SAGI: Semantically Aligned and Uncertainty Guided AI Image Inpainting

Supplementary Material

1. Implementation details

This section provides additional implementation details of our approach to ensure reproducibility. Code is available on https://github.com/mever-team/SAGI.

1.1. Source of Authentic Images

Since RAISE [5] contains RAW images, we processed these images before using them for inpainting experiments. We utilized the RAISE dataset as described in [8].

1.2. Dataset Splits

As shown in Table 1, we structure our dataset to evaluate both in-domain performance and generalization to new data. For in-domain evaluation, we use COCO (60,000 randomly selected training images and nearly all 5,000 validation images for validation and testing) and RAISE (7,735 images processed with Φ_{seg}, yielding 25,674 image-mask-model combinations through 1-7 masks or prompts per image, with derived images kept in the same split, as each image was inpainted up to 4 times only in this dataset). To test generalization, we create an out-of-domain testing split using OpenImages [3]—a dataset not used during training—comprising 6,000 randomly selected test images. This split uses a different language model Θ_{llm} (Claude) than COCO and RAISE (ChatGPT), providing a way to evaluate how well models perform on both new data and different prompting approaches. Throughout our experiments, we refer to the COCO and RAISE test splits as in-domain and the OpenImages test split as out-of-domain.

	Training	Validation	Testing
COCO [9]	59,708 (75%)	1,950 (31%)	2,922 (29%)
RAISE [5]	19,741 (25%)	4,262 (69%)	1,671 (16%)
OpenImages [3]	*N/A*	*N/A*	5,585 (55%)
Inpainted	79,449	6,212	10,178
Authentic	79,449	6,212	9,071

Table 1. Overview of dataset splits across COCO, RAISE, and OpenImages. The table shows the number of images in each split. The total number of images, including authentic and inpainted versions, is provided. Percentages represent the distribution of each dataset within the total split for inpainted images.

1.3. SAOR configuration

The specific API endpoints used in our implementation were gpt-3.5-turbo [12] (as of June 2024) and claude-3-5-sonnet-20240620 [2]. The system prompt used for the LLMs in SAOR was configured as shown in Fig. 1. For double inpainting cases, where two objects needed to be sequentially modified, we used an adapted system prompt to select a second object and generate a prompt as shown in Fig. 1. For images designated for object removal, we used a simplified system prompt focused solely on object selection that is shown in Fig. 1.

All LLM interactions were configured with hyperparameters including a temperature of 1.2 to encourage creative variations in the generated prompts, a top-p (nucleus sampling threshold) of 0.8, and a maximum token limit of 40 for prompt length. Some prompts from initial experiments, conducted without the maximum token restriction, were retained in our final dataset. The prefix "Inpaint the masked area with..." was included in the system prompts to maintain a consistent format in the LLMs' responses but was omitted from the actual saved prompts to avoid potential misinterpretation by diffusion models.

1.4. Inpainting Pipelines Configuration

The text-guided inpainting models support Stable Diffusion [15] by default, along with certain community versions. Specifically, HD-Painter [11] supports Stable Diffusion v1.5[1], Stable Diffusion v2[2], and DreamShaper v8[3]. BrushNet [7] supports Stable Diffusion v1.5, Stable Diffusion XL[4], DreamShaper v8, Realistic Vision[5], epiCRealism[6], and JuggernautXL[7]. PowerPaint [20] combines Realistic Vision and BrushNet, while ControlNetInpaint [18] supports Stable Diffusion v1.5. Inpaint-Anything [17] supports Stable Diffusion v2.

Each inpainting pipeline received an equal number of images for processing, with Remove-Anything being treated as a separate pipeline, and the settings for each pipeline, such as diffusion models and post-processing tech-

[1] https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5

[2] https://huggingface.co/stabilityai/stable-diffusion-2

[3] https://civitai.com/models/4384/dreamshaper

[4] https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0

[5] https://civitai.com/models/4201/realistic-vision-v60-b1

[6] https://civitai.com/models/25694/epicrealism

[7] https://civitai.com/models/133005/juggernaut-xl

LLM System Prompt (1st Inpainting)

You write prompts for text-to-image image inpainting models (AI-inpainting). In these models, you give an image, a mask of an area that will be inpainted, and a text prompt to tell the model what to inpaint the masked area with. You will be given a caption of the original image (the whole image) to understand the context and a list of objects. Then you choose an object, **THAT EXISTS IN THE LIST GIVEN TO YOU**. You need to generate a suitable prompt to alter the masked area of the image that covers the object you chose.
Remember to make a prompt that alters the image. If you decide to replace the said object, replace it with something that makes sense given the object that is to be replaced and the caption. Also, do not mention the original object in the prompt unless you want to replace the said object with one of the same class. Generate the prompt like this:
Object: {object on the original list}
Prompt: Inpaint the masked area with...

LLM System Prompt (2nd Inpainting)

You write prompts for text-to-image image inpainting models (AI-inpainting). In these models, you give an image, a mask of an area that will be inpainted, and a text prompt to tell the model what to inpaint the masked area with. An object has already been replaced in the image, and we need to generate a **DIFFERENT** prompt for a second object.
You will be given a caption of the image to understand the context, the class of the 1st object, and the prompt of the 1st object. You will then **SELECT A 2ND OBJECT** from the image that is to be inpainted. You need to generate a suitable prompt to alter the masked area of the image that covers the 2nd object.
Remember to make a prompt that alters the image. If you decide to replace the said object, replace it with something that makes sense given the object that is to be replaced and the caption. Also, do not mention the original object in the prompt unless you want to replace the said object with one of the same class. Generate the prompt like this:
Object: {name of the 2nd object}
Prompt: Inpaint the masked area with...

LLM System Prompt (Removal)

You will be given a list of objects that exist in an image. You must choose an object to be removed with inpainting methods. Choose an object that makes sense.
Answer like this:
Object: {object in the list}

Figure 1. System Prompts for selecting objects and generating prompts for inpainting and removal. The first prompt is for the 1st inpainting, the second for the 2nd inpainting, and the third for object removal.

niques, were distributed uniformly. Despite efforts to maintain uniformity, small discrepancies occurred due to constraints such as excluding NSFW images flagged by Stable Diffusion.

1.5. UGDA Configuration

The Uncertainty-Guided Deceptiveness Assessment (UGDA) was implemented using the chatgpt-4o-latest [13] API endpoint (as of October 2024). We conducted preliminary experiments with various prompt formulations and found that allowing the model to explain its reasoning before providing a verdict led to more reliable assessments. The final implementation uses a two-stage process with carefully engineered prompts. The initial realism check uses the system prompt shown in Fig. 2. For images passing the initial check, a comparative assessment is performed using the system prompt shown in Fig. 2.

The prompt design choices were made based on empirical observations from a small validation set. Through our experiments, we found that requiring explanations before verdicts significantly improved assessment quality. Additionally, explicit mention of common artifacts (e.g., unusual textures, biological anomalies) helped focus the model's attention on relevant features.

The VLM was configured with conservative hyperparameters to ensure consistent responses, using a temperature of 0.1 to minimize response variability, a top-p of 1.0 with no nucleus sampling, and a maximum token limit of 2048 to allow for detailed explanations.

1.6. Detailed Classification Process of UGDA

Here we present the classification process of the second stage of UGDA in more detail:

- Case 1: The VLM identifies I_i in one order and $\hat{I}_i$ in the other, indicating order-dependent bias
- Case 2: The VLM consistently identifies $\hat{I}_i$ as more realistic ($s_1 = s_2 = \hat{I}_i$)
- Case 3: The VLM responds with "both" in one case and chooses $\hat{I}_i$ in the other
- Case 4: The VLM consistently responds that both images

VLM System Prompt (First Realism Check)
You are an expert in analyzing images for signs of modification by AI models and assessing their realism. Your task is to examine an image and determine if it is realistic. In addition, provide a brief explanation highlighting the key elements that informed your judgment, such as unusual textures, unnatural elements, unusual biological phenomena (animals with multiple limbs, heads), unusual arrangement of objects (weirdly clustered objects), or any visible artifacts. **# Output Format** **Assessment:** (your explanation goes here) **Verdict:** (final verdict, answer "Yes, it is realistic" or "No, it is not realistic")

VLM System Prompt (Second Realism Check)
You are an expert in analyzing images for signs of modification by AI models and assessing their realism. Your task is to examine two images and determine which one is more realistic. In addition, provide a brief explanation highlighting the key elements that informed your judgment, such as unusual textures, unnatural elements, unusual biological phenomena (animals with multiple limbs, heads), unusual arrangement of objects (weirdly clustered objects), or any visible artifacts. **# Output Format** **Assessment:** (your explanation goes here) **Verdict:** (final verdict, answer "First is more realistic" or "Second is more realistic" or "Both look realistic")

Figure 2. VLM system prompts for realism checks. The first prompt is for assessing the realism of a single image, and the second prompt is for comparing the realism of two images, original and inpainted.

are equally realistic ($s_1 = s_2 =$ both)

In all other response combinations, $\hat{I}_i$ is classified as *non-deceiving*. This classification scheme captures cases where the VLM either consistently prefers the inpainted image or shows uncertainty in its assessment, all of which indicate potential deceptiveness in the synthetic content. Case 1 indicates model uncertainty manifested through order sensitivity, Case 2 represents clear preference for synthetic content, Case 3 captures uncertainty biased toward synthetic content, and Case 4 reflects complete inability to distinguish between real and synthetic content. These patterns suggest varying degrees of image deceptiveness that warrant classification as deceiving.

1.7. Human Benchmark

The application implementing the human benchmark was developed using Gradio [1]. At the start of the demo, participants were provided with clear instructions on how to proceed. They were asked to evaluate whether an image had been inpainted and to draw bounding boxes around the areas they believed to be inpainted. Additionally, participants were asked to complete a short demographics questionnaire before beginning the task. The questions included are shown in Tab. 2.

2. Unmasked Area Preservation

Fidelity metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Peak Signal-to-Noise Ratio (PSNR), and Learned Perceptual Image Patch Similarity (LPIPS) [19] assess the preservation of the non-inpainted area. Fidelity metrics are most meaningful for FR images, whereas for SP images, where the compared areas are nearly identical, they provide limited insight. The results are presented in Tab. 3. When comparing our dataset with existing alternatives, our SAGI-D significantly outperforms TGIF across all FR image fidelity metrics. We achieve a PSNR of 25.79 compared to TGIF's 14.41, with substantially better LPIPS (44.24 vs 289.55), MSE (5.08 vs 60.43), and MAE (41.16 vs 173.97). These improvements indicate that our inpainting approach better preserves the original image context while implementing the intended modifications. CocoGlide is not included in Tab. 3 since it contains only SP images.

3. Localization and Detection Results

In this section, we present extended results on localization and detection, studying various cases for forensic models PSCC-Net [10], CAT-Net [8], TruFor [6], and MMFusion (MMFus) [16].

Since mean IoU and detection Accuracy require a threshold, we also report AUC metrics in Table 5 as they are threshold-agnostic. We calculate AUC at both pixel level (localization) and image level (detection). For localization AUC, we resize and flatten all localization maps and their ground truths into two vectors for ROC computation in each group. Note that detection AUC cannot be calculated for SP and FR sets, as they contain only forged images. The AUC metrics further confirm that retraining improves performance significantly. TruFor's localization AUC increases from 68.9% to 99.5% for in-domain and 79.9% to 99.6% for out-of-domain testing. Similarly, CAT-Net shows strong in-domain gains (60.0% to 95.6%) but smaller out-of-domain improvement (51.7% to 90.8%). Domain generalization varies across models. While retrained CAT-Net

Gender	**Age Range**	**Highest Education Completed**
☐ Male ☐ Female ☐ Other ☐ Prefer not to say	☐ Under 18 ☐ 18-24 ☐ 25-34 ☐ 35-44 ☐ 45-54 ☐ 55-64 ☐ 65+ ☐ Prefer not to say	☐ EQF 1-4 (Primary/Upper Secondary) ☐ EQF 5 (Post-Secondary Diploma) ☐ EQF 6 (Bachelor's Degree) ☐ EQF 7 (Master's Degree) ☐ EQF 8 (Doctorate) ☐ Prefer not to say
Current Education Status	**Familiarity with AI-Generated Images**	**Knowledge of Digital Photography**
☐ EQF 1-4 (Primary/Upper Secondary) ☐ EQF 5 (Post-Secondary Diploma) ☐ EQF 6 (Bachelor's Degree) ☐ EQF 7 (Master's Degree) ☐ EQF 8 (Doctorate) ☐ Not currently studying ☐ Prefer not to say	☐ Very familiar ☐ Somewhat familiar ☐ Slightly familiar ☐ Not familiar ☐ Prefer not to say	☐ Professional level ☐ Advanced ☐ Intermediate ☐ Basic ☐ No experience ☐ Prefer not to say

Table 2. Demographic and background questionnaire.

Dataset	**PSNR↑**	**LPIPS↓**	**MSE↓**	**MAE↓**	**SSIM↑**
TGIF	14.4	289.6	60.4	174.0	0.53
Ours	**25.8**	**44.2**	**5.1**	**41.2**	**0.81**

Table 3. Comparison based on fidelity metrics for FR images. Top: object labels vs. Caption prompts vs LLM prompts. Bottom: our dataset vs. TGIF. LPIPS, MSE, and MAE values are $\times 10^3$. CocoGlide is omitted as it contains only SP images.

achieves high in-domain detection AUC (99.6%), it drops to 76% for out-of-domain. In contrast, retrained TruFor maintains consistent performance across domains in both localization (99.5%/99.6%) and detection (99.2%/98.0%). SP localization remains easier for both original and retrained models than FR, with all retrained models achieving localization AUCs above 90.0% for SP tasks. For FR images, original models perform poorly (AUCs 52.7%-74.1%) but show clear improvements after retraining, with TruFor reaching 98.6% AUC.

Tables 6 demonstrate model performance across inpainting methods. The SP/FR performance gap persists across methods - e.g., TruFor† achieves 89.9 IoU on BrushNet-SP versus 77.6 on BrushNet-FR, with similar patterns for PowerPaint (90.9 SP, 78.1 FR). HDPainter presents the most challenging case, with TruFor† achieving only 55.4 IoU compared to 76.6-78.1 for other FR methods. BrushNet and PowerPaint FR manipulations are more detectable, likely due to distinctive inpainting artifacts. This trend holds across models, with MMFusion† achieving 58.0 IoU on PowerPaint-FR but only 42.5 on HDPainter-FR. For SP cases, InpaintAnything is well-detected even by original models (34.9-63.6 IoU), likely due to its traditional copy-paste operations. HDPainter remains challenging in SP scenarios, showing consistently lower scores. HDPainter's difficulty could stem from its greater impact beyond masked regions in FR cases (Figure 3) and its blending/upscaling post-processing in SP cases.

Table 4 compares model performance between single and double inpainting cases. Original models show decreased performance on double inpainting, particularly evident in CAT-Net's IoU drop from 27.2 to 7.0. This suggests that multiple manipulations make detection more challenging for models not specifically trained for such cases. Interestingly, retrained models show more robust performance across both scenarios. TruFor† maintains similar IoU scores (81.0/79.0) while slightly improving in accuracy (95.0/99.0). CAT-Net† even shows a small improvement in IoU for double inpainting (42.4 to 49.0) while maintaining near-perfect accuracy (99.9/100.0), suggesting that retraining helps models adapt to more complex manipulation patterns.

Figure 3. Examples of inpainted images using HDPainter. Left is the original image and right is the inpainted result. HDPainter's inpainting significantly affects regions beyond the masked area, evident in the faces of the people next to the balloons.

3.1. Analysis of Model Detection Performance

The varying performance across models can be attributed to their architectural choices and training strategies. TruFor's

Data	Model	Mean IoU		Accuracy	
		Single	Double	Single	Double
Original	PSCC-Net	15.8	16.0	37.6	39.0
	CAT-Net	27.2	7.0	66.6	49.0
	MMFusion	34.5	22.0	47.9	42.0
	TruFor	31.8	21.0	29.6	32.0
SAGI-D	PSCC-Net	33.7	27.0	44.3	41.0
		+17.9	+11.0	+6.7	+2.0
	CAT-Net	42.4	49.0	99.9	100.0
		+15.2	+42.0	+33.3	+51.0
	MMFusion	64.0	59.0	84.1	88.0
		+29.5	+37.0	+36.2	+46.0
	TruFor	81.0	79.0	95.0	99.0
		+49.2	+58.0	+65.4	+67.0

Table 4. Performance comparison between original models and models retrained on SAGI-D . The table shows Mean IoU and Accuracy for both single and double inpainting manipulations. Green numbers indicate improvements compared to the original models. Retraining on SAGI-D yields significant performance improvements across all metrics and models, with TruFor showing the most substantial gains in both localization (Mean IoU) and detection (Accuracy).

Data	Model	AUC (loc)				AUC (det)	
		id	ood	SP	FR	id	ood
Original	CAT-Net	60.0	51.7	69.9	58.7	67.2	50.8
	PSCC-Net	71.6	59.2	64.8	52.7	83.4	55.8
	MMFusion	76.5	76.0	84.9	70.5	70.5	65.6
	TruFor	68.9	79.9	81.1	74.1	72.3	65.1
SAGI-D	CAT-Net	95.6	90.8	93.3	88.8	99.6	76.7
		+35.5	+39.1	+23.4	+30.1	+32.4	+26.0
	PSCC-Net	83.5	84.2	90.0	68.7	80.8	74.2
		+11.8	+25.0	+25.2	+16.1	-2.6	+18.4
	MMFusion	96.8	95.0	98.5	90.9	98.2	89.9
		+20.3	+19.0	+13.6	+20.4	+27.8	+24.3
	TruFor	99.5	99.6	99.9	98.6	99.2	98.0
		+30.5	+19.7	+18.8	+24.6	+26.9	+33.0

Table 5. Performance comparison of image forensics methods across different domains. The table shows AUC scores for both localization and detection tasks, comparing original models with those retrained on our dataset. "ID" indicates in-domain and "OOD" indicates out-of-domain performance, while SP (Splicing) and FR (Fully Regenerated) represent different forgery types. Green numbers show improvements and red numbers show decreases compared to original models. Retraining on our dataset yields significant performance improvements across most metrics and models.

superior performance likely stems from its extensively pretrained Noiseprint++ component, which was trained using self-supervised methods on images with diverse processing procedures. While MMFusion shares architectural similarities with TruFor, its use of multiple modalities may lead to overfitting, potentially explaining its lower performance compared to TruFor. On the other hand, PSCC-Net's poor localization performance can be attributed to its relatively small model size, suggesting possible underfitting. CAT-Net's performance is particularly affected by our evaluation setup for two reasons. First, its design leverages JPEG double compression artifacts for detection, but our dataset contains PNG images where quantization tables are not preserved. Second, while JPEG compression was intrinsic to CAT-Net's original training data, it lacks explicit augmentations for compression robustness unlike other models. This explains its vulnerability to JPEG compression artifacts compared to models with more robust training strategies.

3.2. Model Compression Robustness Analysis

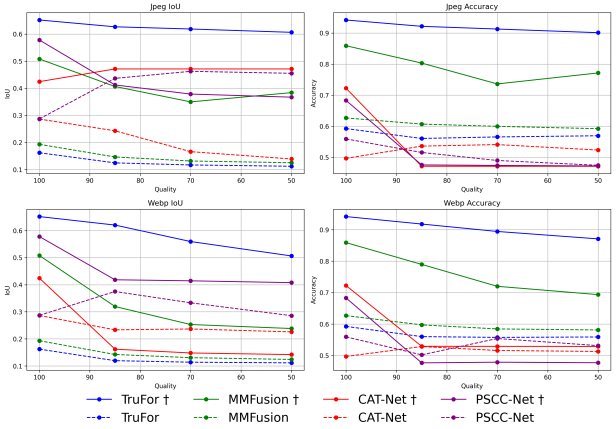


Figure 4. Robustness of model detection performance under compression. Top row shows model detection performance when subjected to JPEG compression at varying quality levels, while bottom row shows detection performance under WEBP compression.

We evaluate model robustness against JPEG and WEBP image compression at quality levels 0.85, 0.7, and 0.5. Figure 4 presents detection and localization results. Retrained TruFor shows the strongest resilience, maintaining stable performance across quality levels for both compression types. WEBP compression affects performance more than JPEG, particularly for localization tasks. All models show higher degradation in IoU scores compared to accuracy metrics, indicating that manipulation localization is more sensitive to compression artifacts compared with detection.

3.3. Comparison with human performance

In this section, we present extended results of our human evaluation study, analyzing both the participants' detection performance in more cases and the relationship between demographic factors and classification accuracy.

Data	Model	FR								SP									
		BN		CN		HDP		PPt		BN		HDP		IA		PPt		RA	
		IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc
Original	CN	3.0	36.8	9.0	40.4	5.0	34.4	4.1	48.0	31.6	72.2	1.5	58.3	62.6	98.7	22.4	67.0	52.4	96.2
	PS	12.2	36.6	7.9	35.9	5.7	25.9	12.7	33.8	16.5	40.3	17.3	53.8	34.9	48.4	10.6	23.2	14.6	35.5
	MM	20.2	24.1	18.5	29.2	10.9	24.8	17.2	20.9	67.7	75.3	26.7	50.2	63.6	88.7	46.4	57.2	30.5	43.6
	TF	22.5	14.0	21.7	14.3	12.1	9.9	21.3	10.6	56.7	49.8	23.7	26.6	60.7	70.1	42.5	35.7	20.9	21.0
SAGI-D	CN	38.3	99.9	38.9	100.0	28.3	99.9	37.9	99.7	41.5	100.0	34.3	100.0	48.5	99.8	42.0	100.0	54.0	99.9
		+35.3	+63.1	+29.9	+59.6	+23.3	+65.5	+33.8	+51.7	+9.9	+27.8	+32.8	+41.7	-14.1	+1.1	+19.6	+33.0	+1.6	+3.7
	PS	23.8	67.9	9.2	15.2	16.5	15.5	38.1	64.1	32.3	43.2	39.7	68.0	53.0	51.7	46.2	51.4	39.8	43.9
		+11.6	+31.3	+1.3	-20.7	+10.8	-10.4	+25.4	+30.3	+15.8	+2.9	+22.4	+14.2	+18.1	+3.3	+35.6	+28.2	+25.2	+8.4
	MM	45.1	63.8	53.4	89.4	42.5	82.1	58.0	83.5	80.5	91.7	58.5	73.0	73.6	85.1	80.7	92.7	72.2	86.3
		+24.9	+39.7	+34.9	+60.2	+31.6	+57.3	+40.8	+62.6	+12.8	+16.4	+31.8	+22.8	+10.0	-3.6	+34.3	+35.5	+41.7	+42.7
	TF	77.6	92.1	76.6	94.6	55.4	92.5	78.1	98.3	89.9	96.0	80.6	95.0	84.7	92.5	90.9	98.7	88.0	96.6
		+55.1	+78.1	+54.9	+80.3	+43.3	+82.6	+56.8	+87.7	+33.2	+46.2	+56.9	+68.4	+24.0	+22.4	+48.4	+63.0	+67.1	+75.6

Table 6. Performance comparison of image forensics methods CAT-Net (CN), MMFusion (MM), PSCC-Net (PS), and TruFor (TF) for both FR (Fully Regenerated) and SP (Splicing) scenarios. For each method, Mean IoU and Accuracy (Acc) scores are shown. Inpainting Models: BN (BrushNet), CN (ControlNet), HDP (HDPainter), PPt (PowerPaint), IA (InpaintAnything), and RA (RemoveAnything). The green values in the second row for each retrained model indicate the improvement over the corresponding original model.

Participants included 26 males, 6 females, and 10 undisclosed. Ages ranged from 18 to 65+, with the largest group being 18-24 (19), followed by 25-34 (8) and 35-44 (6). Users were asked to detect inpainting and draw bounding boxes around suspected manipulated regions. For IoU computation, ground truth masks were converted to bounding boxes.

Table 7 shows the results of chi-square tests [14] for independence between demographic variables and classification accuracy (i.e., the proportion of correctly identified images), after filtering out participants with fewer than 20 votes (reducing the sample from 42 to 34 participants). The chi-square tests indicated that the independence hypothesis could not be rejected in any case, and the effect sizes (measured by Cramer's V [4]) showed negligible to weak associations, suggesting that these demographic factors have limited practical significance.

Demographic Factor	Cramer's V	p-value
Gender	0.0661	0.0014
Age	0.0988	0.0003
Education Completed	0.0929	0.0004
Current Education	0.0986	0.0001
AI Familiarity	0.0811	0.0013
DIP Familiarity	0.0978	0.0002

Table 7. Chi-square test results for demographic factors vs verification accuracy

Table 8 presents the comparison between human evaluators and automated models. Human performance reached 67.4% accuracy and 15.2 IoU, significantly lower than retrained models like TruFor (95.3% accuracy, 68.3%) and MMFusion (87.9% accuracy, 50.9% IoU). Results are broken down into four categories: *All* represents performance on the complete test set, while *Deceiving*, *Non-Deceiving* and *Intermediate* correspond to UGDA's classification of images based on their potential to fool human perception. The *Intermediate* category includes images that passed the initial realism check but not the second. Users particularly struggled with deceiving images (35.2% accuracy, 12.9% IoU) compared to non-deceiving ones (73.7% accuracy, 40.4% IoU), validating UGDA's effectiveness in identifying manipulations that are challenging for human perception. Also, the performance of humans on the intermediate category (59.7% accuracy, 28.3% IoU) confirms that the second stage is indeed effective in discarding images that are not truly deceiving. In contrast, retrained models maintain high performance even on these challenging cases, with TruFor achieving 98.9% accuracy and 87.4% IoU on deceiving images. The performance gap between humans and models emphasizes the importance of automated detection methods, particularly for high-quality inpainting that can bypass human perception.

In Table 9 we see the performance comparison between spliced (SP) and fully regenerated (FR) images, across human evaluators and the forensic models. The results reveal that human performance remains consistent across both manipulation types, showing no significant advantage in detecting either SP (0.34 for Deceiving, 0.76 for Non-Deceiving) or FR manipulations (0.37 for Deceiving, 0.74 for Non-Deceiving) in contrast to the forensic models.

Model	Accuracy				Mean IoU			
	All	Dec.	Int.	ND.	All	Dec.	Int.	ND.
Human	67.4	35.2	59.7	73.7	15.2	12.9	28.3	40.4
PSCC	32.1	29.6	37.5	29.2	14.4	15.2	14.1	13.8
CAT-Net	62.5	70.4	59.4	57.6	19.5	29.4	14.2	14.9
PSCC†	51.3	49.6	50.0	54.4	36.3	35.2	30.4	43.3
TruFor	27.3	35.2	25.0	21.6	29.0	35.4	24.7	27.0
MMFus	39.9	48.8	34.4	36.4	32.1	38.1	30.9	27.3
CAT-Net†	**100**	**100**	**100**	**100**	47.6	45.8	44.5	52.6
MMFus†	90.5	89.2	90.6	91.6	69.6	66.5	70.3	72.1
TruFor†	99.7	99.2	**100**	**100**	**87.6**	**86.9**	**86.6**	**89.4**

Table 8. Human vs. model performance comparison on inpainting detection. Results show accuracy and IoU for full test set (All) and images classified by UGDA as Deceiving (Dec.) or Non-Deceiving (ND.). † indicates models retrained on our dataset. Bold values indicate the best performance per column.

Model	Accuracy				IoU			
	Dec.		Non-Dec.		Dec.		Non-Dec.	
	SP	FR	SP	FR	SP	FR	SP	FR
Human	34.1	37.2	75.8	74.4	12.0	14.0	41.3	41.9
TruFor	47.4	11.8	33.7	10.7	43.5	20.2	38.0	20.0
MMFus	61.5	25.0	50.0	23.2	49.3	18.7	42.1	17.5
PSCC-Net	31.4	25.0	31.6	28.6	17.3	11.5	17.2	11.5
PSCC-Net†	51.9	39.5	72.4	41.1	44.8	16.5	61.1	31.7
CAT-Net	84.6	44.7	77.6	41.1	44.2	4.5	30.4	4.5
MMFus†	92.9	84.2	91.8	92.9	78.0	49.3	82.0	67.8
TruFor†	99.4	98.7	**100**	**100**	91.7	78.9	95.3	88.0
CAT-Net†	**100**	**100**	**100**	**100**	48.8	41.0	53.4	53.5

Table 9. Human vs. model performance comparison on inpainting detection. Results show accuracy and IoU for full test set images classified by UGDA as Deceiving (Dec.) or Non-Deceiving (Non-Dec.), SP and FR. † indicates models retrained on our dataset. Bold values indicate the best performance per column.

3.4. Qualitative Analysis

In Figure 5, we present a comparison of the localization maps before and after fine-tuning. The results demonstrate that fine-tuning can significantly improve localization performance. For PSCC-Net, while improvements are observed in the second and fifth rows, poor localization results persist in other cases. Regarding the remaining models, localization improvements are evident in all cases, with TruFor consistently demonstrating the most accurate localization maps. The second row showcases an example where the original CAT-Net, MMFusion, and TruFor successfully identified the inpainted area, while the fourth row presents a case where the original model could only partially detect the inpainted region. The fifth row presents an interesting case involving the original MMFusion model. If the predicted mask were inverted, it would have successfully identified the inpainted area. This can be attributed to the fact that in splicing it can be ambiguous which area is spliced and which is original. In AI inpainting cases, however, there is no ambiguity about which region has been modified.

4. Example Outputs

Figures 6 and 7 present a qualitative analysis of some cases from our dataset. In Figure 6, we show successful inpainting examples across different models and datasets (COCO, RAISE, and OpenImages), where the models correctly follow the prompts while producing realistic results.

Figure 7 presents different failure modes. The top two rows reveal problems with LLM-generated prompts, showing cases where prompts either fail to match the scene context or lead to technically sound but unrealistic results. Row 3 demonstrates technical limitations with visible artifacts and blurs. Row 4 presents cases where the inpainting appears realistic but deviates from the given prompt. Row 5 shows examples of poor inpainting quality where the models fail to generate coherent content. Finally, row 6 illustrates a subtle failure mode where the inpainting is technically well-executed but produces results that appear unnatural to human observers upon closer inspection. While these cases might be easily identifiable as manipulated by careful observers, they could potentially deceive viewers who are not actively looking for signs of manipulation, highlighting the importance of including such examples in inpainting datasets for developing robust detection methods.

References

[1] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. Gradio: Hassle-free sharing and testing of ml models in the wild, 2019. 3

[2] Anthropic. Claude 3.5 sonnet, 2024. Large Language Model. 1

[3] Rodrigo Benenson and Vittorio Ferrari. From couloring-in to pointillism: revisiting semantic segmentation supervision. In *ArXiv*, 2022. 1

[4] Harald Cramér. *Mathematical Methods of Statistics*. Princeton University Press, Princeton, 1946. Chapter 21, The two-dimensional case, page 282. Table of contents archived at Wayback Machine, 2016-08-16. 6

[5] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: a raw images dataset for digital image forensics. In *Proceedings of the 6th ACM Multimedia Systems Conference*, page 219–224, New York, NY, USA, 2015. Association for Computing Machinery. 1

[6] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization, 2023. 3

[7] Xiaodan Ju et al. Brushnet: Plug-and-play image inpainting with user guidance. In *Proceedings of the IEEE/CVF Con-*

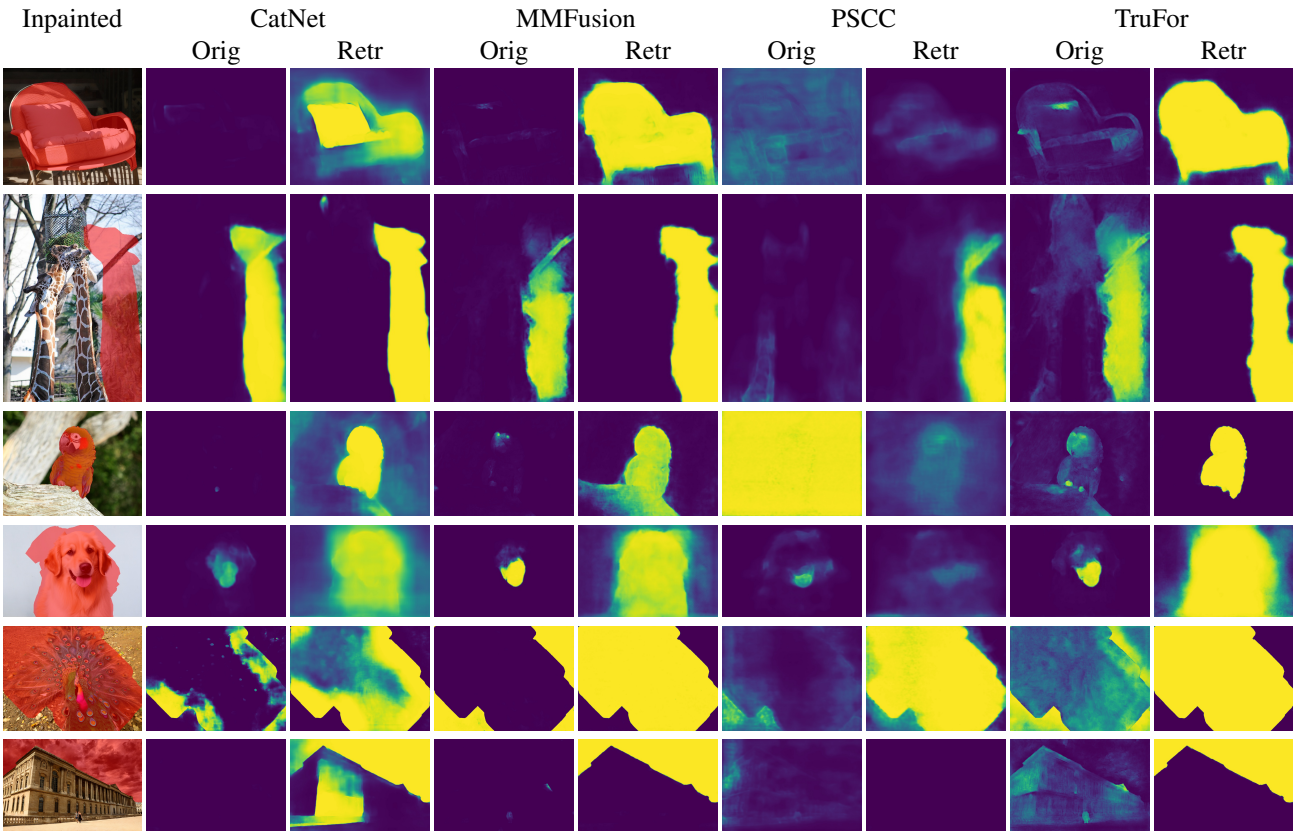


Figure 5. Comparison of forgery localization results. For each row, from left to right: inpainted image, followed by localization maps from CatNet, MMFusion, PSCC, and TruFor models, showing both original (Orig) and retrained (Retr) versions.

ference on Computer Vision and Pattern Recognition, pages 5678–5687, 2024. 1

[8] Myung-Joon Kwon, Seung-Hun Nam, In-Jae Yu, Heung-Kyu Lee, and Changick Kim. Learning jpeg compression artifacts for image manipulation detection and localization. *International Journal of Computer Vision*, 130(8):1875–1895, 2022. 1, 3

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 1

[10] Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu. Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32 (11):7505–7517, 2022. 3

[11] Ara Manukyan. Hd-painter: High-resolution prompt-faithful text-guided image inpainting, 2024. 1

[12] OpenAI. Chatgpt-3.5, 2023. 1

[13] OpenAI. Chatgpt-4, 2023. 2

[14] Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50 (302):157–175, 1900. 6

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1

[16] Konstantinos Triaridis and Vasileios Mezaris. Exploring multi-modal fusion for image manipulation detection and localization, 2023. 3

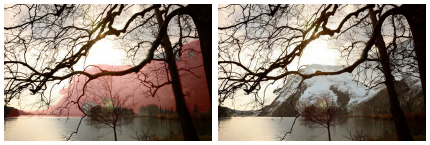
[17] Ning Yu, Xiang Zhao, and Bo Chen. Inpaint-anything: Segment meets inpaint. *arXiv preprint arXiv:2304.06790*, 2023. 1

[18] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 1

[19] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 3

[20] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting, 2024. 1

(a) “a juicy orange to add a vibrant pop of color to the composition”



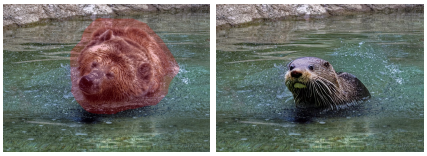
(b) “a majestic snow-capped mountain to create a scenic landscape”

(c) “a vibrant blue poison dart frog”

(d) “a cozy blanket and fluffy pillows to complete the bedroom scene”

(e) “a grand marble fountain surrounded by lush greenery”

(f) “a decorative ceramic vase”

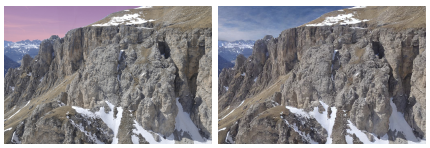


(g) “a playful otter swimming in the river stream”

(h) “a cluster of small red berries growing in the grass”

(i) “a rustic wooden barrel planter”

(j) “a fresh, delicious sandwich to complete the meal”



(k) “a clear blue sky to enhance the mountain landscape”

(l) “a ripe golden delicious apple”

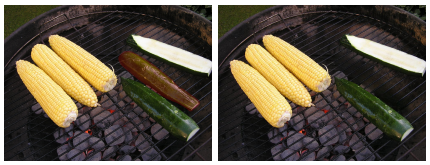
(m) “a delicious cheeseburger to make the meal even more tempting”

(n) “a lush green meadow, adding a touch of nature to the serene landscape”

(o) “an intricately carved wooden eagle head”

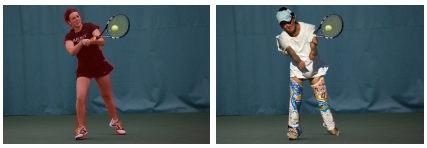
(p) No prompt

(q) No prompt



(r) No prompt

Figure 6. Example pairs of original images (with inpainting mask overlaid in semi-transparent red) and their corresponding inpainted results across three datasets: COCO (first column), RAISE (second column), and OpenImages (third column). Each row showcases results from a different inpainting model: BrushNet, PowerPaint, HD-Painter, ControlNet, Inpaint-Anything, and Remove-Anything. The text below each pair shows the prompt used for text-guided models.



(a) “a person playing volleyball on the beach”

(b) “a colorful hot air balloon floating in the sky”

(c) “a red London phone booth”

(d) “a blue bus traveling down the tracks”

(e) “a playful panda bear imitating a martial arts move”

(f) “a fluffy orange tabby cat with bright blue”

(g) “vibrant red cherries to create a fruity collage effect”

(h) “a majestic deer with large ant”

(i) No prompt

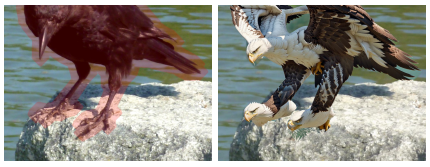
(j) “a group of young adults playing frisbee”

(k) “a bright red mailbox to blend seamlessly into the park scene”

(l) “a colorful bowl of fruit salad”

(m) “a majestic lion standing proudly in the savanna”

(n) “a large, majestic white husky standing”



(o) “a majestic eagle perched on”

(p) “a woman wearing a scarf and holding a bouquet of flowers”

(q) “a bowl of fresh fruits”

(r) “a friendly raccoon walking across a stone wall near trees”

Figure 7. Examples of failure cases in inpainting. Row 1: LLM-generated prompts that fail to match the image context. Row 2: Technically sound inpainting results that generate improbable real-world scenarios. Row 3: Results with visible artifacts and blurs. Row 4: Realistic inpaintings that don’t follow the given prompts. Row 5: Cases where the inpainting fails to produce coherent results. Row 6: Realistic but uncanny results that human observers can potentially identify as artificial.