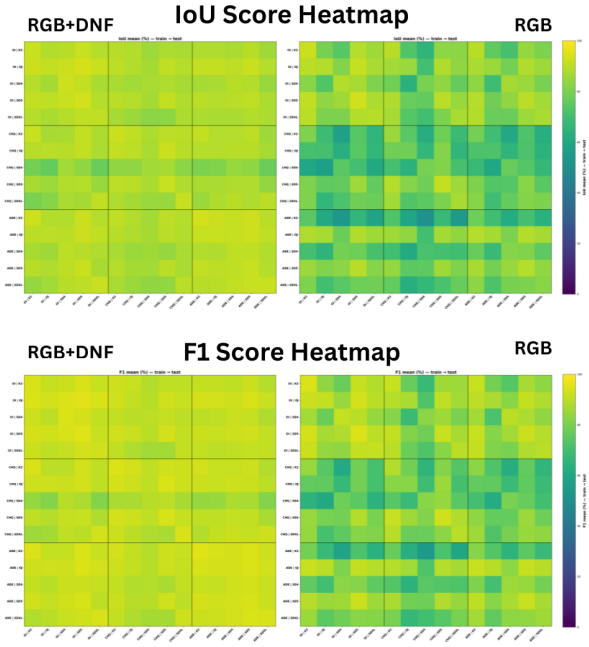

Fig. 6. F1 and IoU Score Heatmap comparison between our RGB + DNF and a RGB baseline where the y-axis is the train dataset, and the x-axis is the test dataset.

D. Generalization

To assess the generalization of the proposed method, we perform a cross-dataset generalization test in which each model trained on a single dataset is evaluated on every other dataset. Specifically, we train separate Swin U-Net models for each dataset (OpenImages, ADE20k, and CelebA-HQ) and its generators (*Stable Diffusion 4, 5, and XL, OpenJourney*, and *Kandinsky 2.2*) and test them across all remaining domains, including their own. This setup examines how well the learned representations transfer to unseen image distributions and generator-specific noise characteristics.

Overall, the RGB+DNF configuration consistently exhibits higher F1 and IoU scores in out-of-domain evaluations as shown in heatmaps Fig. 6. In particular, models trained on one dataset utilizing DNFs retain stronger discriminative ability when transferred to another, suggesting that the inclusion of DNFs improves generalization by learning generator noise priors rather than dataset-specific appearance cues.

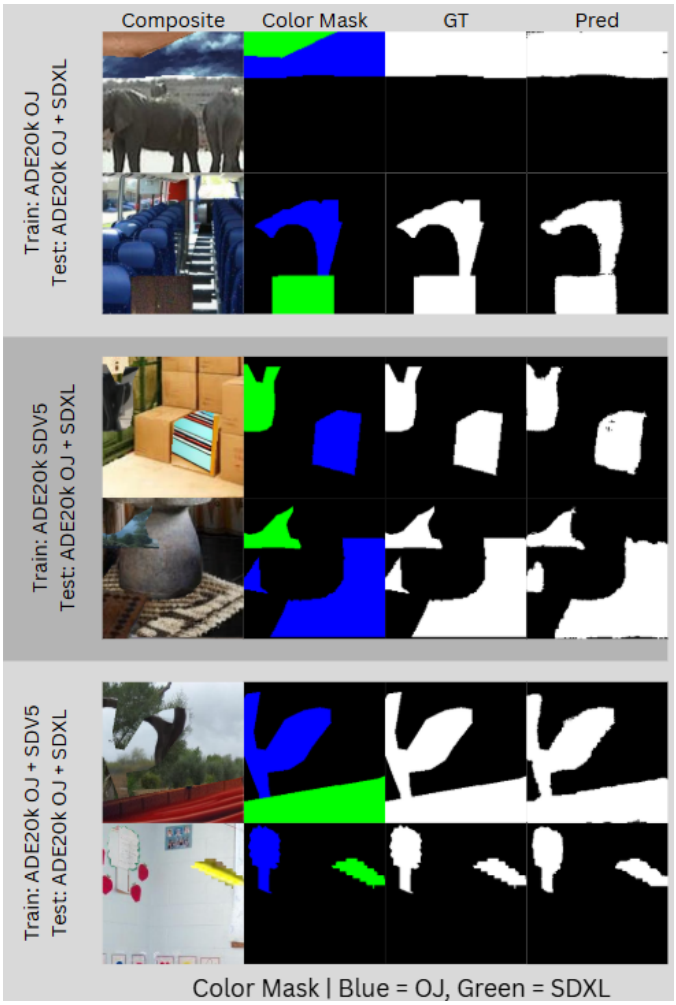

Fig. 7. Examples from robustness evaluation illustrating the results on a multi-source composite dataset. The ADE20k base image and mask are first combined with OpenJourney- and Stable Diffusion XL–generated regions. The resulting mixed-edit dataset is then tested on models trained individually on ADE20k-OpenJourney and ADE20k–Stable Diffusion XL, followed by retraining and evaluation on the newly conjoined ADE20k (OpenJourney + Stable Diffusion XL) dataset.

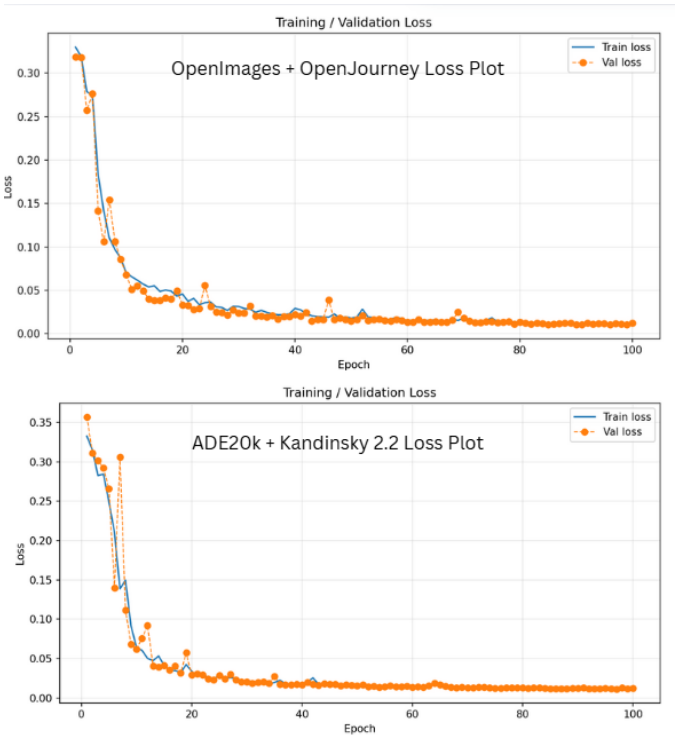

Fig. 8. Example loss plots to highlight smooth convergence. This plot displays the training and validation loss for the OpenImages dataset and the OpenJourney generator as well as the ADE20k dataset and the Kandinsky 2.2 generator.

VI. Discussion

A. Key Findings

The experiments demonstrate that concatenating DNF with RGB inputs consistently improves segmentation performance across multiple datasets and diffusion generators. Gains are observed not only in F1 score but also in IoU and accuracy. The greatest relative improvements are seen in more advanced models such as *Stable Diffusion XL*, where subtle manipulations are harder to capture with RGB alone. In contrast, simpler generators such as *Kandinsky 2.2* show smaller improvements or comparable results, suggesting that DNFs provide the greatest benefit when manipulations are visually subtle or statistically well hidden.

To further contextualize the contribution of DNFs, we introduced a noiseprint-based variant of our pipeline in which a classical model-generated noiseprint replaces the DNF in the fourth input channel, enabling a direct comparison between diffusion-derived residuals and a traditional noise-fingerprint prior. Across nearly all datasets and generators, DNFs substantially outperform noiseprint, with mean improvements of +1.49 F1 / +1.45 IoU on OpenImages, +7.22 F1 / +8.14 IoU on ADE20k, and +5.69 F1 / +6.04 IoU on CelebA-HQ (Tables I–II), indicating that DNFs capture a richer, residual signal than the more generic sensor-noise patterns recovered by noiseprint. A single exception arises on ADE20k with *Stable Diffusion v4*, where noiseprint marginally exceeds DNF performance. We hypothesize that this effect arises because Stable Diffusion v4 produces residual noise patterns on ADE20k images that more closely match the type of altered-region signatures that noiseprint is designed to detect. However, the difference is small in magnitude and does not appear in any other dataset–generator combination, underscoring that DNFs provide a more robust and discriminative residual representation for diffusion-based manipulation localization.

Furthermore, generalization studies (Fig. 6) reveal that DNFs contribute substantially to the model's ability to generalize beyond its training domain. When each model trained on one dataset and generator was tested on entirely different datasets and generators, the RGB + DNF configuration maintained high segmentation performance with minimal degradation. This consistency across out-of-domain evaluations indicates that DNFs capture generator-independent residual statistics that persist across varying image distributions and

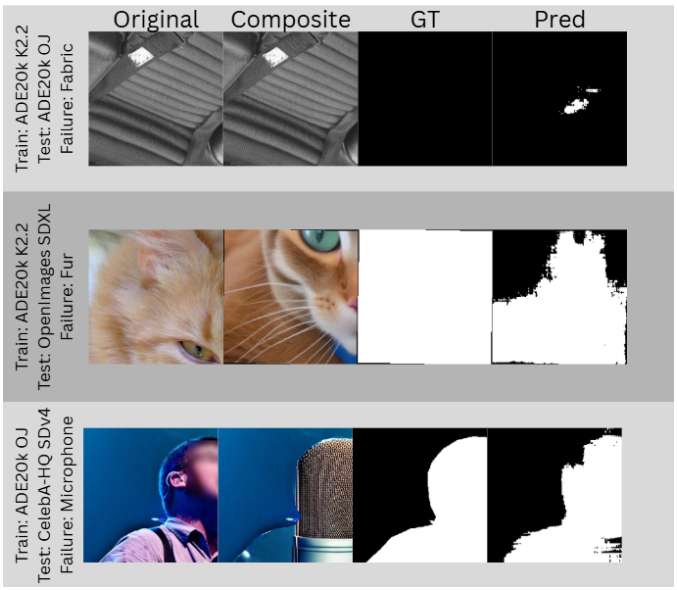

Fig. 9. Examples of original, composite, ground truth (GT) masks, and predicted (Pred) masks to highlight the failure cases due to noisy image qualities such as fur and fabric.

TABLE III
PERFORMANCE ON MULTI-GENERATOR EDITS BY A SINGLE- AND MULTI-GENERATOR TRAINED MODEL.

Train Model	F1	IoU	Accuracy
OpenJourney (OJ)	92.24	88.80	98.51
Stable Diffusion XL (SDXL)	91.80	88.03	98.46
OJ + SDXL (mixed)	93.14	90.12	99.06

semantic contexts. In effect, DNFs allow the model to learn a generalized representation of diffusion-based alterations rather than overfitting to dataset-specific color or texture features, as shown by the smooth loss curve for both training and validation Fig. 8. These results highlight DNFs as a robust signal representation that enhances the Swin U-Net's capacity to localize manipulations under diverse conditions.

B. Robustness

Beyond cross-domain generalization, we evaluate robustness under multi-source edits by sequentially applying OpenJourney and Stable Diffusion XL manipulations to the same base image (Fig. 7). As shown in Table III, the mixed-edit configuration achieves an F1 of 93.14, IoU of 90.12, and accuracy of 99.06, slightly higher than either single-source model trained on OpenJourney (F1 92.24) or Stable Diffusion XL (F1 91.80). This improvement suggests that the inclusion of heterogeneous generator noise patterns reinforces, rather than degrades, the learned DNF representation. Notably, the single-generator models also maintain strong performance when evaluated on the mixed-generator dataset, indicating that the learned representations remain stable even when exposed to unseen combinations of diffusion noise. Combined with cross-dataset evaluations (Fig. 6), these results demonstrate that DNFs capture generator-agnostic cues that generalize across both single and composite manipulations, supporting robust forgery localization.

C. Failure Cases and Limitations

Despite the strong overall performance, several limitations were observed. The model occasionally misclassifies highly textured or patterned regions where the intrinsic noise characteristics resemble those of diffusion artifacts, such as fur, fabric, or microphone meshes. This confusion probably arises from the similarity between the granularity of the natural texture and the synthetic residual noise patterns, as shown in Fig. 9. Furthermore, strong compression, resizing, or filtering can distort the DNF signal, leading to reduced boundary precision in the predicted masks. The computational overhead of extracting DNFs through diffusion inversion also remains a light bottleneck. We observe that 1000 DNFs are generated in 143.05 seconds on average, which could limit scalability for large-scale or real-time applications. These observations underscore the need for adaptive inversion strategies, compression-invariant noise modeling, and more lightweight approximations of DNFs in future research.

D. Broader Applicability

By introducing a visually interpretable, generator-agnostic signal representation, this work contributes toward establishing reliable and generalizable tools for forensic detection of AI-manipulated media. The proposed framework demonstrates how noise-based channels can complement semantic segmentation, enabling a more robust understanding of how generative image noise cues differ from natural images.

Beyond research, the approach has practical relevance for digital forensics, journalism, and platform integrity, where localized identification of manipulated regions can provide verifiable evidence of authenticity. The reproducible nature of the pipeline, together with its adaptability to different segmentation backbones, also positions it as a foundation for future multimodal or cross-domain detection systems, spanning imagery, video, and even sensor data.

VII. CONCLUSION & FUTURE WORK

While our results demonstrate the effectiveness of DNFs combined with Swin Transformer-based segmentation, several avenues remain open for future research. First, broader evaluation across diverse, more rigorous, datasets is necessary to assess generalization beyond the OpenImages, ADE20k, and CelebA-HQ portions of the Semi-Truths Composite benchmark. Applying our pipeline to domains such as documents, natural landscapes, or medical imagery would reveal whether the benefits of DNFs extend across other real-world content types. Second, the computational overhead of DNF extraction could be reduced. Since each sample requires a forward pass through a pre-trained diffusion model, future work should explore lighter inversion strategies. Third, while our segmentation backbone is based on Swin U-Net, alternative architectures could be explored. Vision-language models or

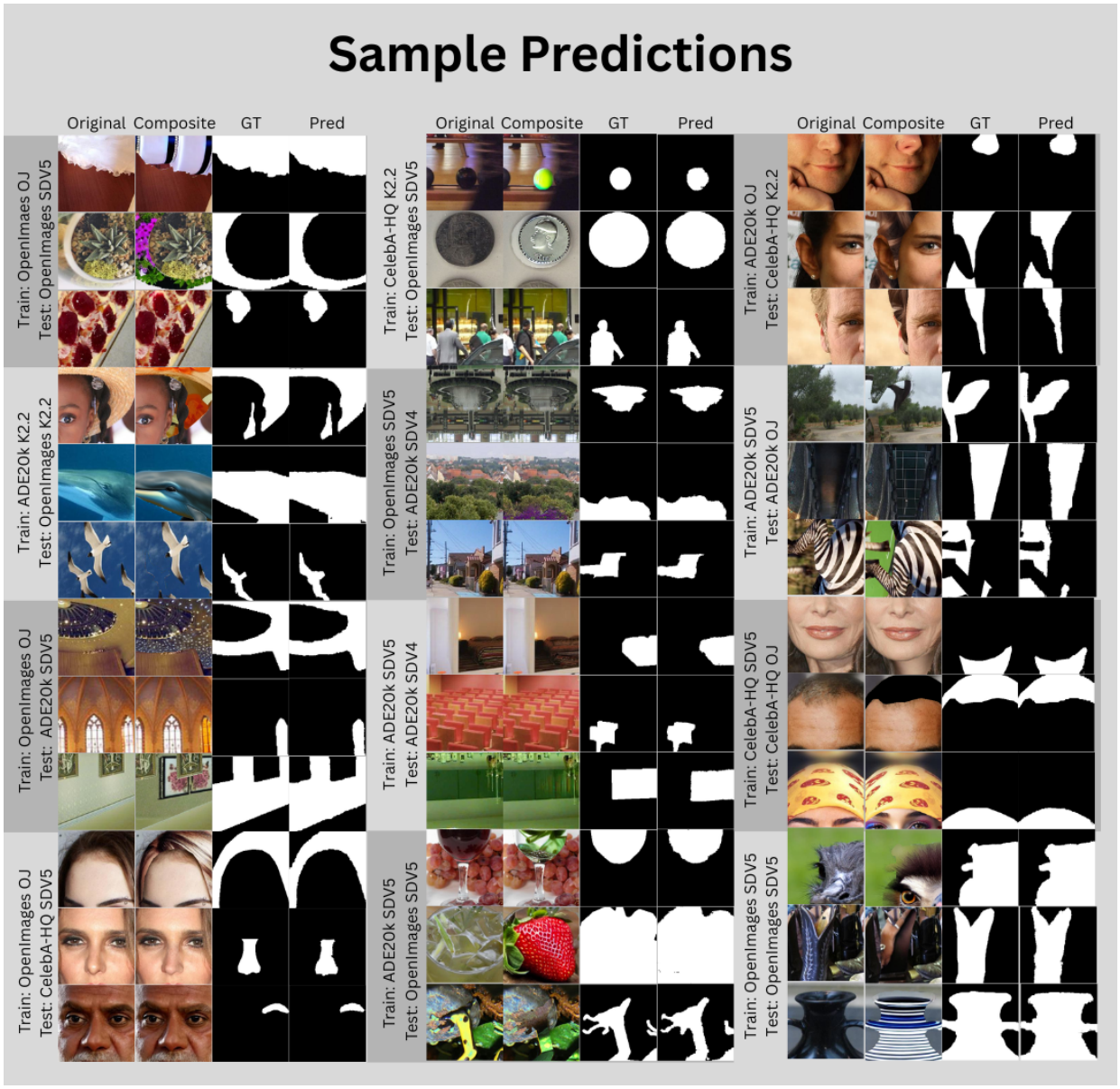

Fig. 10. Thirty six sample original images, composite images, ground truth (GT) masks, and the predicted (Pred) masks across various train and test datasets.

multi-modal transformers may enable cross-modal detection, while lighter backbones could support real-time forensic applications. Generalization studies across different transformer scales (Swin-T, Swin-S, Swin-B) would clarify the trade-off between accuracy and efficiency. Finally, real-world deployment scenarios demand robustness to post-processing. Future research should evaluate the resilience to compression, resizing, filtering, and adversarial manipulations, which are common in practical media pipelines. Extending this work to domains such as video could also uncover additional consistency cues across frames, offering new opportunities for localization of generative manipulations.

REFERENCES

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2022.

[2] PromptHero, "Openjourney: Stable diffusion fine-tuned on midjourney v4 style images." https://huggingface.co/prompthero/openjourney, 2022.

[3] A. Razzhigaev, A. Shakhmatov, A. Maltseva, V. Arkhipkin, I. Pavlov, I. Ryabov, A. Kuts, A. Panchenko, A. Kuznetsov, and D. Dimitrov, "Kandinsky: An improved text-to-image synthesis with image prior and latent diffusion," *arXiv preprint arXiv:2310.03502*, 2023.

[4] A. Pal, J. Kruk, M. Phute, M. Bhattaram, D. Yang, D. H. Chau, and J. Hoffman, "Semi-truths: A large-scale dataset of ai-augmented images for evaluating robustness of ai-generated image detectors," 2024.

[5] Y. Zhang and X. Xu, "Diffusion noise feature: Accurate and fast generated image detection," 2025.

[6] Z. Wang, J. Bao, W. Zhou, W. Wang, H. Hu, H. Chen, and H. Li, "Dire for diffusion-generated image detection," *arXiv preprint arXiv:2303.09295*, 2023.

[7] D. Karageorgiou, S. Papadopoulos, I. Kompatsiaris, and E. Gavves, "Any-resolution ai-generated image detection by spectral learning,"

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2025.

[8] S. Wu, J. Liu, J. Li, and Y. Wang, "Few-shot learner generalizes across ai-generated image detection," 2025.

[9] J. Li, X. Li, and B. Chen, "Image copy-move forgery detection and localization based on super-bpd segmentation and dcnn," *Scientific Reports*, vol. 12, no. 1, p. 15644, 2022.

[10] T. Wang, M. Liu, W. Cao, and K. P. Chow, "Deepfake noise investigation and detection," *Forensic Science International: Digital Investigation*, vol. 42, p. 301395, 2022. Proceedings of the Twenty-Second Annual DFRWS USA.

[11] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006.

[12] D. Cozzolino and L. Verdoliva, "Noiseprint: A cnn-based camera model fingerprint," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 144–159, 2020.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 6000–6010, Curran Associates Inc., 2017.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[15] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, eds.), (Cham), pp. 234–241, Springer International Publishing, 2015.

[17] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[18] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," in *Proceedings of the European Conference on Computer Vision Workshops(ECCVW)*, 2022.

[19] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *IJCV*, 2020.

[20] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," 2018.

[21] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5122–5130, 2017.

[22] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *arXiv preprint arxiv:2006.11239*, 2020.

[23] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems* (D. Touretzky, ed.), vol. 2, Morgan-Kaufmann, 1989.

[24] PyTorch Developers, *torch.nn.CrossEntropyLoss — PyTorch Documentation*. PyTorch, 2025. Accessed: 2025-10-26.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[26] F. Milletarì, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, 2016.

[27] S. Taghanaki, Y. Zheng, S. K. Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu, and G. Hamarneh, "Combo loss: Handling input and output imbalance in multi-organ segmentation," *Computerized Medical Imaging and Graphics*, vol. 75, 05 2019.

[28] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[29] C. J. Van Rijsbergen, *Information Retrieval*. London, UK: Butterworth-Heinemann, 2nd ed., 1979.

[30] P. Jaccard, "Étude comparative de la distribution florale dans une portion des alpes et des jura," *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, pp. 547–579, 1901.