

Multi-Model Forensic Ensemble and Partially Fine-Tuned CLIP Detector for Identifying AI-Generated Images: The Xabarnavis System Case Study

Solijonov Abduxoliq¹

¹ Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan

Abstract. Photorealistic images generated by generative models (Mid-journey, DALL·E, Flux, Stable Diffusion, GPT-image) are increasingly becoming a means of spreading disinformation in the mass-media information flow. This paper proposes a two-stage approach developed for the Xabarnavis media-forensics platform: (i) a degradation-robust binary detector in which the last 8 blocks of the CLIP ViT-L/14 encoder are fine-tuned, and (ii) an ensemble that performs weighted fusion of the outputs of 16 independent detectors. The model was trained on 134,108 images from the open-source OpenFake and CommunityForensics datasets (including 14,404 images from commercial generators) and evaluated on four held-out test sets: AUC 0.968 on new generators, AUC 0.958 on social-media images, and AUC 0.984 and 0.998 on the external GenImage++ and DRAGON datasets, respectively. The mathematical formulation of the task, loss function, temperature calibration, ensemble fusion formula, and model scheduling under constrained GPU memory are described.

Keywords: AI-generated image · Deepfake · Media forensics · CLIP · Vision Transformer · Ensemble · Calibration · Disinformation

1 Introduction

With the emergence of diffusion models, the ability to generate a high-quality “photograph” within seconds became widely accessible, raising ethical concerns about the deployment of generative AI [14]; generative models now synthesise images realistic enough to be used even as training data in medical imaging [17]. A fake image circulating on social media can be perceived as an authentic photograph taken at the scene of an event, creating a serious threat to journalism, forensic analysis, and information security [16]. Studies of user comments on social networks show how quickly such content can raise the level of social tension [5]. The main problem of existing detectors is generalization: a model trained on images from a single generator (for example, ProGAN or Stable Diffusion 1.4) may degrade sharply on new generators [4,11,18]. In addition, in real-

world use images are recompressed (JPEG), resized, and transmitted through messaging platforms, which removes low-level artifacts.

The contributions of this work are as follows:

- A degradation-robust, partially fine-tuned detector based on CLIP ViT-L/14 (Xabarnavis Image 0.2 and 0.3) and its complete mathematical model;
- A training corpus assembled from open datasets covering more than 1,300 generators, together with cross-dataset evaluation on four test sets not used during training;
- A comparative analysis of 16 detectors under identical conditions and a formal description of the ensemble fusion rule;
- A memory-aware scheduling algorithm that accelerates multi-model analysis on a GPU with 12 GB of memory.

2 Related Work

Wang et al. [18] showed that a ResNet-50 trained on ProGAN images with strong augmentation can generalize to other GAN models (CNNSpot). Ojha et al. [11] demonstrated that a linear classifier on top of CLIP features (UniversalFakeDetect) captures generator-independent cues in the semantic space. Corvi et al. [4] analyzed frequency traces for diffusion models. In recent years, AIDE [19], which combines semantic and low-level cues; Effort [20], which decomposes the CLIP space into orthogonal subspaces; and CO-SPY [3], which combines semantic and pixel-level features, have been proposed. Park and Owens [12] showed that generator diversity (more than 4,800 models) is a major factor in generalization. OpenFake [9] pairs real photographs on political and social topics with outputs from modern commercial generators, while DRAGON [2] and GenImage++ [21] are used as benchmarks for diffusion models. The Transformer architecture [6] and CLIP [13] form the basis of this work. Transformer-based classifiers have also proved effective for fine-grained image classification in other domains [1], and partial fine-tuning of pretrained backbones is an established transfer-learning strategy [8].

3 Problem Formulation and Mathematical Model

3.1 Classification Task

Let the image space be $\mathcal{X}$ and the label set be $y \in \{0, 1\}$, where $y = 1$ denotes an image generated by artificial intelligence and $y = 0$ denotes a real camera photograph. The detector is a function f_θ that maps an image to its probability of being AI-generated, and the decision is made using threshold τ :

$$\hat{y}(x) = \mathbb{I}[f_\theta(x) \geq \tau], \quad f_\theta : \mathcal{X} \rightarrow [0, 1]. \quad (1)$$

In real-world conditions, an image changes during distribution; therefore, the parameters are obtained by minimizing the expected error under a random degradation operator D :

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \mathbb{E}_{D \sim \mathcal{D}_{\text{deg}}} [\ell(f_{\theta}(D(x)), y)]. \quad (2)$$

Here, the operator D is a composition of JPEG compression (J_q), rescaling (R_s), and Gaussian blur (B_{σ}), and is applied identically to both classes:

$$D = J_q \circ R_s \circ B_{\sigma}, \quad q \sim \mathcal{U}\{45, \dots, 98\}, \quad s \sim \mathcal{U}[0.5, 0.95], \quad \sigma \sim \mathcal{U}[0.2, 1.2]. \quad (3)$$

Applying the same degradation to both classes is important: real photographs are usually stored as JPEG, whereas generated images are often stored as PNG. Otherwise, the model may “learn” the file format instead of generation traces. In addition, with 30 % probability, a social-media copy is simulated (720–1600 pixels, JPEG quality 68–92).

3.2 Detector Architecture

The input image is resized to 224×224 , divided into 256 patches of size 14×14 , and passed through the 24 Transformer blocks of the CLIP ViT-L/14 encoder [6, 13]:

$$\mathbf{h}_0 = [\mathbf{x}_{\text{cls}}; \mathbf{E}\mathbf{x}_1; \dots; \mathbf{E}\mathbf{x}_{256}] + \mathbf{E}_{\text{pos}}, \quad \mathbf{h}_l = \text{Block}_l(\mathbf{h}_{l-1}), \quad l = 1, \dots, 24. \quad (4)$$

The normalized state of the CLS token is transformed by a linear layer into two-class logits, where T is the temperature parameter:

$$\mathbf{z} = \text{LN}(\mathbf{h}_{24}^{\text{cls}}), \quad \mathbf{s} = \mathbf{W}\mathbf{z} + \mathbf{b} \in \mathbb{R}^2, \quad \mathbf{p} = \text{softmax}(\mathbf{s}/T). \quad (5)$$

To preserve pretrained semantic knowledge and prevent overfitting, the first 16 blocks are frozen; only the last 8 blocks, the final normalization layer, and the classifier are trained:

$$\frac{\partial \mathcal{L}}{\partial \theta_l} = 0 \quad (l \leq 16), \quad |\theta_{\text{train}}| = 100.8\text{M}, \quad |\theta| = 303.2\text{M}. \quad (6)$$

3.3 Loss Function and Optimization

Weighted cross-entropy accounting for class imbalance and label smoothing ($\varepsilon = 0.05$) are applied:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N w_{y_i} \sum_{c \in \{0,1\}} q_{ic} \log p_{ic}, \quad \mathbf{q}_i = (1 - \varepsilon)\mathbf{e}_{y_i} + \frac{\varepsilon}{2}, \quad w_c = \frac{N}{2N_c}. \quad (7)$$

Optimization is performed using AdamW [10]: $\eta_0 = 5 \cdot 10^{-4}$ for the classifier and $\eta_0 = 2 \cdot 10^{-5}$ for the encoder blocks, with weight decay 0.05. The learning rate follows linear warm-up and cosine decay:

$$\eta(t) = \begin{cases} \eta_0 \frac{t+1}{t_w}, & t < t_w, \\ \eta_0 \left[0.02 + 0.49 \left(1 + \cos \frac{\pi(t-t_w)}{t_{\max} - t_w} \right) \right], & t \geq t_w. \end{cases} \quad (8)$$

Instead of an exponential moving average of the parameters, the epoch with the best validation AUC is selected; training stops if AUC does not improve for two consecutive epochs.

3.4 Test-Time Augmentation and Calibration

During inference, logits from the image and its horizontal flip are averaged, after which the optimal temperature T^* determined on the validation set $\mathcal{V}$ is applied [7]:

$$\mathbf{g}(x) = \frac{1}{2} (\mathbf{s}(x) + \mathbf{s}(\text{flip}(x))), \quad f_\theta(x) = \text{softmax}(\mathbf{g}(x)/T^*)_1, \quad (9)$$

$$T^* = \arg \min_{T>0} - \sum_{i \in \mathcal{V}} \log \text{softmax}(\mathbf{g}(x_i)/T)_{y_i}. \quad (10)$$

The area under the ROC curve (AUC), accuracy, balanced accuracy, and expected calibration error (ECE) are used as quality metrics:

$$\text{ECE} = \sum_{b=1}^B \frac{|B_b|}{N} |\text{acc}(B_b) - \text{conf}(B_b)|, \quad \text{AUC} = P(f(x^+) > f(x^-)). \quad (11)$$

3.5 Ensemble Fusion Rule

In the Xabarnavis platform, the final decision is based not on a single model but on a set of detectors. The vote a_0 of the primary model and its assigned weight w_0 are amplified threefold, while the remaining pretrained detectors in set S are added using their registry weights w_j :

$$A = \frac{3w_0a_0 + \sum_{j \in S} w_j a_j}{3w_0 + \sum_{j \in S} w_j}, \quad R = 1 - A, \quad \hat{y} = \mathbb{I}[A \geq R]. \quad (12)$$

An important point is that the “real” score R is not computed independently but is derived from the fused score A . During the study, when R was taken only from the primary model, a single outlier detector ($a_0 = 0.999$) overrode the “real” votes of 11 out of 15 detectors; after switching to the formulation in (12), this error was eliminated.

3.6 Scheduling Under Limited GPU Memory

The analysis contains 23 models, 13 of which are launched in separate processes and reload their weights each time. The GPU memory consumption c_i was measured for model i (for example, CO-SPY: 5.1 GB; AIDE: 4.5 GB). At any time t , the set of simultaneously running models $\mathcal{R}(t)$ must satisfy the following condition:

$$\max_{\mathcal{R}(t)} |\mathcal{R}(t)| \quad \text{s.t.} \quad \sum_{i \in \mathcal{R}(t)} c_i \leq C = M_{\text{free}} - r, \quad |\mathcal{R}(t)| \leq K. \quad (13)$$

Here, M_{free} is the actual free memory after clearing the PyTorch cache, r is the reserve, and K is the number of parallel workers. The problem is solved with a “largest-first” greedy algorithm: among the queued models, the largest model that fits is launched, while smaller models fill the remaining space. As a result, analysis time for one image decreased from ~ 76 s to ~ 51 s ($1.5\times$), while model outputs remained bit-for-bit identical.

4 Dataset

Table 1 presents the training corpus. The Image 0.2 set contains 1,293 unique generators; Image 0.3 adds 14,404 images from commercial generators such as Midjourney 5.2/6.1, Adobe Firefly 2/3, Imagen 3, Ideogram 1/2, DALL·E 2/3, and Flux. Real photographs were obtained from LAION, Pexels, COCO, FFHQ, VISION (smartphone cameras), and LandscapesHQ. Large images were down-scaled using the same rule for both classes (long side 1024 pixels).

Table 1: Composition of the training set

Source	Real	AI	Used in model
CommunityForensics-Small [12]	24,702	24,998	Image 0.2, 0.3
OpenFake [9]	35,003	35,001	Image 0.2, 0.3
CommunityForensics Commercial [12]	0	14,404	Image 0.3 only

Evaluation was performed on four test sets: OpenFake test (new generators unseen during training: GPT-image-1.5, Midjourney 7, Flux.2, Sora-2 frames, and others), OpenFake Reddit (real-world social-media images), Gen-Image++ [21] (Flux and SD3 photorealistic and “amateur photo” styles), and DRAGON [2]. The latter two do not overlap with the training sources of any model and measure cross-dataset generalization.

5 Experimental Setup

Experiments were conducted on a workstation with an NVIDIA GeForce RTX 5070 (12 GB), 64GB RAM, and a 24-thread processor, using PyTorch 2.11

(CUDA 12.8) and timm 1.0. Mixed precision (bfloat16), gradient-norm clipping to 1.0, batch size 32, 500 warm-up steps, and 6 epochs were used. One epoch of Image 0.3 took ~ 16 minutes. For comparative evaluation of the detectors, 75 + 75 images were randomly selected from the OpenFake test set, and 30 + 30 images (real + AI) were selected from each of GenImage++ and DRAGON. Because every image was processed by 17 detectors, this sample size was limited by computation time; the Xabarnavis Image models, however, were evaluated on the full test sets.

6 Results and Discussion

Table 2 presents the results of the Xabarnavis Image models on all test sets. Image 0.2 achieved AUC 0.968 and accuracy 91.5 % on new generators not seen during training, and AUC 0.958 and accuracy 89.5 % on real social-media images. With commercial generators added, Image 0.3 changed from AUC 0.968 to 0.970 on the OpenFake test and from AUC 0.958 to 0.948 on the Reddit set. On GenImage++ (Flux/SD3 photorealistic), which does not overlap with any training source, Image 0.2 achieved AUC 0.987 (accuracy 95.3 %) and Image 0.3 achieved AUC 0.984 (accuracy 93.6 %), whereas the EfficientNet-B0-based Image 0.1 reached only AUC 0.630. On the DRAGON test set (25 diffusion models), also disjoint from all training sources, Image 0.2 achieved AUC 0.994 (accuracy 96.2 %), Image 0.3 achieved AUC 0.998 (accuracy 96.4 %), and Image 0.1 reached only AUC 0.745.

Table 2: Xabarnavis Image models: AUC / accuracy (threshold 0.5), full test sets

Model	OpenFake test	OpenFake Reddit	GenImage++	DRAGON test
Image 0.1 (EfficientNet-B0)	0.704 / 55.5 %	0.398 / 41.9 %	0.630 / 56.5 %	0.745 / 66.1 %
Image 0.2	0.968 / 91.5 %	0.958 / 89.5 %	0.987 / 95.3 %	0.994 / 96.2 %
Image 0.3	0.970 / 88.9 %	0.948 / 87.7 %	0.984 / 93.6 %	0.998 / 96.4 %

Thus, the primary source of cross-dataset generalization is the semantic representation of the CLIP encoder together with a diverse training corpus containing more than 1,300 generators; adding commercial images provided no substantial additional benefit on these sets. The calibrated temperature was $T^* = 0.868$; ECE on validation was 0.016. The training process is shown in Fig. 1.

The ablation analysis revealed an important result: after adding commercial images, the AI-image detection rate increased from 92.8 % to 94.8 %, but correct recognition of real photographs decreased from 90.3 % to 82.9 %; consequently, accuracy at the 0.5 threshold decreased while AUC remained almost unchanged. The reason is that the additional 14.4 thousand images belonged only to the AI class and were not accompanied by matched real photographs: the class prior changed and the decision boundary shifted toward “AI”. Therefore, when adding

new generators, they should be paired with topically matched real images or the threshold should be reselected on the validation set.

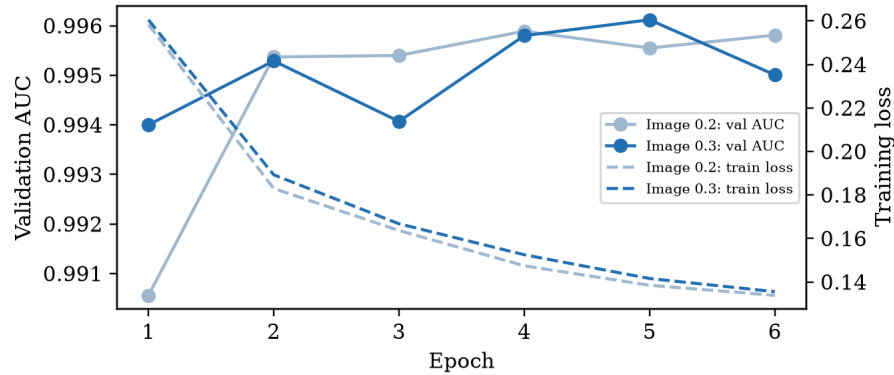


Fig. 1: Training process: validation AUC (solid) and training loss (dashed)

As shown in Table 3, among individual detectors Xabarnavis Image 0.2 achieved the strongest result (AUC 0.949, accuracy 93.3 %). Among open scientific detectors, CO-SPY and Effort performed well, whereas UniversalFakeDetect and DMImageDetection performed below random level on modern generators because they were trained on older GANs and early diffusion models. The ensemble had the highest AUC (0.976), meaning that it ranked images best by probability, but its accuracy at the 0.5 threshold was only 57.3 % and ECE was 0.235. This indicates that the weights and threshold in (12) need to be recalibrated for current generators: to convert good ranking ability into correct decisions, temperature or logistic calibration should also be applied to the ensemble output.

Table 4 clearly demonstrates detector specialization. For example, SAFE achieved AUC 1.000 on GenImage++, but only AUC 0.551 on the OpenFake test dominated by modern commercial generators. AIDE performed strongly on both external datasets. Because of the small sample size ($n = 60$), detectors should be considered statistically comparable when differences are smaller than 0.01–0.02. The fact that no single detector is best across all distributions is a key argument for the ensemble approach, consistent with the gains reported for ensemble learners in other recognition tasks [15].

By generator, the most difficult cases were (Image 0.2 $\rightarrow$ Image 0.3, AI-image detection rate; Fig. 2): recraft-v3, 43 % $\rightarrow$ 60 %; halfmoon-4-4-25, 54 % $\rightarrow$ 85 %; frames-23-1-25, 73 % $\rightarrow$ 95 %; ideogram-2.0, 78 % $\rightarrow$ 97 %; midjourney-7, 83 % $\rightarrow$ 92 %; z-image-turbo, 92 % $\rightarrow$ 93 %. Image 0.3 improved substantially on these commercial generators, which is the main benefit of adding commercial images. However, correct recognition of real images also changed (Image 0.2 $\rightarrow$ Image 0.3): imagenet 88.3 % $\rightarrow$ 78.5 %, docci 94.4 % $\rightarrow$ 91.9 %; thus, the gain came at the cost of more false positives on real photographs and requires threshold reselection.

Table 3: Comparative evaluation of detectors (OpenFake test, 75 + 75 images)

Detector	AUC	Accuracy	ECE
Xabarnavis Ensemble (Section 3.5)	0.976	57.3 %	0.235
Xabarnavis Image 0.2	0.949	93.3 %	0.055
CO-SPY	0.919	74.0 %	0.151
Effort	0.915	77.3 %	0.115
Xabarnavis 0.5	0.903	81.3 %	0.184
AIDE	0.839	69.3 %	0.210
Xabarnavis 0.7	0.759	57.3 %	0.189
Xabarnavis 0.2	0.729	65.3 %	0.230
Xabarnavis Image 0.1	0.725	56.0 %	0.370
FatFormer	0.719	54.7 %	0.457
Xabarnavis 0.3	0.716	55.3 %	0.431
Xabarnavis 0.1	0.713	62.7 %	0.338
Xabarnavis 0.6	0.660	62.0 %	0.111
SAFE	0.551	62.7 %	0.335
Xabarnavis 0.4	0.457	50.0 %	0.498
UniversalFakeDetect	0.411	48.0 %	0.507
DMImageDetection	0.370	50.0 %	0.502

Table 4: Cross-dataset evaluation (30 + 30 images): top 8 detectors

Test	Detector	AUC	Accuracy
GenImage++	SAFE	1.000	100.0 %
GenImage++	AIDE	0.998	96.7 %
GenImage++	Xabarnavis Image 0.3	0.996	93.3 %
GenImage++	Xabarnavis Image 0.2	0.989	95.0 %
GenImage++	Ensemble	0.981	75.0 %
GenImage++	Effort	0.924	83.3 %
GenImage++	CO-SPY	0.807	70.0 %
GenImage++	Xabarnavis 0.3	0.803	71.7 %
DRAGON test	Ensemble	0.999	95.0 %
DRAGON test	AIDE	0.998	91.7 %
DRAGON test	Xabarnavis Image 0.3	0.998	91.7 %
DRAGON test	Xabarnavis 0.3	0.997	96.7 %
DRAGON test	Xabarnavis Image 0.2	0.994	96.7 %
DRAGON test	SAFE	0.983	91.7 %
DRAGON test	Effort	0.957	80.0 %
DRAGON test	DMImageDetection	0.881	65.0 %

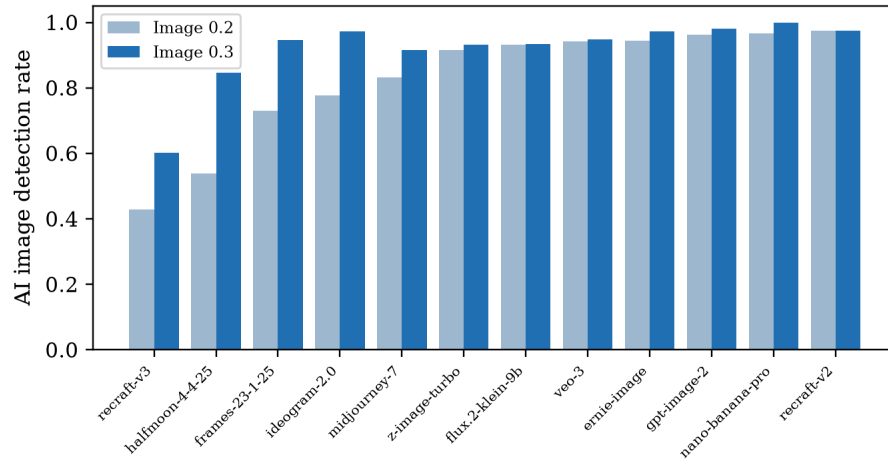


Fig. 2: AI-image detection rate for the most difficult generators in the OpenFake test

6.1 Practical Example

The platform was tested on a phone photograph taken at the CyberCryptUz event held at Tashkent University of Information Technologies. Of 15 AI detectors, 11 classified the image as real; according to the fused formula (12), the probability of authenticity was 66.8%, the probability of AI generation was 33.2%, and the probability of manipulation was 12.7%. During analysis, it was found that one outlier detector could shift the final conclusion in the wrong direction, and the correction described in Section 3.5 was introduced.

7 Limitations

- The output is a probabilistic estimate rather than absolute proof; important decisions should be made together with source information, metadata, and expert assessment.
- The comparative detector evaluation was performed on 150 images (OpenFake) and 60 images (GenImage++, DRAGON); the approximate 95% confidence intervals for AUC are ± 0.03 and ± 0.05 , respectively. The Xabarnavis Image model results were obtained on 2,000–10,000 images.
- The training corpus includes sources licensed under CC-BY-NC; model weights are distributed for research purposes.
- Recraft, Ideogram, and the latest Midjourney versions remain difficult classes.

8 Conclusion

This work proposed a mathematical model of a partially fine-tuned CLIP ViT-L/14 detector and a multi-model ensemble for the Xabarnavis platform. Image 0.2 achieved AUC 0.968 (accuracy 91.5%) on new generators not used during training and AUC 0.958 (accuracy 89.5%) on social-media images, outperforming all open detectors tested on the modern-generator benchmark; on GenImage++ and DRAGON, it performed at a level comparable to AIDE and SAFE. Both CLIP models preserved strong generalization on completely external datasets (Image 0.3: GenImage++ AUC 0.984, DRAGON AUC 0.998); however, adding unmatched images from commercial generators shifted the decision threshold toward “AI” and increased false positives on real photographs. The ensemble achieved the highest ranking quality (AUC 0.976), but its decision threshold and weights require calibration. GPU-memory-aware scheduling reduced analysis time by $1.5\times$. Future work will focus on calibrating the ensemble with logistic regression on the validation set, increasing the share of the newest commercial generators (Recraft, Ideogram, Midjourney 7), and creating a dataset of real images from the Uzbek media environment.

References

1. Agarwal, A.K., Hamid, A.B.B.A., Ather, D., Tiwari, R.G., De, I., Krishna, K.R.: Lesion-aware ordinal transformer for diabetic retinopathy classification from fundus images. *Biomedical and Pharmacology Journal* **19**(1), 233–245 (2026). <https://doi.org/10.13005/bpj/3349>
2. Bertazzini, G., Baracchi, D., Shullani, D., Echizen, I., Piva, A.: DRAGON: A large-scale dataset of realistic images generated by diffusion models. *arXiv preprint* (2025). <https://doi.org/10.48550/arXiv.2505.11257>
3. Cheng, S., et al.: CO-SPY: Combining semantic and pixel features to detect synthetic images by AI. In: *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2025)
4. Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., Verdoliva, L.: On the detection of synthetic images generated by diffusion models. In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 1–5. IEEE (2023). <https://doi.org/10.1109/ICASSP49357.2023.10095167>
5. Donchenko, D., Ovchar, N., Sadovnikova, N., Parygin, D., Shabalina, O., Ather, D.: Analysis of comments of users of social networks to assess the level of social tension. *Procedia Computer Science* **119**, 359–367 (2017). <https://doi.org/10.1016/j.procs.2017.11.195>
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16×16 words: Transformers for image recognition at scale. In: *Proc. International Conference on Learning Representations (ICLR)* (2021). <https://doi.org/10.48550/arXiv.2010.11929>
7. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: *Proc. International Conference on Machine Learning (ICML)*. pp. 1321–1330 (2017). <https://doi.org/10.48550/arXiv.1706.04599>

8. Gupta, S., Abdul Hamid, A.B., Nyamasvisva, T.E., Jain, V., Tyagi, N., Mun, N.K., Ather, D.: Cotton disease prediction using deep transfer learning: Comparative analysis of ResNet50, VGG16 and InceptionV3 models. *International Journal of Engineering, Science and Information Technology* **5**(3), 588–594 (2025). <https://doi.org/10.52088/ijesty.v5i3.1116>
9. Livernoche, V., Arodi, A., Musulan, A., Yang, Z., Salvail, A., Caron, G., Godbout, J.-F., Rabbany, R.: OpenFake: An open dataset and platform toward large-scale deepfake detection. *arXiv preprint* (2025). <https://doi.org/10.48550/arXiv.2509.09495>
10. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: *Proc. International Conference on Learning Representations (ICLR)* (2019)
11. Ojha, U., Li, Y., Lee, Y.J.: Towards universal fake image detectors that generalize across generative models. In: *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 24480–24489 (2023). <https://doi.org/10.48550/arXiv.2302.10174>
12. Park, J., Owens, A.: Community Forensics: Using thousands of generators to train fake image detectors. In: *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2025)
13. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: *Proc. International Conference on Machine Learning (ICML)*. pp. 8748–8763 (2021)
14. Saleem, M., Chawla, C., Ather, D.: Ethical considerations in the deployment of generative AI systems in academia. In: *Generative AI: Current Trends and Applications. Studies in Computational Intelligence*, vol. 1177, pp. 245–256. Springer, Singapore (2024). https://doi.org/10.1007/978-981-97-8460-8_12
15. Sattarov, K., Shakhriyorbek, B., Chaudhary, N., Ather, D., Kler, R., Shukla, B.: Automated sentiment analysis and emotion recognition of Steam game reviews using ensemble machine learning techniques. In: *2024 1st International Conference on Sustainable Computing and Integrated Communication in Changing Landscape of AI (ICSCAI)*. pp. 1–10. IEEE (2024). <https://doi.org/10.1109/ICSCAI61790.2024.10867053>
16. Shukla, R.S., Ather, D., Raj, D., Jain, A., Sarkar, T., Rakhra, M.: A secure and intelligent framework for cyber threat detection using machine learning. *Journal of Discrete Mathematical Sciences and Cryptography* **29**(6), 2441–2457 (2026). <https://doi.org/10.47974/JDMSC-2752>
17. Tripathi, R.P., Khatri, S.K., Van Greunen, D., Ather, D.: Enhancing breast cancer diagnosis through segmentation-driven generative adversarial networks for synthetic mammogram generation. In: *Proc. 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS)*. pp. 1078–1082. IEEE (2023). <https://doi.org/10.1109/ICTACS59847.2023.10389935>
18. Wang, S.-Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: CNN-generated images are surprisingly easy to spot... for now. In: *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 8692–8701 (2020). <https://doi.org/10.1109/CVPR42600.2020.00872>
19. Yan, S., Li, O., Cai, J., Hao, Y., Jiang, X., Hu, Y., Xie, W.: A sanity check for AI-generated image detection. In: *Proc. International Conference on Learning Representations (ICLR)* (2025). <https://doi.org/10.48550/arXiv.2406.19435>
20. Yan, Z., et al.: Orthogonal subspace decomposition for generalizable AI-generated image detection (Effort). In: *Proc. International Conference on Machine Learning (ICML)* (2025)

21. Zhou, Y., He, X., Lin, K., Fan, B., Ding, F., Li, B.: Breaking latent prior bias in detectors for generalizable AIGC image detection. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 34606–34636 (2025). <https://doi.org/10.52202/085713-1030>