IA Image Generated Detection

By Ángel Pérez Castro

Executive Summary

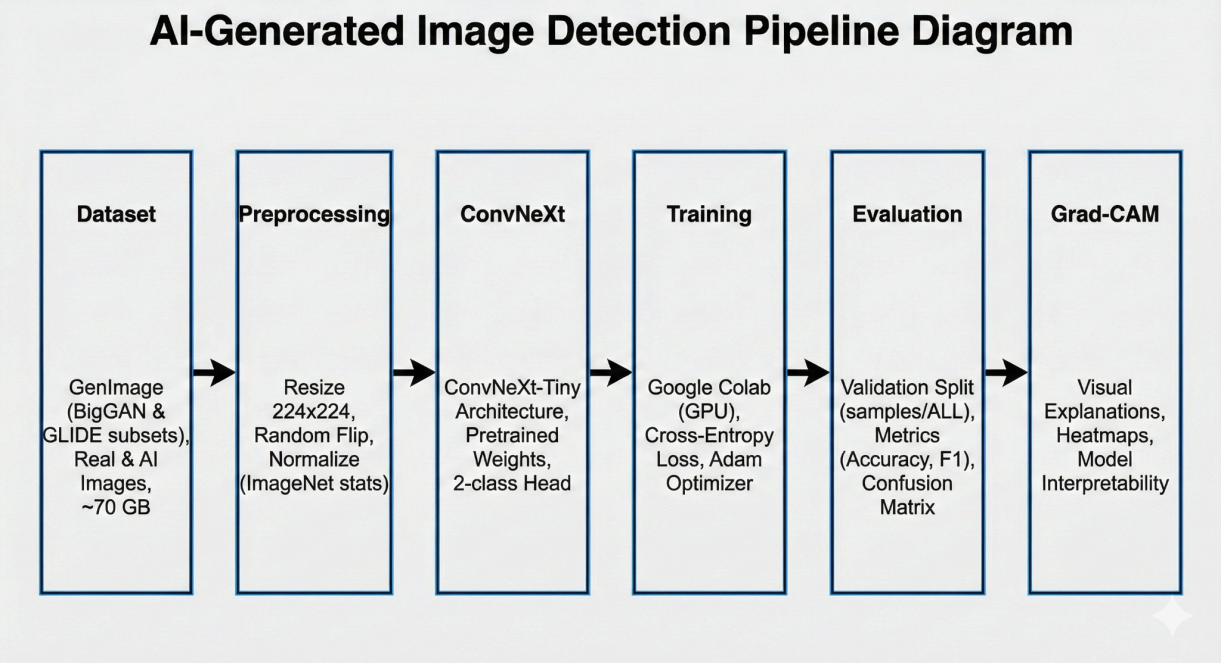
The rapid advancement of generative artificial intelligence has significantly increased the difficulty of distinguishing between real photographs and AI-generated images. This project presents a **complete, modular, and reproducible pipeline** for AI-generated image detection using deep learning techniques. A **ConvNeXt-Tiny** convolutional neural network is employed to classify images as *real* or *AI-generated*, leveraging subsets of the large-scale **GenImage** dataset.

The full GenImage dataset comprises images from **8 different generative models**, totaling approximately **400 GB**, which makes full-scale training impractical under typical academic hardware constraints. Consequently, the final model was trained in **Google Colab** using two representative generators (**BigGAN** and **GLIDE**), accounting for approximately **70 GB** of data. To enable this training, **additional cloud computing resources** were contracted. Furthermore, the project integrates **Grad-CAM** explainability techniques to provide visual insight into the model's decision-making process.

---

1. Introduction

The emergence of high-quality generative models has introduced new challenges in digital media verification, misinformation detection, copyright enforcement, and forensic analysis. Modern generative adversarial networks and diffusion models are capable of producing images that are perceptually indistinguishable from real-world photographs, even to expert human observers.



Within this context, the ability to automatically detect AI-generated content has become a critical research problem. This project addresses this challenge by designing and implementing a deep learning-based image classifier capable of distinguishing between real and synthetic images. The work emphasizes not only classification performance, but also **reproducibility, scalability, and interpretability**, which are essential aspects of real-world deployment.

The primary objectives of the project are:

- To analyze and integrate a state-of-the-art convolutional architecture for image classification.
- To manage and process an extremely large dataset under realistic resource constraints.
- To train and evaluate a robust AI-generated image detector using cloud-based infrastructure.
- To incorporate explainability methods that improve transparency and trust in the model.

---

2. Dataset Description

2.1 GenImage Dataset

The **GenImage dataset** is a large-scale benchmark specifically designed for the task of AI-generated image detection. It contains images produced by multiple state-of-the-art generative models, as well as real images collected from natural image distributions.

Key characteristics of the full dataset are:

- **Number of generators:** 8
- **Approximate total size:** 400 GB
- **Classes:** AI-generated images and real images
- **Data split:** Predefined training and validation subsets per generator

While the dataset provides excellent coverage of generative diversity, its size poses substantial challenges in terms of storage, memory, and computational requirements.

2.2 Generator Selection for Final Training

Due to the aforementioned constraints, the final training phase focuses on two representative generators:

- **BigGAN**, a large-scale GAN-based image generator
- **GLIDE**, a text-guided diffusion-based image generation model

These generators were selected to capture fundamentally different generative paradigms, thereby increasing the likelihood of learning discriminative features that generalize beyond a single model family. The combined size of these two subsets is approximately **70 GB**, which represents a practical compromise between dataset diversity and computational feasibility.

2.3 Evaluation Subset (samples/ALL)

For evaluation, visualization, and metric computation, a reduced subset of the dataset was constructed in `data/samples/ALL`. This subset was created by uniformly sampling:

- 50 images per class (AI / Real)
- For both training and validation splits
- Across all 8 generators

This results in a total of **1,600 images**, which were exclusively used to generate the plots and metrics stored in `results/figures` and `results/metrics`. This approach ensures fast, reproducible evaluation without biasing the final training process.

---

3. Data Preprocessing

All images are resized to **224 × 224 pixels**, following the standard input resolution used by ImageNet-pretrained models. The preprocessing pipeline includes:

- Spatial resizing
- Random horizontal flipping (training only)
- Tensor conversion
- Channel-wise normalization using ImageNet statistics

These transformations ensure compatibility with ConvNeXt pretrained weights and contribute to stable and efficient training.

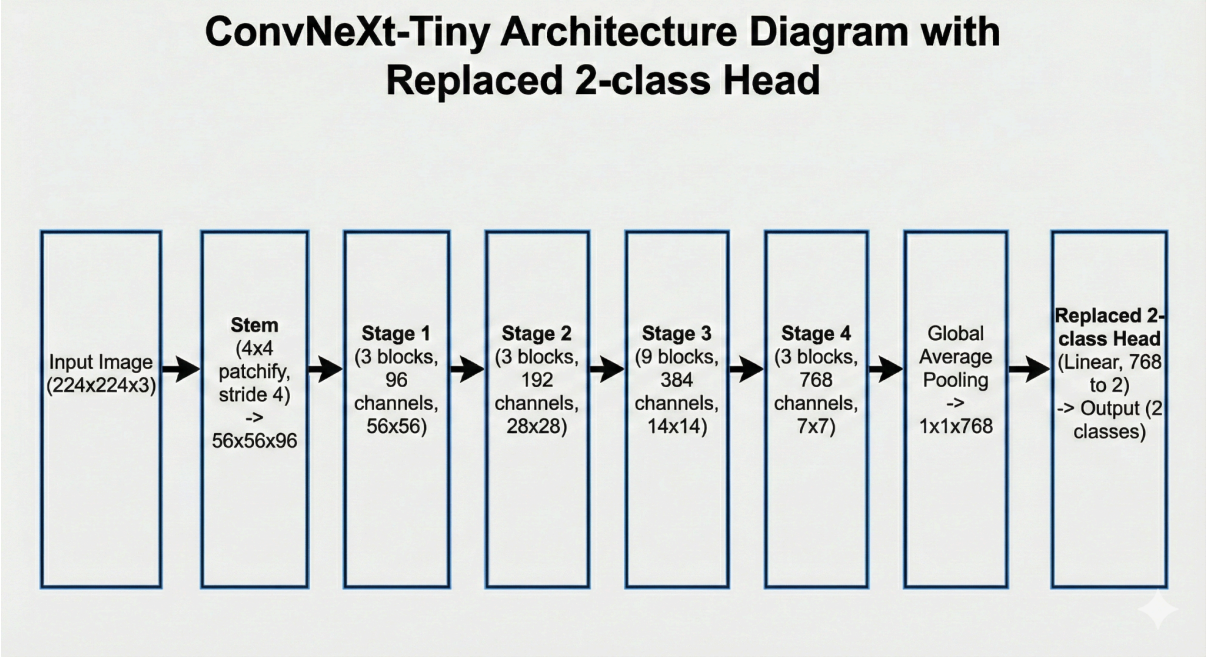
---

4. Model Architecture

4.1 ConvNeXt-Tiny

The classifier is based on **ConvNeXt-Tiny**, a modern convolutional neural network architecture that revisits classical CNN design principles while incorporating insights from Vision Transformers. ConvNeXt achieves competitive performance while maintaining architectural simplicity and computational efficiency.

The original classification head is replaced with a fully connected layer containing **two output neurons**, corresponding to the binary classification task.



4.2 Transfer Learning Strategy

To accelerate convergence and improve generalization, the network is initialized using **ImageNet-pretrained weights**. Transfer learning is particularly beneficial in this context, as it allows the model to reuse low-level and mid-level visual features learned from large-scale natural image datasets.

---

5. Training Strategy

5.1 Cloud-Based Training in Google Colab

The final training process was conducted in **Google Colab**, leveraging GPU acceleration to handle the computational demands of ConvNeXt training. The dataset was uploaded in compressed form and extracted directly within the Colab environment.

5.2 Additional Computational Resources

The standard free-tier Colab environment proved insufficient for training on a 70 GB dataset. As a result, **additional paid computational resources** were contracted, enabling extended GPU usage, improved stability, and reduced training time. This decision was essential to complete the experiments within reasonable time constraints.

5.3 Training Configuration

The training process uses the following configuration:

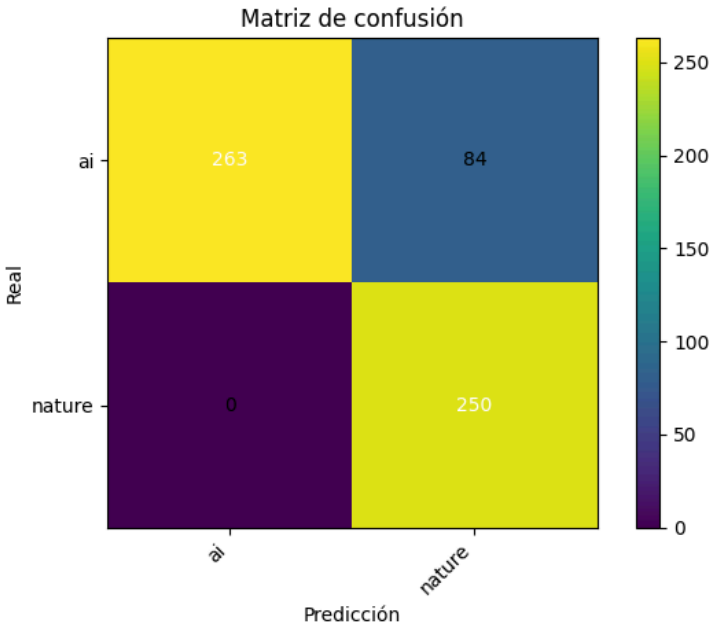
- Optimizer: Adam
- Loss function: Cross-Entropy Loss
- Learning rate: 1e-4
- Batch size: 16
- Number of epochs: 5–10

Model checkpoints are saved after each epoch to allow later evaluation and inference.

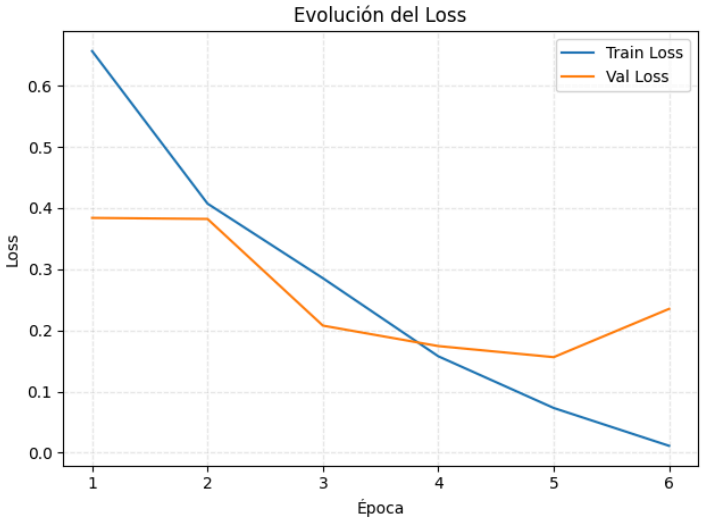
---

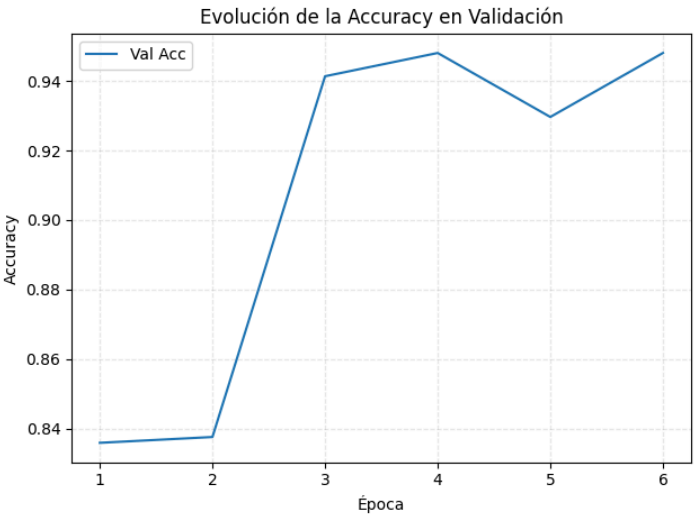
6. Evaluation Methodology

Model performance is assessed on the validation split using standard classification metrics, including accuracy, precision, recall, F1-score, and confusion matrices. These metrics provide a comprehensive understanding of classification behavior and potential class imbalance effects.



All reported metrics and plots are generated using the `samples/ALL` subset to ensure reproducibility and computational efficiency.

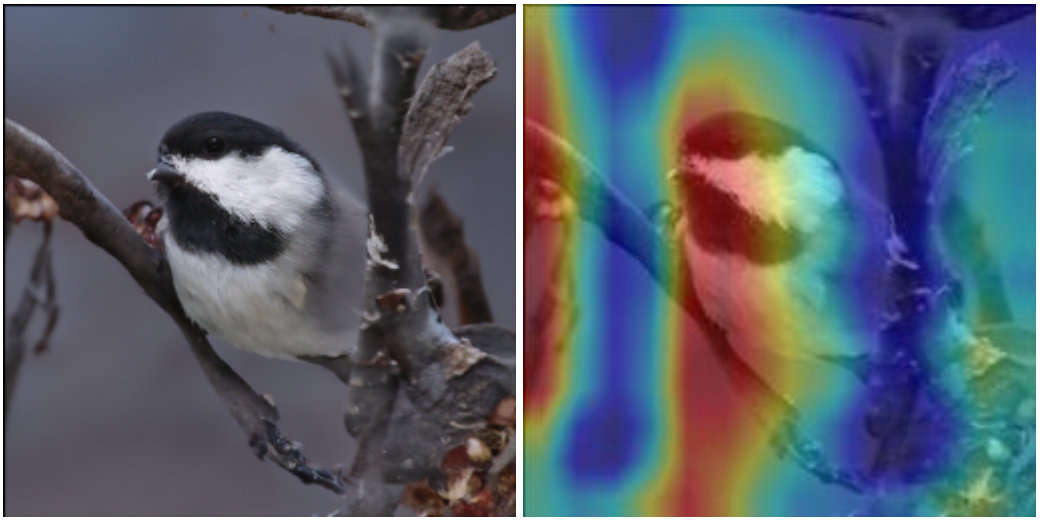




7. Explainability with Grad-CAM

To improve interpretability, **Grad-CAM (Gradient-weighted Class Activation Mapping)** is applied to the final convolutional stages of the ConvNeXt network. Grad-CAM highlights the spatial regions that contribute most strongly to the model's predictions.

AI-Generated Image Heatmap:



Real Image Heatmap:

This qualitative analysis helps verify that the model focuses on semantically meaningful regions rather than spurious artifacts, thereby increasing confidence in its predictions.

---

8. Project Organization

The project follows a clear and scalable directory structure:

- `data/raw/`: Full datasets per generator
- `data/samples/ALL/`: Reduced evaluation subset
- `models/`: Saved model checkpoints
- `results/figures/`: Training and evaluation plots
- `results/metrics/`: Numerical evaluation metrics
- `results/gradcams/`: Grad-CAM visualizations
- `src/`: Core implementation scripts

This organization facilitates reproducibility and future extensions.

---

9. Limitations and Future Work

The primary limitation of this work is the restriction to two generators during final training. While this choice was necessary due to hardware constraints, it limits generalization to unseen generators. Future work could incorporate additional generators, larger compute budgets, and cross-dataset evaluation protocols.

---

10. Conclusion

This project demonstrates that modern convolutional architectures such as ConvNeXt are highly effective for AI-generated image detection when combined with carefully curated datasets and appropriate training strategies. Despite significant hardware constraints, the use of cloud-based resources enabled the successful training and evaluation of a robust classifier.

The integration of Grad-CAM provides valuable insights into model behavior, contributing to transparency and interpretability. Overall, the proposed system constitutes a solid and extensible foundation for further research in AI-generated content detection.

References

[1] GenImage Dataset Contributors. *GenImage: A Large-scale Benchmark for AI-Generated Image Detection*. GitHub repository, 2023. Available at: https://github.com/GenImage-Dataset/GenImage

[2] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. *A ConvNet for the 2020s (ConvNeXt)*. Facebook AI Research, 2022. GitHub repository: https://github.com/facebookresearch/ConvNeXt

[3] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*. International Journal of Computer Vision (IJCV), 2020. PyTorch implementation available at: https://github.com/jacobgil/pytorch-grad-cam

[4] Paszke, A. et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. NeurIPS, 2019.

[5] Brock, A., Donahue, J., and Simonyan, K. *Large Scale GAN Training for High Fidelity Natural Image Synthesis (BigGAN)*. ICLR, 2019.

[6] Nichol, A. Q. et al. *GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models*. ICML, 2022.