Any-Resolution AI-Generated Image Detection by Spectral Learning

Supplementary Material

1. Evaluation Setup in Details

To better facilitate the reproducibility of our results, in this paragraph, we provide a more extensive description of the implementation of our evaluation setup.

1.1. Projection Operators

$\mathbf{P}_n$: In our architecture we employed N projection operators $\mathbf{P}_n : R^d \rightarrow R^D$ for projecting the representations computed by each block of the vision transformer [6] $\mathcal{G}$ into a space that facilitates the operations of spectral reconstruction similarity. These operators apply a LayerNorm [16] operation to their input and then process the normalized results using a sequence of a linear, a GELU [10] and another linear operation. Finally, a LayerNorm operation is applied to their output.

$\mathcal{P}_1$ **and** $\mathcal{P}_2$: To build the spectral context vector based on the learnable spectral map C we used the projection operators $\mathcal{P}_1(\cdot) : R^{2D} \rightarrow R^D$ and $\mathcal{P}_1(\cdot) : R^D \rightarrow R^D$ that share the same architecture. In particular, they first process input features using two linear layers with GELU [10] activations and then normalize their output using LayerNorm [16].

1.2. Implementation and Training Details

We implement our approach using PyTorch and train it for 35 epochs using the AdamW [19] optimizer. In the first five epochs we perform a linear warmup of the learning rate (lr) and increase it from $2.5e-7$ to $5e-4$. Then, from the 5th epoch we apply a per-step cosine decay to ultimately decrease it to $2.5e-7$ at the last step of the 35th epoch. We set the latent dimensionality $D = 1024$ and the masking radius $r = 16$ according to the hyperparameter tuning procedure that we presented in Sec. 4.3 of the main paper. The vision transformer (ViT) [6] that we employ in our model of real images $\mathcal{G}$ uses a patch size $p = 16$ and includes $N = 12$ transformer [26] blocks with a latent dimensionality $d = 768$. To compute the 2D Discrete Fourier Transform we use the Fast Fourier Transform algorithm.

We split each image into K patches of size $h = w = 224$, while we empirically set the latent dimensionality of the spectral context attention to $D_h = 1536$. During training we employ $K_{training} = 4$ patches. These patches are generated as augmented views of the input image, using random resizing, cropping, rotation, Gaussian blur, Gaussian noise and JPEG compression augmentations. The training of spectral reconstruction similarity, spectral context vector and spectral context attention is performed on a single Nvidia L40S 48GB GPU, using mixed-precision arithmetic, and takes about 50 hours.

Approach	Split	Gen. #	Real #	Size
Latent Diff. [2]	train	180k	180k	0.1 MP
Latent Diff. [2]	val.	20k	20k	0.1 MP

Table 1. Training and validation data. The average image size in megapixels is presented for each split.

	Origin	Images #	Size
AI-Generated	Glide [1]	1k	0.1 MP
	SD1.3 [1]	1k	0.3 MP
	SD1.4 [1]	1k	0.3 MP
	Flux [15]	1k	0.8 MP
	DALLE2 [1]	1k	1.0 MP
	SD2 [1]	1k	1.0 MP
	SDXL [1]	1k	1.0 MP
	SD3 [8]	1k	1.0 MP
	GigaGAN [12]	1k	1.0 MP
	MJv5 [1]	1k	1.2 MP
	MJv6.1 [21]	631	1.2 MP
	DALLE3 [1]	1k	1.2 MP
	Firefly [1]	1k	4.1 MP
Real	ImageNet [5]	1k	0.2 MP
	COCO [17]	1k	0.3 MP
	OpenImages [14]	1k	0.8 MP
	FODB [9]	1k	2.8 MP
	RAISE-1k [3]	1k	15 MP

Table 2. Test data. The average image size in megapixels is presented for each generative model and source of real images.

Ultimately, we select the best epoch as the one that minimizes loss on the validation split of our training dataset [2]. However, we noticed that the performance on these validation samples would saturate very quickly, reaching a near-optimal level during the first few epochs, without further indicating whether the model would learn useful patterns. To make validation setup more challenging, without using any external data, we applied once our augmentation policy to the validation data. Then, we used this augmented validation split for selecting the best epoch. We provide an overview of our training and validation data in Tab. 1 as well as our test data in Tab. 2.

2. Runtime Analysis

To evaluate the computational performance of our approach we analyze its runtime using the proposed spectral context attention as well as by solely relying on the scaled dot prod-

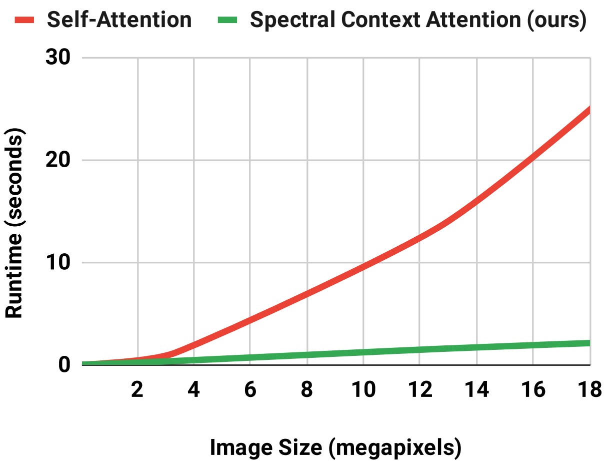


Figure 1. Runtime comparison between vision transformer's self-attention and spectral context attention for different image resolutions. Lower is better. 18 megapixels was the maximum possible size to scale self-attention due to memory constraints, while spectral context attention did not face similar issues.

Attention	0.1 MP	1 MP	10 MP	1000 MP
Self-Att.	0.68 GB	2.05 GB	17.7 GB	N/A
SCA (ours)	0.62 GB	0.68 GB	0.97 GB	38.6 GB

Table 3. Comparison of the required GPU memory to process images of different size between vision transformer's self-attention and spectral context attention. Using spectral context attention we managed to analyze gigapixel sized images using a single GPU. MP stands for megapixel and GB for gigabyte. Lower is better.

uct self-attention of the vision transformer [6]. We report the runtime across images of different resolution in Fig. 1. As we see, the runtime increases rapidly when relying on the self-attention of quadratic computational complexity, while, due to memory constraints, the maximum image resolution we could process using an Nvidia L40S 48GB GPU was limited to 18 megapixels. Instead, employing spectral context attention enables our approach to scale linearly w.r.t the size of the image, without being limited by the available memory. To better highlight the difference in memory requirements, we present in Tab. 3 the required GPU memory for different image sizes. We see that for a 10 megapixels image self-attention requires almost 18 gigabytes of memory. Instead, when using spectral context attention less than 1 gigabyte of memory is required. Finally, using spectral context attention we managed to analyze with our architecture images of 1 gigapixel, under a single-GPU setup, exceeding by a large margin the size of the images produced by commercial cameras at the time of writing this manuscript. To eliminate any possible inconsistencies due to inefficient implementations, in our runtime analysis we employed the cuda-optimized FlashAttention [4].

3. Feature Space Analysis

To study the effect of our key architectural components in the feature space we embed the spectral context vector (SCV) z^C, the spectral reconstruction similarity (SRS) values z^λ and the image-level spectral vector $\mathbf{z}^S$ generated by spectral context attention (SCA) using t-SNE [25] and present the results in Fig. 2. We embed these three latent representations for AI-generated images of different resolution, originating from Stable Diffusion 1.4, Stable Diffusion XL and MidJourney-v5, while, for illustration purposes, limiting our source of real images to COCO. As we see, the spectral context vector itself provides minimal discriminative capability, highlighting that our approach does not rely upon the context of the images. On the other hand, while SRS values provide significant discriminative capability, there is lots of noise involved, as different SRS values are useful for image patches with different spectral context. Finally, using the SCA to combine the most discriminative of the SRS values according to the spectral context of each patch produces highly separable image-level embeddings, verifying our original intuition for building this mechanism.

4. Additional Metrics

To study the calibration of our approach with respect to the state-of-the-art detectors as well as to facilitate comparison across popular metrics in the field, we expand the analysis of the main paper by computing the balanced accuracy on the 0.5 threshold and the average precision metrics. We report the results across our test set of 5 sources of real images and 13 generative models in Tab. 4 and Tab. 5 respectively. As we see in the former, our approach, based on spectral learning, not only provides significant discriminative ability, but is also better calibrated around the common 0.5 threshold, providing an absolute increase of 2.5% in balanced accuracy. Moreover, SPAI achieves an absolute increase of 3.5% in terms of average precision, reinstating its superior discriminative ability.

5. Ethical Considerations

Introducing an approach for distinguishing AI-generated content from real one intends to prevent the malicious exploitation of generative AI. However, any detection method, will inevitably fail to correctly predict some cases, allowing malicious actors to exploit such results to either promote generated content or discredit real one. Yet, we believe that the improved generalization performance of our approach across several generative models as well as its superior robustness against several common attacks, ultimately decrease the potential of exploitation.

Image Size	< 0.5 MPixels			0.5 - 1.0 MPixels						> 1.0 MPixels				**Acc.**
Approach	**Glide**	**SD1.3**	**SD1.4**	**Flux**	**DALLE2**	**SD2**	**SDXL**	**SD3**	**GigaGAN**	**MJv5**	**MJv6.1**	**DALLE3**	**Firefly**	**AVG**
Dire [28]	38.2	57.7	58.5	46.0	51.5	61.7	47.3	49.6	40.9	43.8	50.4	62.6	51.1	50.7
CNNDet. [27]	52.0	52.2	52.3	49.1	58.0	52.2	56.6	47.5	63.0	50.8	52.6	47.6	55.5	53.0
NPR [24]	79.9	79.9	79.4	29.9	29.9	30.1	29.9	79.9	79.9	31.4	29.9	79.9	29.9	53.1
Fusing [11]	54.1	52.9	53.0	50.5	59.4	52.6	51.4	48.2	62.6	53.2	58.6	48.2	55.9	53.9
FreqDet. [7]	46.9	80.4	80.6	42.6	42.8	46.3	61.8	65.1	60.9	44.0	41.6	46.9	77.2	56.7
UnivFD [20]	50.8	67.9	67.4	45.9	81.0	73.2	65.5	45.0	73.6	50.0	53.8	44.9	88.2	62.1
LGrad [23]	69.3	78.6	78.8	68.9	79.2	54.6	64.3	34.0	80.4	63.3	73.6	37.2	42.1	63.4
GramNet [18]	72.7	78.7	79.1	73.9	80.4	58.5	72.3	32.4	80.4	58.2	80.3	37.0	34.8	64.5
DeFake [22]	76.3	59.5	59.0	79.6	45.2	61.7	51.7	78.3	67.0	62.2	77.1	81.5	44.0	64.8
DMID [2]	52.4	99.3	99.3	82.1	49.5	97.9	96.5	57.9	54.3	98.4	78.1	49.3	57.5	74.8
PatchCr. [29]	69.6	86.6	86.7	79.7	71.7	86.8	89.4	40.2	89.0	72.6	87.9	40.6	72.5	74.9
RINE [13]	88.5	96.6	96.4	84.1	82.0	89.6	95.5	47.0	83.3	90.1	69.2	47.6	67.0	79.8
SPAI (Ours)	81.3	92.2	92.3	71.9	83.4	88.8	90.5	61.3	76.2	87.2	74.3	80.8	90.3	**82.3**

Table 4. Comparison against state-of-the-art. Average accuracy over 5 sources of real images is reported. Lower values are highlighted in red, while higher values are highlighted in green. Best overall average value is highlighted in bold, while second best is underlined.

Image Size	< 0.5 MPixels			0.5 - 1.0 MPixels						> 1.0 MPixels				**AP**
Approach	**Glide**	**SD1.3**	**SD1.4**	**Flux**	**DALLE2**	**SD2**	**SDXL**	**SD3**	**GigaGAN**	**MJv5**	**MJv6.1**	**DALLE3**	**Firefly**	**AVG**
Dire [28]	39.5	56.6	57.4	45.2	52.2	70.2	47.8	48.0	41.7	44.1	40.1	59.7	48.6	50.1
CNNDet. [27]	58.0	57.6	59.4	44.1	70.8	57.2	67.0	37.8	75.6	50.6	47.2	35.5	69.2	56.1
FreqDet. [7]	46.3	90.6	90.8	42.6	45.2	46.7	60.5	64.7	63.7	43.6	30.5	47.5	71.7	57.3
NPR [24]	75.4	87.6	72.3	39.3	31.3	35.2	38.3	72.4	84.0	38.2	31.0	97.4	50.4	57.9
Fusing [11]	64.7	63.6	63.0	57.7	77.1	65.6	61.9	42.1	80.7	64.6	67.6	36.6	74.0	63.0
LGrad [23]	80.1	76.8	78.4	71.7	83.1	59.3	69.2	32.6	89.7	67.2	69.3	38.3	44.6	66.2
GramNet [18]	74.9	79.4	79.8	74.6	80.3	65.2	74.0	36.2	80.1	63.5	73.3	48.0	45.5	67.3
UnivFD [20]	62.6	82.1	81.9	43.3	91.4	86.1	79.9	38.6	87.0	57.9	52.9	39.6	95.9	69.2
DeFake [22]	87.0	63.0	62.5	90.8	44.9	66.7	54.7	87.4	73.9	66.6	83.8	93.2	42.9	70.6
PatchCr. [29]	79.7	96.3	96.7	86.8	80.0	96.1	95.3	41.5	97.7	81.1	94.1	39.0	77.5	81.7
DMID [2]	71.5	100.0	100.0	97.2	54.9	99.7	99.7	71.8	70.2	99.9	93.2	45.3	87.7	83.9
RINE [13]	95.5	99.9	99.9	94.0	93.2	96.8	99.4	46.9	93.4	97.0	78.0	48.6	83.1	86.6
SPAI (Ours)	90.9	99.3	99.4	82.0	90.7	96.8	97.2	72.4	84.7	95.1	79.1	88.8	94.9	**90.1**

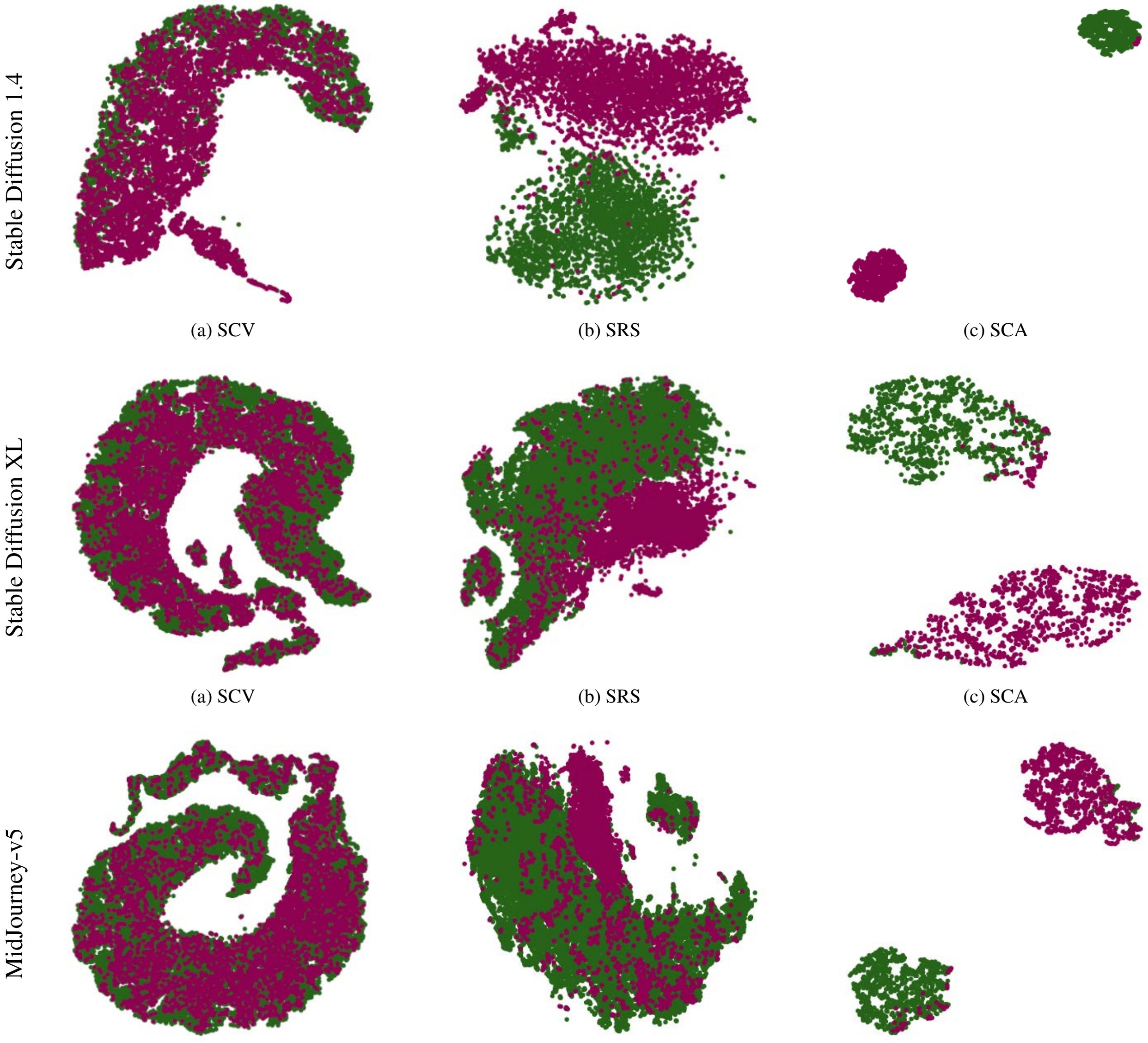
Table 5. Comparison against state-of-the-art. Average precision over 5 sources of real images is reported. Lower values are highlighted in red, while higher values are highlighted in green. Best overall average value is highlighted in bold, while second best is underlined.

6. Qualitative Evaluation

We perform a qualitative evaluation of our approach across all the considered datasets and present samples for the 13 generative models in Figs. 3 to 15 as well as for the 5 sources of real images in Figs. 16 to 20. As we see, our approach accurately detects images generated by all the considered generative approaches, depicting a diverse set of topics and incorporating different levels of visual fidelity and aesthetics. At the same time, SPAI accurately classifies real images originating from all the employed sources. Therefore, employing spectral learning enables our architecture to not rely on some high-level semantics, but to effectively detect the subtle inconsistencies introduced by the generative models, to distinguish between AI-generated and real imagery.

7. Source Code

To facilitate the reproduction of our results as well as further research in the field we make publicly available our source code, data and trained models on https://mever-team.github.io/spai.



(a) SCV (b) SRS (c) SCA

(a) SCV (b) SRS (c) SCA

(a) SCV (b) SRS (c) SCA

Figure 2. t-SNE embeddings for the spectral context vector (SCV) z^C, the spectral reconstruction similarity (SRS) values z^λ and the image-level spectral vector $\mathbf{z}^S$ generated by the spectral context attention (SCA) for AI-generated images from three generative models and a common source of real images. Each dot corresponds to the embeddings of different image patches in the case of (a) SCV and (b) SRS and to different images in the case of (c) SCA. Embeddings for AI-generated samples are denoted in green, while for real ones in purple. The spectral context itself cannot discriminate between real and AI-generated samples (a). While spectral reconstruction similarity values provide significant discriminative ability, lots of noise is involved (b). Instead, using the spectral context attention to combine the most discriminative of the SRS values, according to the spectral context of each patch, produces highly separable image-level embeddings (c).

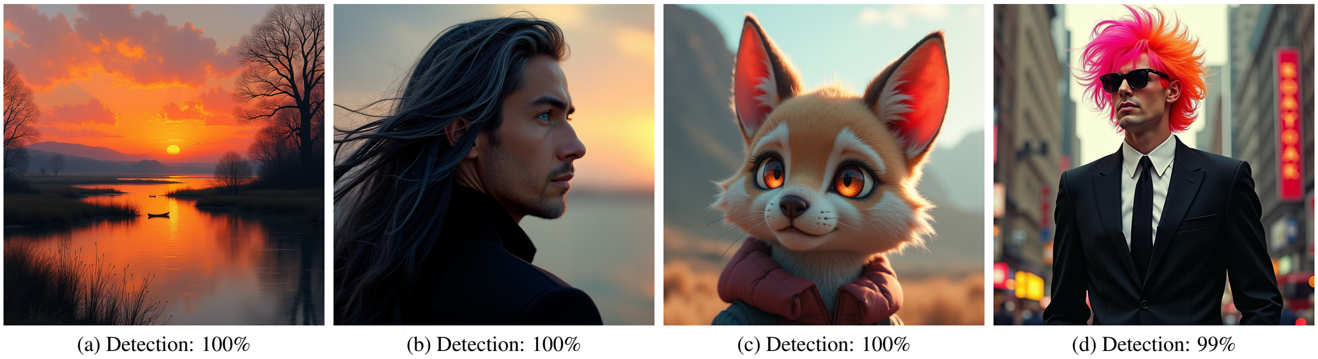
References

[1] Quentin Bammey. Synthbuster: Towards detection of diffusion model generated images. *IEEE Open Journal of Signal Processing*, 2023. 1

[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. IEEE, 2023. 1, 3

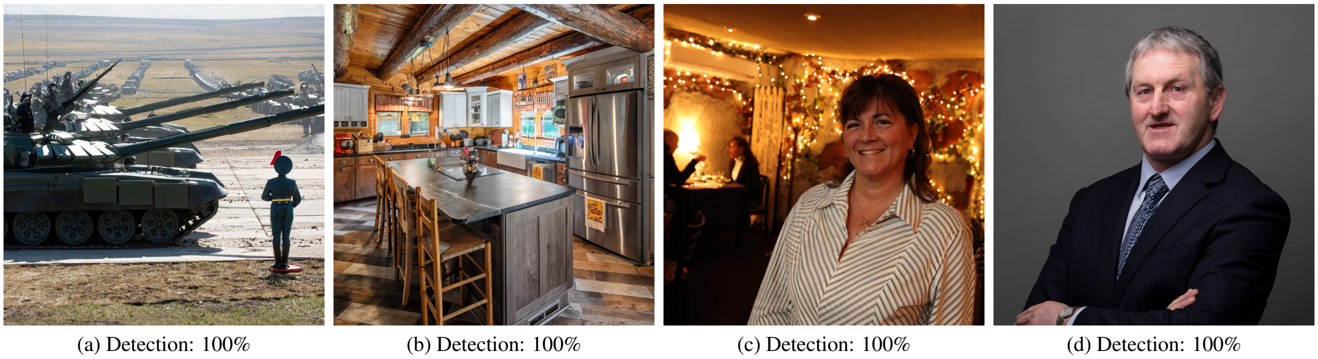
[3] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM multimedia systems conference*, pages 219–224, 2015. 1

[4] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information*



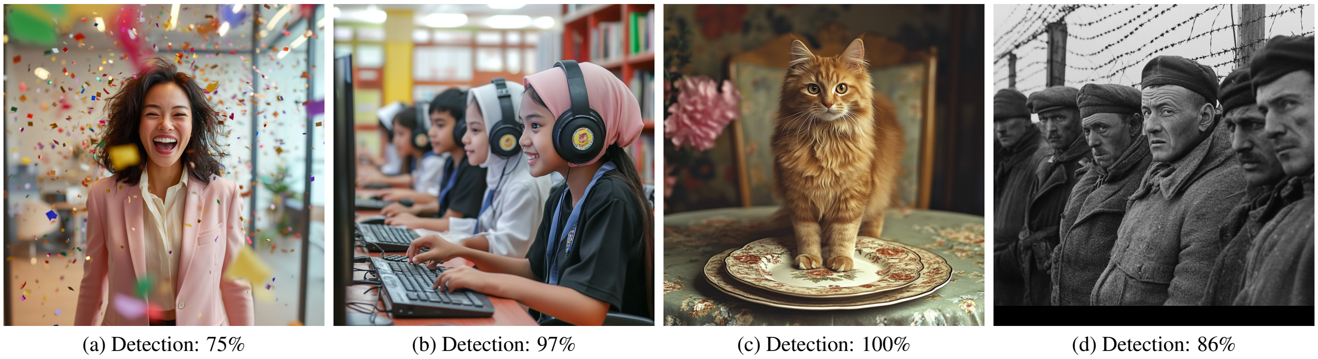
(a) Detection: 100% (b) Detection: 100% (c) Detection: 100% (d) Detection: 99%

Figure 3. Accurately detected Flux images. For illustration purposes cropped to a square aspect ratio.



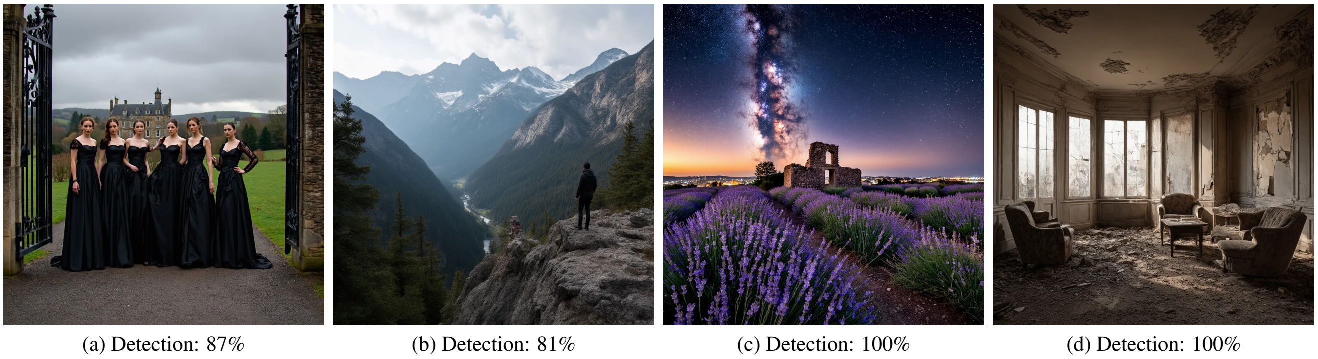
(a) Detection: 100% (b) Detection: 100% (c) Detection: 100% (d) Detection: 100%

Figure 4. Accurately detected GigaGAN images. For illustration purposes cropped to a square aspect ratio.



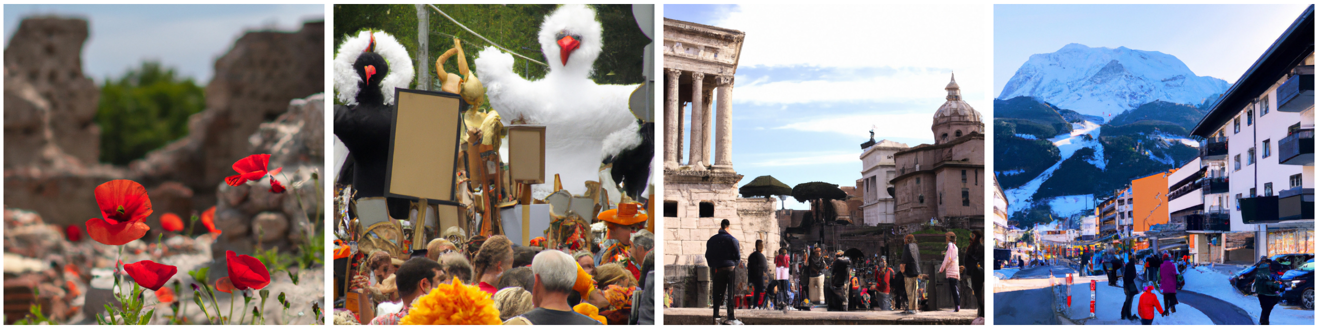
(a) Detection: 75% (b) Detection: 97% (c) Detection: 100% (d) Detection: 86%

Figure 5. Accurately detected MidJourney-v6.1 images. For illustration purposes cropped to a square aspect ratio.



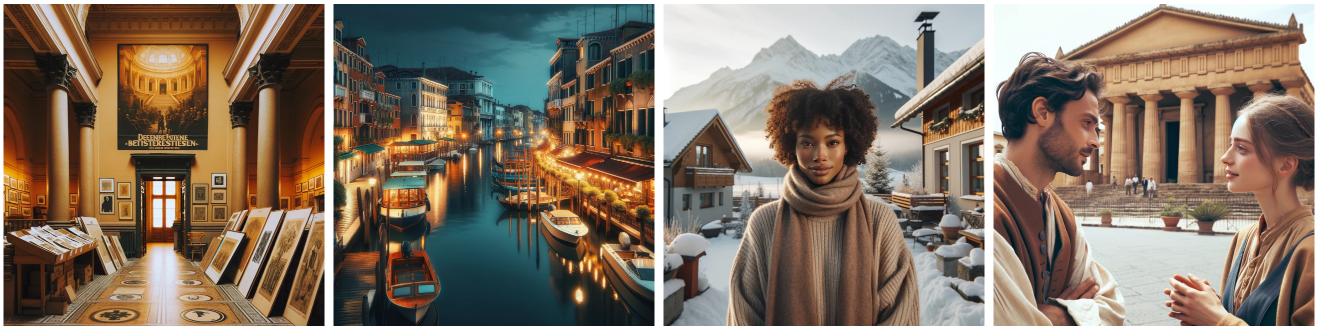
(a) Detection: 87% (b) Detection: 81% (c) Detection: 100% (d) Detection: 100%

Figure 6. Accurately detected Stable Diffusion 3 images. For illustration purposes cropped to a square aspect ratio.



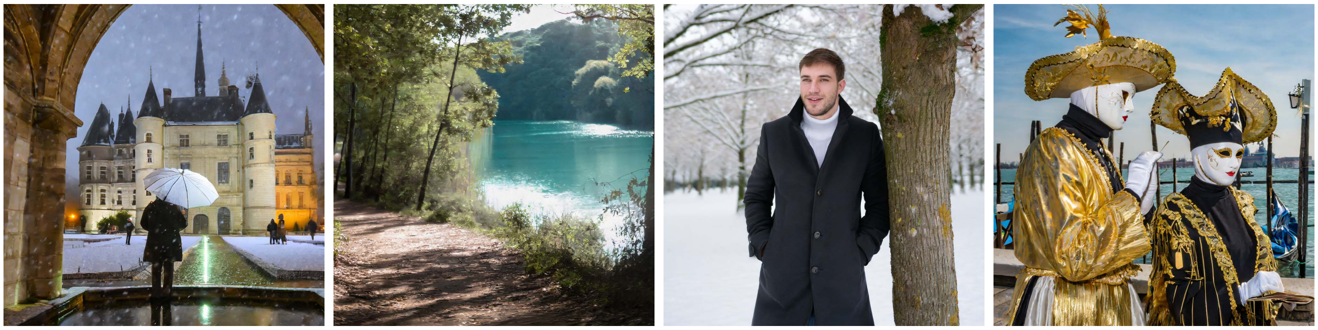
(a) Detection: 99% (b) Detection: 100% (c) Detection: 100% (d) Detection: 99%

Figure 7. Accurately detected DALLE2 images. For illustration purposes cropped to a square aspect ratio.



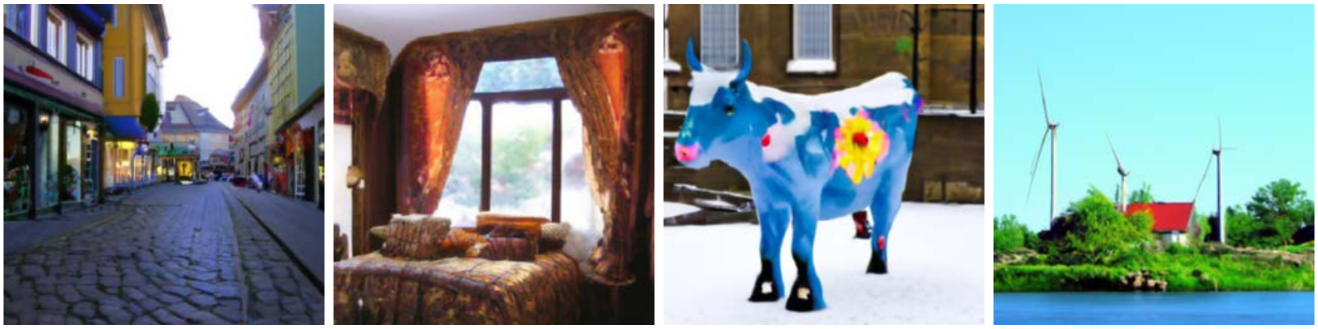
(a) Detection: 90% (b) Detection: 100% (c) Detection: 99% (d) Detection: 87%

Figure 8. Accurately detected DALLE3 images. For illustration purposes cropped to a square aspect ratio.



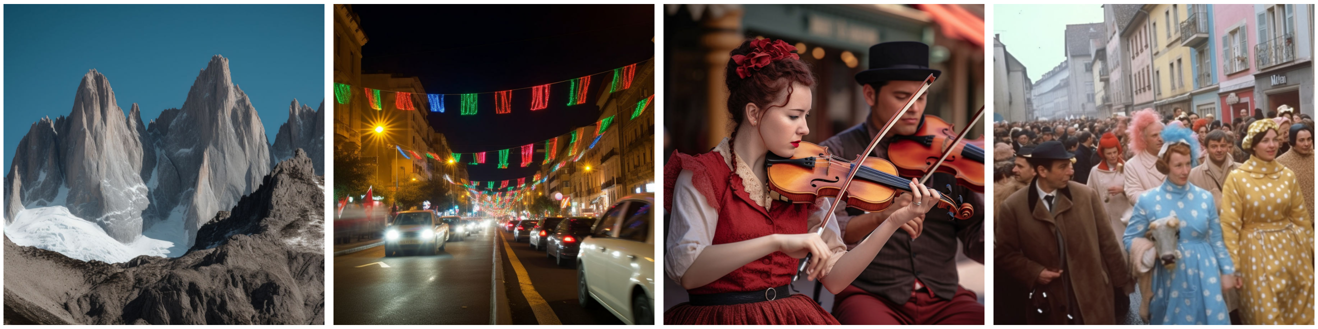
(a) Detection: 100% (b) Detection: 86% (c) Detection: 100% (d) Detection: 100%

Figure 9. Accurately detected Firefly images. For illustration purposes cropped to a square aspect ratio.



(a) Detection: 100% (b) Detection: 100% (c) Detection: 100% (d) Detection: 100%

Figure 10. Accurately detected Glide images. For illustration purposes cropped to a square aspect ratio.



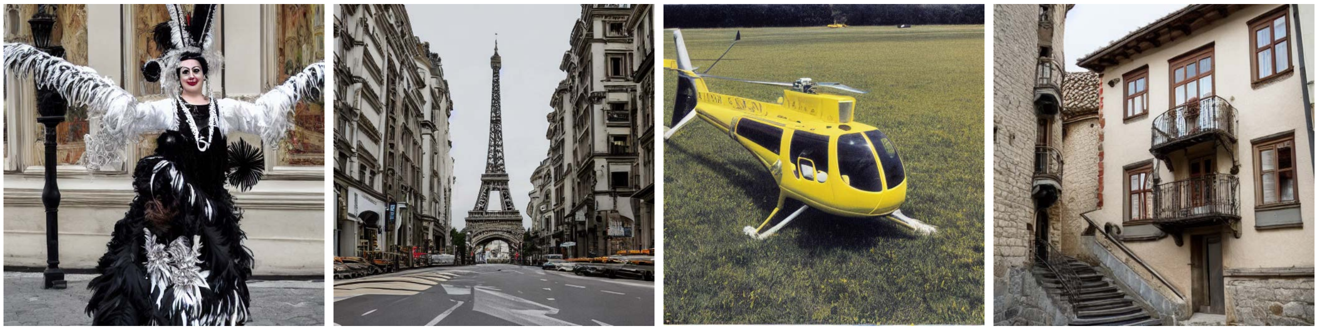
(a) Detection: 100%

(b) Detection: 100%

(c) Detection: 100%

(d) Detection: 100%

Figure 11. Accurately detected MidJourney-v5 images. For illustration purposes cropped to a square aspect ratio.



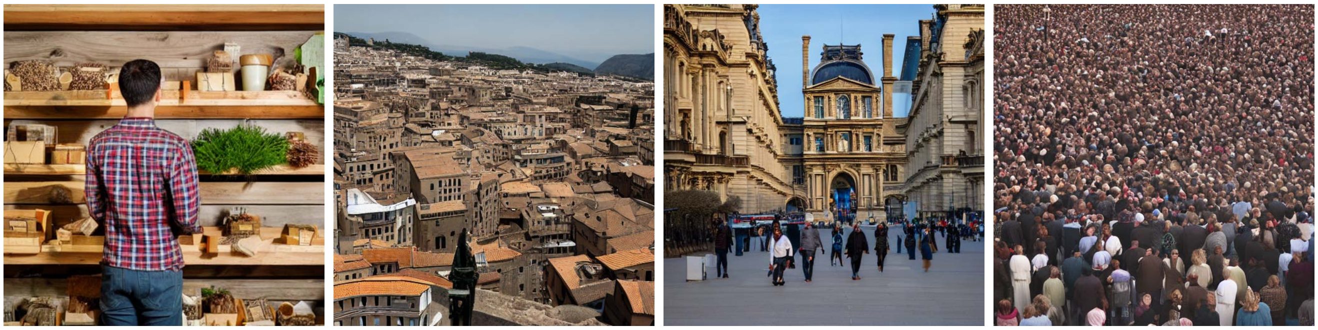
(a) Detection: 100%

(b) Detection: 100%

(c) Detection: 100%

(d) Detection: 100%

Figure 12. Accurately detected Stable Diffusion 1.3 images. For illustration purposes cropped to a square aspect ratio.



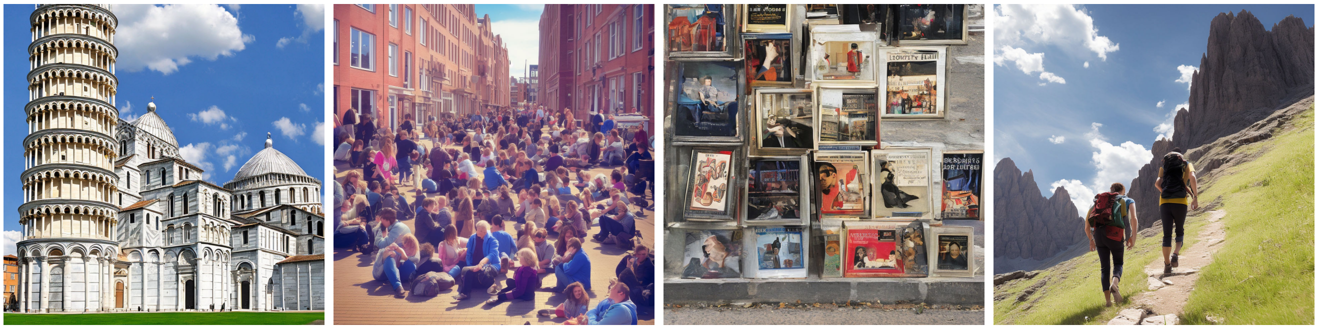
(a) Detection: 100%

(b) Detection: 100%

(c) Detection: 100%

(d) Detection: 100%

Figure 13. Accurately detected Stable Diffusion 1.4 images. For illustration purposes cropped to a square aspect ratio.



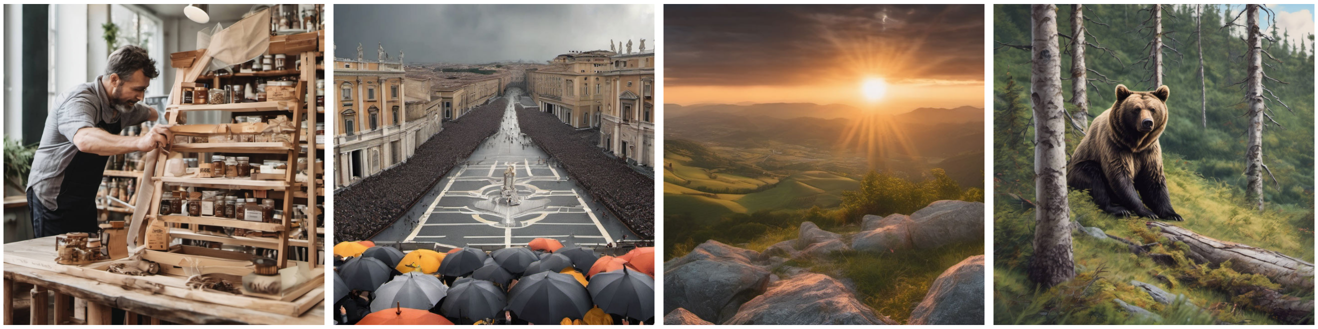
(a) Detection: 100%

(b) Detection: 98%

(c) Detection: 100%

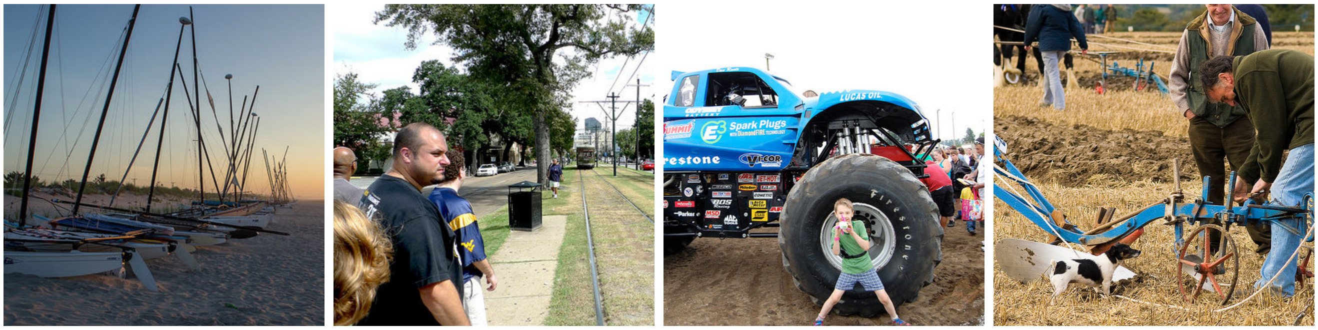
(d) Detection: 100%

Figure 14. Accurately detected Stable Diffusion 2 images. For illustration purposes cropped to a square aspect ratio.



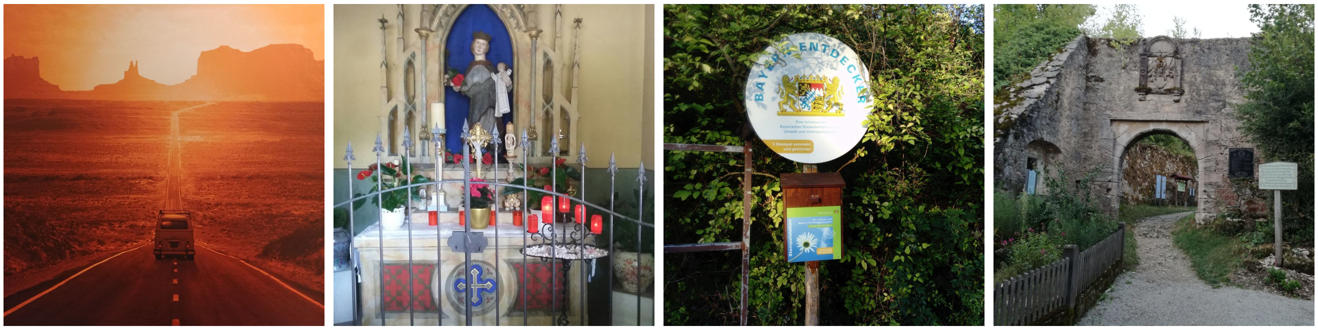
(a) Detection: 100% (b) Detection: 100% (c) Detection: 82% (d) Detection: 100%

Figure 15. Accurately detected Stable Diffusion XL images. For illustration purposes cropped to a square aspect ratio.



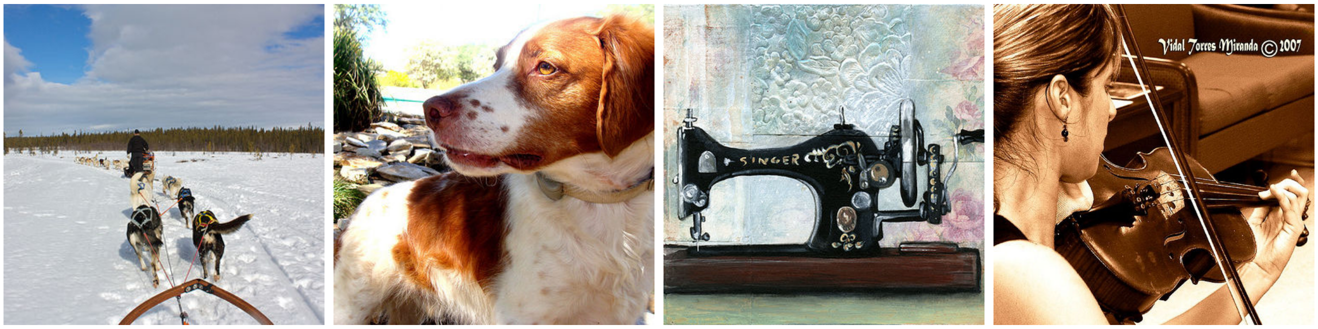
(a) Detection: 0% (b) Detection: 0% (c) Detection: 0% (d) Detection: 0%

Figure 16. Accurately classified real images from COCO. For illustration purposes cropped to a square aspect ratio.



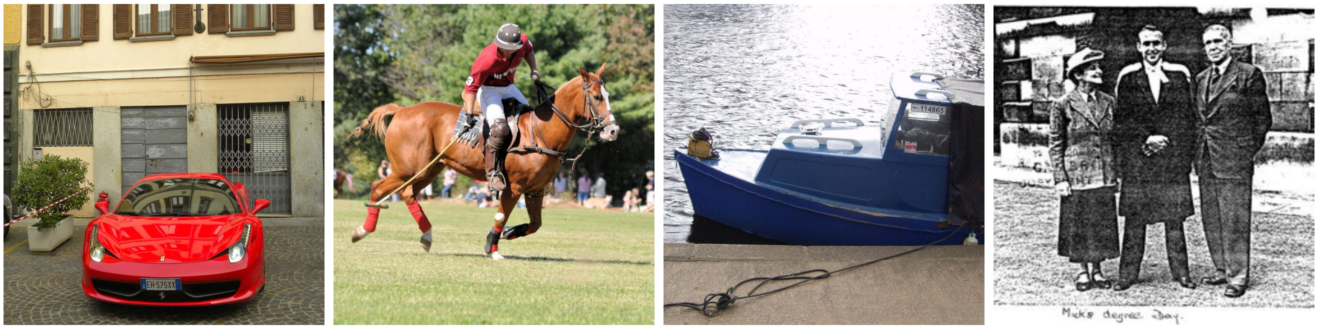
(a) Detection: 0% (b) Detection: 0% (c) Detection: 0% (d) Detection: 0%

Figure 17. Accurately classified real images from FODB. For illustration purposes cropped to a square aspect ratio.



(a) Detection: 0% (b) Detection: 0% (c) Detection: 0% (d) Detection: 0%

Figure 18. Accurately classified real images from ImageNet. For illustration purposes cropped to a square aspect ratio.



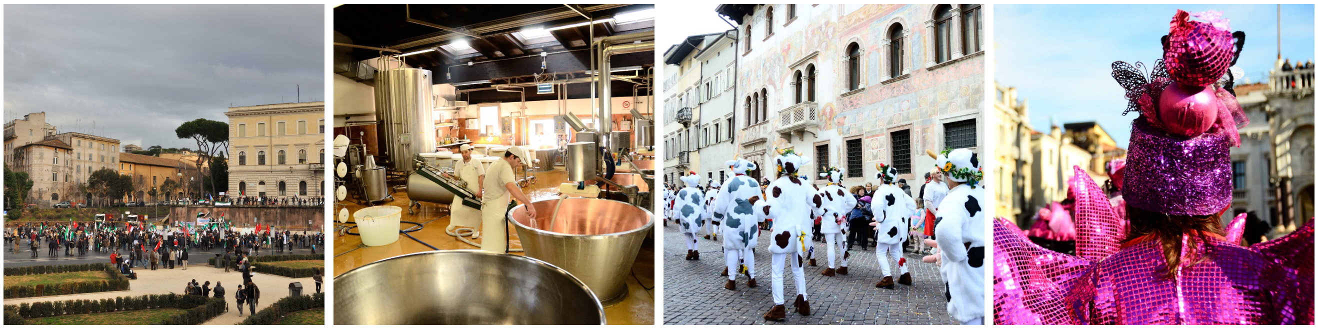
(a) Detection: 4%

(b) Detection: 9%

(c) Detection: 0%

(d) Detection: 0%

Figure 19. Accurately classified real images from Open Images. For illustration purposes cropped to a square aspect ratio.



(a) Detection: 0%

(b) Detection: 0%

(c) Detection: 0%

(d) Detection: 0%

Figure 20. Accurately classified real images from RAISE-1k. For illustration purposes cropped to a square aspect ratio.

Processing Systems, 35:16344–16359, 2022. 2

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[6] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[7] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pages 3247–3258. PMLR, 2020. 3

[8] GeroldMeisinger. Huggingface: Geroldmeisinger / laion2b-en-a65 cogvlm2-4bit captions, 2024. accessed 11th Nov. 2024. 1

[9] Benjamin Hadwiger and Christian Riess. The forchheim image database for camera identification in the wild. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part VI*, pages 500–515. Springer, 2021. 1

[10] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 1

[11] Yan Ju, Shan Jia, Lipeng Ke, Hongfei Xue, Koki Nagano, and Siwei Lyu. Fusing global and local features for generalized ai-synthesized image detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3465–3469. IEEE, 2022. 3

[12] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023. 1

[13] Christos Koutlis and Symeon Papadopoulos. Leveraging representations from intermediate encoder-blocks for synthetic image detection. In *Computer Vision – ECCV 2024*, pages 394–411, Cham, 2025. Springer Nature Switzerland. 3

[14] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020. 1

[15] LatentSpacer. Civitai: Dataset with 6000+ flux.1 dev images, 2024. accessed 11th Nov. 2024. 1

[16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *ArXiv e-prints*, pages arXiv–1607, 2016. 1

[17] 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: 13th European Conference,*

Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014. 1

[18] Zhengzhe Liu, Xiaojuan Qi, and Philip HS Torr. Global texture enhancement for fake face detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8060–8069, 2020. 3

[19] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[20] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24480–24489, 2023. 3

[21] saq1b. Huggingface: saq1b/midjourney-v6.1, 2024. accessed 11th Nov. 2024. 1

[22] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image generation models. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 3418–3432, 2023. 3

[23] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12105–12114, 2023. 3

[24] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28130–28139, 2024. 3

[25] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 2

[26] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 1

[27] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8695–8704, 2020. 3

[28] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22445–22455, 2023. 3

[29] Nan Zhong, Yiran Xu, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. Patchcraft: Exploring texture patch for efficient ai-generated image detection. *arXiv preprint arXiv:2311.12397*, pages 1–18, 2024. 3