DeepFakeDetector: A Multi-Branch Fusion Framework for Cross-Generator Detection of AI-Generated Images

Lukhsaan Elankumaran
McMaster University
elankuml@mcmaster.ca

Md Nafieu Hossain Alif
McMaster University
hossam49@mcmaster.ca

Zuhair Qureshi
McMaster University
quresz23@mcmaster.ca

Krish Bhagirath
McMaster University
bhagirak@mcmaster.ca

Oriana Rueckert
McMaster University
rueckero@mcmaster.ca

Vihaan Singhal
McMaster University
singhv74@mcmaster.ca

Andrew Wu
McMaster University
wua87@mcmaster.ca

***Abstract*—AI-generated imagery from diffusion and GAN-based models is now photorealistic, enabling misinformation, fraud, and non-consensual synthetic media. A key challenge for detectors is *cross-generator generalization*: models that overfit to generators seen during training often fail on new architectures. We present *DeepFakeDetector*, a multi-branch framework that combines complementary forensic cues: (i) a fine-tuned Vision Transformer (ViT) for global semantic consistency, (ii) a distilled DeiT model for efficient inference, (iii) a transfer-based EfficientNet-B0 CNN baseline for robust convolutional features, and (iv) a Gradient Field CNN that exploits structure-tensor coherence in image gradients. On an OpenFake subset, individual branches range from 72.69% (frozen ViT) to 87.90% (EfficientNet-B0) accuracy. We further evaluate on the AI-GenBench benchmark [1] and report fusion baselines via logistic regression stacking and a meta-classifier. Code: https://github.com/McMasterAI-Society/DeepFakeDetector. Models: https://huggingface.co/DeepFakeDetector. Demo: https://www.deepfake-detector.app/.**

I. Introduction

A. Motivation

The democratization of generative AI has dramatically lowered the barrier to producing photorealistic synthetic images. Modern text-to-image diffusion systems and fine-tuned community models can generate images that are difficult for humans to authenticate. This raises urgent concerns for media integrity in journalism, politics, and everyday social platforms. As detectors are deployed in user-facing applications, they must be robust to new generators and common post-processing (e.g., compression and resizing).

A core technical difficulty is that many detectors implicitly learn *generator-specific* artifacts and degrade sharply when tested on images from unseen generative families. Prior work shows that naive supervised training can fail to generalize across generative model eras, motivating methods that rely on transferable representations and robust cues rather than memorization [2], [3].

Goal. We aim to build a practical detector that (i) generalizes across generators, (ii) remains performant under common degradations, and (iii) is deployable in an interactive web pipeline. We approach this by fusing diverse forensic signals (semantic, gradient-coherence, and convolutional transfer features) into a single calibrated probability.

B. Related Works

Universal and cross-generator detection. CNNDetection [3] demonstrated that detectors trained on certain GAN images can transfer to other GAN architectures; however, diffusion models often exhibit different artifacts and shifts in frequency space [4]. UnivFD [2] showed that frozen foundation-model features (e.g., CLIP/ViT) can separate real and synthetic images via lightweight probes, improving generalization.

Diffusion-specific detection. DIRE [5] detects diffusion-generated images via reconstruction behavior under diffusion re-encoding; while strong, it is substantially more expensive than single-pass classifiers and less suited for interactive applications.

Patch and region strategies. Patch selection methods (SSP/ESSP) focus on low-texture regions where synthetic artifacts can be more apparent [6]. These methods offer interpretability but may degrade under heavy compression.

Benchmarking and protocols. Recent benchmarks emphasize evaluation protocols that better reflect deployment: (i) *generator holdout* or *temporal splits* to test generalization to future models, and (ii) systematic degradations (JPEG compression, resizing, blur) to simulate social-media pipelines. We follow this philosophy using AI-GenBench [1] (website: [7]) and additional cross-generator benchmarks such as GenImage [8] and Community Forensics [9].

C. Problem Definition

Let $\mathcal{D} = \{(x_i, y_i, g_i)\}_{i=1}^{N}$ be images $x_i \in \mathbb{R}^{H\times W\times 3}$ with labels $y_i \in \{0, 1\}$, where $y_i = 0$ denotes real and $y_i = 1$ denotes AI-generated; g_i indicates the generator family (or `unknown` for real images). We seek a detector $f_\theta : \mathcal{X} \to [0, 1]$ that outputs $\hat{p} = f_\theta(x)$.

Cross-generator requirement. Partition generator families into disjoint sets $\mathcal{G}_{\text{train}}$ and $\mathcal{G}_{\text{test}}$ with $\mathcal{G}_{\text{train}} \cap \mathcal{G}_{\text{test}} = \emptyset$. The goal is to maintain strong performance on unseen generators:

$$\text{AUROC}\big(f_\theta; \mathcal{G}_{\text{test}}\big) \approx \text{AUROC}\big(f_\theta; \mathcal{G}_{\text{train}}\big). \tag{1}$$

Robustness requirement. Given a set of degradations $\mathcal{T}$ (e.g., JPEG compression, blur, downsampling), we require stable performance:

$$\mathbb{E}_{t\sim\mathcal{T}}\Big[\text{AUROC}\big(f_\theta; t(x)\big)\Big] \geq \text{AUROC}\big(f_\theta; x\big) - \epsilon. \tag{2}$$

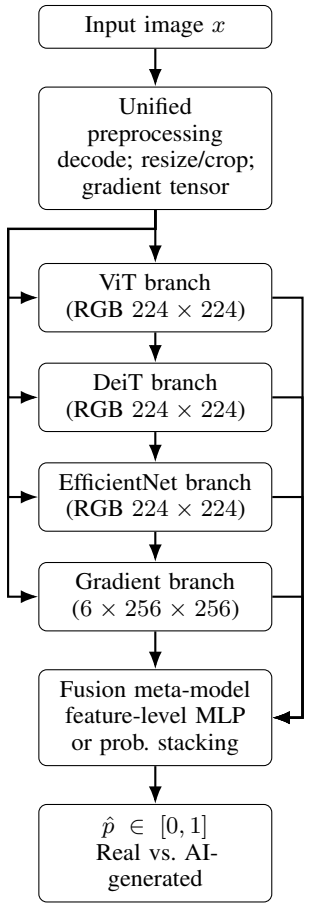

Fig. 1: DeepFakeDetector overview. A single input image is decoded and preprocessed, then passed through four expert branches whose output probabilities are fused into a calibrated prediction $\hat{p}$.

II. Methodology

A. Data

Primary dataset. Our primary corpus is OpenFake [10], a large-scale dataset pairing real images with synthetic counterparts generated by a diverse set of models. For rapid iteration, we trained and evaluated initial branches on curated subsets:

- ViT/DeiT subset: $N_{\text{train}} = 2040$, $N_{\text{test}} = 2420$, balanced.
- Gradient subset: $N_{\text{test}} = 2000$ (1,000 real / 1,000 fake), balanced.

We implement a deterministic sampling pipeline for small-scale experiments using WildFake (via ModelScope). Images are shuffled with fixed seeds, converted to RGB if required, and re-encoded as JPEG (quality=95) to standardize storage and decoding.

AI-GenBench benchmark split. For AI-GenBench [1], we use a single fixed random-seed split:

- Train: 14.4K × 2 (real/fake)
- Val: 3600 × 2 (real/fake)
- Test: approximately 4.94K total

Manual prompt-aligned dataset. We host a small, prompt-aligned manually generated dataset at https://huggingface.co/DeepFakeDetector/manual-gen-images. The dataset was generated using GPT-4o image generation, Gemini Nano Banana Pro, and approximately 27 Bing Image Creator images (usage limited). It follows the repository structure described in the project issue tracker, including prompt IDs and generator-specific subfolders.

Licensing. All datasets are strictly used for academic research. OpenFake is released under CC BY-SA 4.0, with additional non-commercial restrictions for subsets derived from proprietary generators [10]. No images are redistributed beyond permitted dataset hosting; only aggregate metrics are reported in this paper.

B. Unified Preprocessing

A single preprocessing pipeline produces branch-specific tensors from one input, preserving EXIF orientation. Luminance is computed via explicit BT.709 coefficients.

1) Transformer preprocessing: Images are resized and center-cropped to 224×224 and normalized with ImageNet statistics:

$$\hat{x} = \frac{x - \boldsymbol{\mu}}{\boldsymbol{\sigma}}, \tag{3}$$

$$\boldsymbol{\mu} = (0.485,\ 0.456,\ 0.406), \tag{4}$$

$$\boldsymbol{\sigma} = (0.229,\ 0.224,\ 0.225). \tag{5}$$

2) Gradient Field preprocessing: We compute the luminance L (Rec. 709):

$$L = 0.2126R + 0.7152G + 0.0722B, \tag{6}$$

compute Sobel gradients G_x, G_y, then form the (Gaussian-smoothed) structure tensor:

$$J = \begin{bmatrix} \mathcal{G}(G_x^2) & \mathcal{G}(G_x G_y) \\ \mathcal{G}(G_x G_y) & \mathcal{G}(G_y^2) \end{bmatrix}, \tag{7}$$

with eigenvalues $\lambda_1 \geq \lambda_2 \geq 0$ and coherence index:

$$\kappa = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2 \in [0, 1]. \tag{8}$$

We stack six channels: $\mathbf{T} = [G_x, G_y, |G|, \sin\theta, \cos\theta, \kappa] \in \mathbb{R}^{6\times256\times256}$. We use κ as a physically-motivated statistic: real images tend to have more coherent gradient structure than synthetic images.

C. Model Branches

DeepFakeDetector combines four experts that capture complementary evidence for AI-generated image detection. For an input image x, each branch $k \in \{\text{vit}, \text{deit}, \text{eff}, \text{grad}\}$ outputs a logit $z_k(x) \in \mathbb{R}$ and probability $p_k(x) = \sigma(z_k(x))$, where $\sigma(t) = \frac{1}{1+e^{-t}}$.

1) ViT Branch: **ViT.** We initialize from an ImageNet-21K pretrained ViT-Base/16 [11] and fine-tune for binary classification. Let $h_{\text{CLS}}(x) \in \mathbb{R}^{768}$ be the final `[CLS]` representation. A lightweight head maps it to a binary logit:

$$z_{\text{vit}}(x) = w_{\text{vit}}^{\top} \phi_{\text{vit}}(h_{\text{CLS}}(x)) + b_{\text{vit}}, \qquad p_{\text{vit}}(x) = \sigma(z_{\text{vit}}(x)). \tag{9}$$

The head ϕ_{vit} is a small MLP:

$$h_1 = \text{GELU}(W_1 \,\text{LN}(h_{\text{CLS}}(x)) + b_1), \tag{10}$$

$$h_2 = \text{GELU}(W_2\, h_1 + b_2), \tag{11}$$

$$\phi_{\text{vit}}(h_{\text{CLS}}(x)) = W_3 h_2 + b_3, \tag{12}$$

trained under both frozen-backbone and full fine-tuning regimes.

2) DeiT Branch: **DeiT.** We use DeiT-Base Distilled (patch16-224) [12], which adds a *distillation token*. Let $h_{\text{CLS}}(x)$ and $h_{\text{DIST}}(x)$ denote the final class and distillation token embeddings. Unlike standard ViT, DeiT produces two classifier outputs during training. The transformer outputs are processed by a lightweight MLP head:

$$\text{LayerNorm} \to \text{Linear}(512) \to \text{GELU} \to \text{Dropout}(0.2) \to \text{Linear}(2)$$

The model outputs:

$$z_{\text{deit}}(x) = w_{\text{deit}}^{\top} g(h_{\text{CLS}}(x), h_{\text{DIST}}(x)) + b_{\text{deit}},$$
$$p_{\text{deit}}(x) = \sigma(z_{\text{deit}}(x)). \tag{13}$$

Training is conducted under both frozen-backbone and full fine-tuning regimes. In deployment, DeiT can serve as an alternative to ViT in compute-constrained (CPU-centric) settings.

3) EfficientNet Branch: **EfficientNet.** As a convolutional baseline, we fine-tune EfficientNet-B0 pretrained on ImageNet [13]. Let $\psi_{\text{eff}}(\hat{x}) \in \mathbb{R}^d$ be the global pooled feature vector. A linear classifier yields:

$$z_{\text{eff}}(x) = w_{\text{eff}}^{\top} \psi_{\text{eff}}(\hat{x}) + b_{\text{eff}}, \qquad p_{\text{eff}}(x) = \sigma(z_{\text{eff}}(x)). \tag{14}$$

Optimization uses AdamW for 10 epochs with batch size 32.

4) Gradient Field CNN Branch: **Gradient Field CNN.** To capture subtle local structure cues, we employ a compact CNN operating on six-channel gradient tensors. These tensors include Sobel gradients (G_x, G_y), gradient magnitude $|G|$, orientation $(\sin\theta, \cos\theta)$, and coherence κ computed from the structure tensor:

$$\kappa(x) = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2 \in [0, 1], \tag{15}$$

$$T(x) = \left[G_x, G_y, |G|, \sin\theta, \cos\theta, \kappa\right] \in \mathbb{R}^{6\times 256\times 256}. \tag{16}$$

The tensor is processed by a compact CNN with feature widths $(16, 32, 64)$:

$$z_{\text{grad}}(x) = w_{\text{grad}}^{\top} \psi_{\text{grad}}(T(x)) + b_{\text{grad}}, \tag{17}$$

$$p_{\text{grad}}(x) = \sigma(z_{\text{grad}}(x)). \tag{18}$$

Motivation / Evolution. Prior to adopting the Gradient Field CNN, we explored a frequency-domain CNN, `CompactFFTNet`, trained on log-magnitude 2D FFTs of grayscale 256×256 images. It consisted of three convolutional blocks (16, 32, 64 filters) with batch normalization, ReLU, max pooling, and dropout, followed by global pooling and a 128-dimensional embedding feeding a binary classifier.

Although theoretically appealing for detecting frequency artifacts, `CompactFFTNet` underperformed (accuracy ∼77–79%, F1 ∼0.78 on OpenFake) compared to ViT/DeiT and EfficientNet-B0. We hypothesize that single-channel input and limited model capacity reduced its ability to capture subtle spatial–frequency patterns.

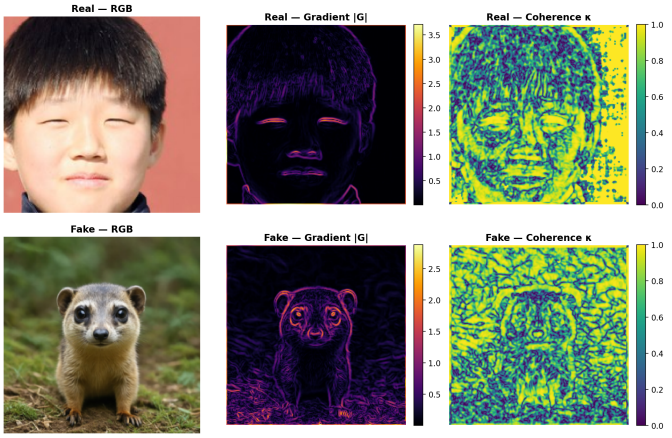

Fig. 2: Modality-specific visualizations used for model interpretability. Example panels include gradient/coherence-derived maps and model attribution overlays that support user-facing explanations.

D. Fusion

Each branch outputs a scalar probability $p_k(x) \in [0, 1]$. We combine these four probabilities using either logistic regression stacking or a shallow meta-classifier.

Let $\mathbf{p}(x) = [p_{\text{vit}}, p_{\text{deit}}, p_{\text{eff}}, p_{\text{grad}}] \in \mathbb{R}^4$ denote the concatenated probability vector.

Probability stacking (logistic regression). We apply logistic regression directly to the branch probabilities:

$$\hat{p} = \sigma\left(\sum_{k=1}^{4} w_k p_k + b\right), \tag{19}$$

which is fast, interpretable, and provides a strong baseline.

Meta-classifier fusion (MLP). Alternatively, we feed $\mathbf{p}(x)$ into a learned shallow network:

$$z_1 = \mathrm{ReLU}(W_1\mathbf{p}(x) + b_1) \in \mathbb{R}^{32}, \tag{20}$$

$$z_2 = \mathrm{ReLU}(W_2 z_1 + b_2) \in \mathbb{R}^{16}, \tag{21}$$

$$\hat{p} = \sigma(W_3 z_2 + b_3) \in [0, 1]. \tag{22}$$

This allows the fusion layer to learn non-linear combinations of branch predictions.

1) Fusion meta-models: We combine the four branch outputs using either (i) probability stacking or (ii) an MLP meta-classifier. For probability stacking (logistic regression):

$$p_{\text{fuse}}(x) = \sigma(\alpha_0 + \alpha_{\text{vit}} p_{\text{vit}}(x) + \alpha_{\text{deit}} p_{\text{deit}}(x) + \alpha_{\text{eff}} p_{\text{eff}}(x) + \alpha_{\text{grad}} p_{\text{grad}}(x)) . \tag{23}$$

For the meta-MLP fusion, we feed $p(x) = [p_{\text{vit}}, p_{\text{deit}}, p_{\text{eff}}, p_{\text{grad}}]$ into a learned network:

$$p_{\text{fuse}}(x) = \sigma(\mathrm{MLP}(p(x))) . \tag{24}$$

III. Results

A. Metrics

We report accuracy, precision, recall, F1, and AUROC. For forensic operating points, we additionally track calibration via Expected Calibration Error (ECE).

B. Branch Results on OpenFake Subsets

TABLE I: Branch model performance on OpenFake subset, $N_{\text{test}} = 2420$)

Model	Acc.	Prec.	Rec.	F1
ViT-Base	85.70%	0.860	0.857	0.857
DeiT-Base Distilled	85.58%	0.873	0.840	0.856
EfficientNet-B0	87.90%	0.879	0.880	0.879
Gradient Field CNN	82.00%	0.821	0.837	0.829

C. AI-GenBench Benchmark Evaluation (10% Subset)

AI-GenBench is an ongoing benchmark designed to evaluate AI-generated image detection under evolving generators and standardized protocols [1]. We follow a fixed random-seed split and report results on the held-out test set. The benchmark website provides additional protocol details and updates [7].

TABLE II: AI-GenBench (10% subset): benchmark results reported on the test split.

Model	Acc.	Prec.	Rec.	F1
ViT fine-tuned	90.85%	0.857	0.958	0.905
DeiT fine-tuned	87.50%	0.895	0.850	0.872
Gradient CNN fine-tuned	74.37%	0.819	0.683	0.745
EfficientNet-B0 fine-tuned	96.70%	0.998	0.973	0.990
Fusion (logistic regression)	84.46%	0.856	0.862	0.858
Fusion (meta-classifier MLP)	78.90%	0.816	0.794	0.804

D. Manual Generated Dataset Evaluation

We evaluate our models on a manually generated dataset containing images from Nano Banana, Bing Image Creator, and DALL-E to assess generalization to diverse contemporary generators. The dataset is publicly available at https://huggingface.co/DeepFakeDetector/manual-gen-images.

TABLE III: Manual Generated Dataset: branch model results on Nano Banana, Bing, and DALL-E images ($N_{\text{test}} = 212$).

Model	Acc.	Prec.	Rec.	F1
ViT-Base fine-tuned	80.66%	0.895	0.802	0.806
DeiT-Distilled fine-tuned	89.62%	0.857	0.941	0.897
EfficientNet-B0 fine-tuned	96.70%	0.998	0.973	0.990
Gradient Field CNN fine-tuned	74.56%	0.815	0.692	0.749

TABLE IV: Manual Generated Dataset: fusion model results ($N_{\text{test}} = 212$).

Model	AUROC
Fusion (logistic regression)	0.9941
Fusion (meta-classifier MLP)	0.9942

E. Calibration

We report calibration via Expected Calibration Error (ECE).

TABLE V: Calibration: Expected Calibration Error (ECE, 15 bins).

Model	ECE
Fusion (logreg)	0.029

F. Fusion ROC

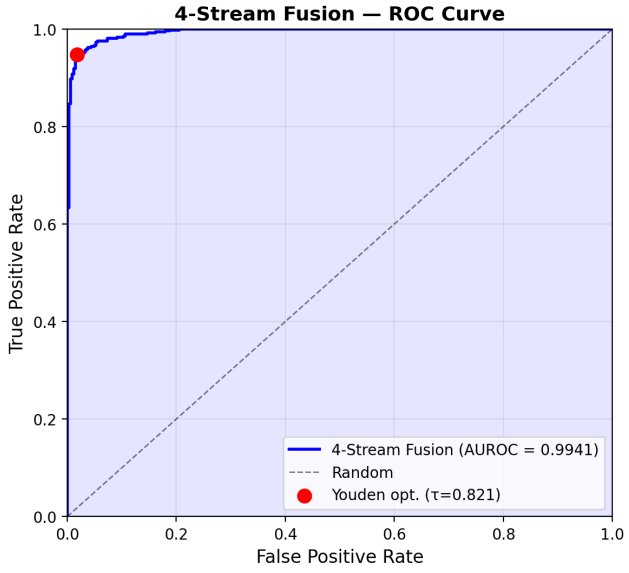

Fig. 3: ROC curves for individual branches and the 4-stream fusion model.

IV. Web Deployment

A web interface was implemented using a FastAPI backend and React frontend to demonstrate real-time inference and visualization of model predictions. The system loads model artifacts from Hugging Face and performs parallel inference across the four branches before fusion. The live demo is available at https://www.deepfake-detector.app/.

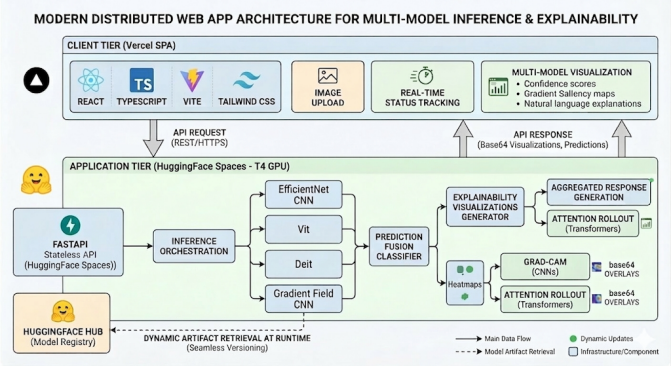

Fig. 4: Distributed deployment architecture for multi-model inference and explainability. The client (Vercel) sends an HTTPS request to a stateless FastAPI service (Hugging Face Spaces), which loads versioned model artifacts from Hugging Face Hub, runs submodels in parallel, fuses predictions, and returns probabilities alongside explainability overlays.

V. Conclusion

We presented DeepFakeDetector, a multi-branch framework that combines transformer-based semantic cues, convolutional transfer learning, and gradient-coherence signals for AI-generated image detection. On OpenFake subsets, individual branches range from 72.69% (frozen ViT) to 87.90% (EfficientNet-B0) accuracy. On the AI-GenBench benchmark, our models demonstrate strong baseline performance and reveal complementary signals across branches, as supported by ROC analysis. Finally, we demonstrated a production web deployment that dynamically serves versioned models from Hugging Face and provides user-facing explainability visualizations.

VI. Acknowledgements

This work was conducted by the McMaster AI Society. We thank the creators of OpenFake, DRAGON, WildFake, and AI-GenBench for releasing datasets and protocols to the research community [1], [10], [14], [15].

References

[1] L. Pellegrini, D. Cozzolino, S. Pandolfini, D. Maltoni, M. Ferrara, L. Verdoliva, M. Prati, and M. Ramilli, "Ai-genbench: A new ongoing benchmark for ai-generated image detection," in *2025 International Joint Conference on Neural Networks (IJCNN)*, 2025, pp. 1–9.

[2] U. Ojha, Y. Li, and Y. J. Lee, "Towards universal fake image detection exploiting vision foundation models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023.

[3] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. 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 (CVPR)*, Jun. 2020, pp. 8695–8704.

[4] R. Corvi, D. Cozzolino, G. Poggi, K. Nagano, and L. Verdoliva, "Intriguing properties of diffusion models: An empirical study of distribution shift in frequency space," in *CVPR Workshops (WMF)*, Jun. 2023.

[5] Z. Wang, J. Bao, W. Zhou, W. Wang, H. Hu, H. Chen, and H. Li, "Dire for diffusion-generated image detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2023.

[6] Z. Li *et al.*, "Simple patch selection for ai-generated image detection," *arXiv preprint*, 2024.

[7] "Ai-genbench benchmark website," https://mi-biolab.github.io/aigenbench-website/, accessed 2026-03-06.

[8] M. Zhu, H. Chen, Q. Yan, X. Huang *et al.*, "Genimage: A million-scale benchmark for detecting ai-generated image," *arXiv preprint*, 2023.

[9] J. Park and A. Owens, "Community forensics: Using thousands of generators to train fake image detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2025, pp. 8245–8257.

[10] ComplexDataLab *et al.*, "Openfake: An open dataset and platform toward real-world deepfake detection," *arXiv preprint*, 2025.

[11] 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," in *International Conference on Learning Representations (ICLR)*, 2021.

[12] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolges, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning (ICML)*, 2021.

[13] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[14] G. Bertazzini, D. Baracchi, D. Shullani, I. Echizen, and A. Piva, "Dragon: A large-scale dataset of realistic images generated by diffusion models," *arXiv preprint*, 2025.

[15] Y. Hong and J. Zhang, "WildFake: A large-scale and hierarchical dataset for AI-generated images detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 4, 2025, pp. 3500–3508.