Robust and Explainable Deepfake Detection: Integrating Vision Transformers, LLMs, and Domain Generalization

Ching-Heng Huang, Chung-Yeh Yang, Liang-Jie Chiu

December 7, 2025

Abstract

The rapid proliferation of generative AI tools, such as Stable Diffusion and StyleGAN, has democratized the creation of hyper-realistic synthetic faces. While these advancements foster creativity, they also pose significant risks regarding identity theft and misinformation. This project develops a robust deepfake detection pipeline utilizing three distinct deep learning architectures: ResNet34, EfficientNet-B4, and Vision Transformer (ViT-B/16). Beyond training classifiers, we engineered a user-facing Streamlit Proof-of-Concept (POC) tool that integrates Grad-CAM heatmaps with the Gemini Large Language Model (LLM) to provide natural language explanations for model decisions. While our models achieve substantial accuracy (approx. 96.5%) on standard benchmarks, we further conducted practical tests on modern GenAI outputs, achieving near-perfect detection on fully synthetic images and ∼90% accuracy on high-quality face swaps, demonstrating significant real-world utility despite challenges in domain generalization.

1 Introduction

1.1 Motivation

The digital landscape has been transformed by AI-generated content. Tools capable of face swapping and image synthesis have become accessible to the general public, leading to a surge in "deepfake" imagery. Beyond the novelty, these technologies facilitate illegal activities, including fraud and non-consensual pornography.

Current detection systems often function as opaque classifiers—outputting a probability score without reasoning. For the general public, a simple "99% Fake" label is insufficient to build trust or understanding. Therefore, our objective is twofold:

1. **Robust Detection:** Train effective models to distinguish between authentic and AI-generated faces.

2. **Explainability:** Bridge the gap between technical metrics and human understanding by visualizing decision regions (Heatmaps) and interpreting them via an LLM.

2 Related Work

2.1 Deep Convolutional Architectures

Early work in deepfake detection relied heavily on standard Convolutional Neural Networks (CNNs). The use of ResNet introduced residual connections to enable the training of deeper, more robust networks. More recently, efficiency-focused architectures like EfficientNet leveraged compound scaling to optimize performance across depth, width, and resolution.

2.2 Vision Transformers (ViT)

The shift from CNNs to transformer-based models represents a pivotal change. By treating image patches as sequences and relying on self-attention mechanisms, ViT is able to model global context effectively, which is critical for detecting non-local, structural artifacts introduced by modern generative models.

2.3 Explainable AI (XAI)

Addressing the opaque nature of deep learning classifiers, methods like Grad-CAM have become standard for visualizing which regions of an image most influence a model's prediction. For transformer-based models, techniques such as Attention Rollout are used. The integration of such visualization with LLMs (like Gemini) is a novel step toward translating technical heatmaps into actionable explanations.

2.4 Domain Generalization

A primary challenge is the lack of robustness against unseen generator architectures. Research such as Domain-Adversarial Training of Neural Networks (DANN) aims to address this by training feature extractors to be invariant to the source domain while maintaining high classification accuracy.

3 Methodology

3.1 Dataset Preparation

To ensure diversity, we curated a composite dataset from multiple high-quality sources:

- **Real Data:** FFHQ-real and CelebDF (Youtube-real).
- **Fake Data:** CelebDF, StyleGAN, and Stable Diffusion.
- **Holdout Set:** Sourced from the DF40 dataset, comprising 40 distinct deepfake techniques (including DeepFaceLab, HeyGen, and MidJourney) and authentic real images.

3.1.1 Data Processing

For video sources, we implemented a selective preprocessing pipeline:

1. **Uniform Sampling:** Sampled 20 candidate frames per video to ensure temporal diversity.
2. **Face Detection:** Utilized MediaPipe (confidence > 0.5) and selected the Top-3 highest-quality faces per video.
3. **Context-Aware Cropping:** Applied a square crop with 30% context padding to preserve critical boundary regions (hairline, ears, chin).

3.2 Model Architectures

We selected three architectures representing different paradigms:

- **ResNet34:** A standard CNN baseline.
- **EfficientNet-B4:** Optimized for efficiency/accuracy.
- **ViT-B/16:** Attention-based global context modeling.

Modifications to the base models are detailed in Table 1.

Table 1: Model Architecture Modifications

Model	Base	Modification
ResNet	`resnet34`	Final FC layer modified from 1000 to 2 classes.
EfficientNet	`efficientnet_b4`	Classifier head modified from 1000 to 2 classes.
ViT	`vit-base`	Final FC layer modified from 1000 to 2 classes.

3.3 Training Strategy

CNNs (ResNet/EfficientNet): Full fine-tuning was employed to adapt early convolutional layers to subtle pixel-level manipulations.

ViT: We implemented a two-stage progressive fine-tuning approach:

- *Stage 1:* Freeze embedding and first 10 encoder layers; train only the last 2 layers and head.
- *Stage 2:* Unfreeze all layers for full fine-tuning.

3.4 Explainability Pipeline (XAI + LLM)

We implemented a pipeline combining Grad-CAM/Attention Rollout with Google Gemini (Flash-2.5):

1. **Inference:** Model predicts Real/Fake.
2. **Visualization:** Compute heatmap (Grad-CAM for CNNs, Attention Rollout for ViT).
3. **Interpretation:** The heatmap and image are fed to Gemini to generate a natural language explanation (e.g., "Inconsistent lighting on the left ear").

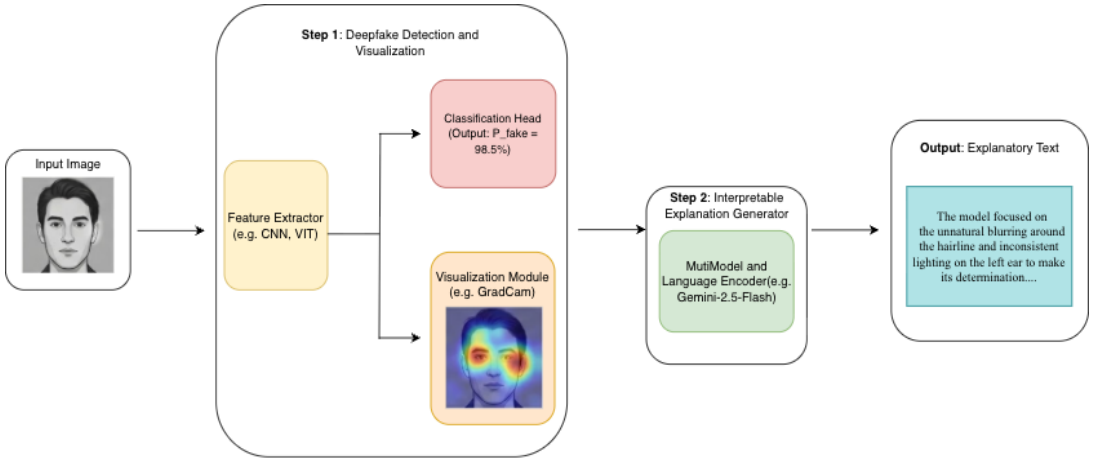


Figure 1: Overview of the Explainability Pipeline: From Input Image to Grad-CAM Heatmap to Gemini Explanation.

4 Experimental Results

4.1 Training Performance

The models showed stable convergence during training. As illustrated in the validation metrics below, the loss decreased consistently while accuracy improved.

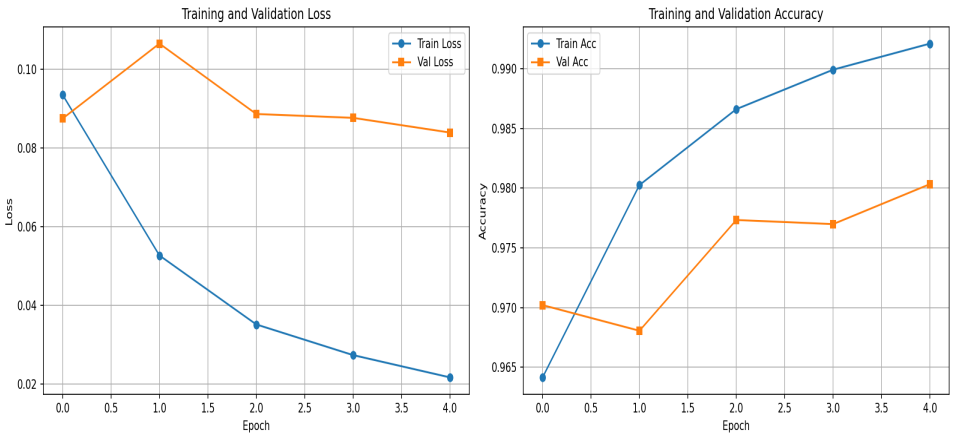


Figure 2: EfficientNet-B4 Training/Validation Loss and Accuracy Curves.

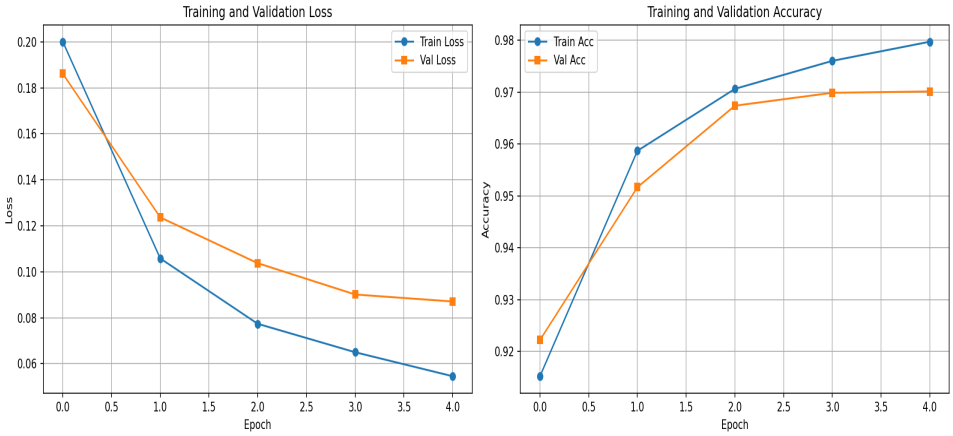


Figure 3: ResNet34 Training/Validation Loss and Accuracy Curves.

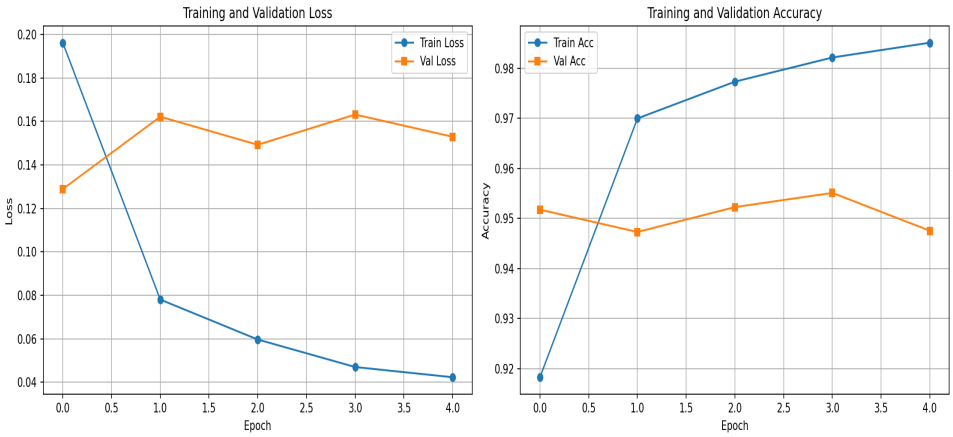


Figure 4: ViT-B/16 Training/Validation Loss and Accuracy Curves.

4.2 Standard Test Set Evaluation

The models were evaluated on a test split of 11,623 images containing similar domains to the training set.

Table 2: Standard Test Set Performance

Model	Accuracy	F1 (Fake)	AUC
ResNet34	94.42%	0.9616	0.9870
EfficientNet-B4	96.58%	0.9758	0.9923
ViT-B/16	96.45%	0.9748	0.9931

4.3 Practical "In-the-Wild" Evaluation

Using our Streamlit tool, we tested outputs from modern generators:

- **Fully Synthetic (Gemini/Perplexity):** Near 100% accuracy. Models easily detected structural inconsistencies.
- **Face Swaps:** ∼90% accuracy. Dependent on image resolution; blending boundaries were key features.

4.4 Detailed Per-Domain Evaluation

To understand the models' strengths and weaknesses, we analyzed performance across specific sub-domains within the test set.

Table 3: Detailed Accuracy by Sub-Domain

Domain (Count)	ResNet34	EffNet-B4	ViT-B/16
Real Faces			
Celeb-real (1716)	83.39%	92.31%	94.76%
YouTube-real (1028)	88.13%	95.14%	95.82%
FFHQ-real (675)	76.44%	93.78%	88.59%
Fake Faces			
Celeb-synthesis (6758)	99.27%	98.08%	97.19%
StableDiffusion (546)	98.90%	99.27%	99.63%
StyleGAN (900)	97.00%	95.56%	98.78%

Key Insight: ResNet34 struggles significantly with high-quality real faces (e.g., 76.44% on FFHQ), often misclassifying them as fake. In contrast, ViT offers the most balanced performance, maintaining high accuracy ($> 94\%$) on real-world footage while accurately detecting diffusion artifacts.

5 Discussion: The Generalization Challenge

While performance was high on in-distribution data, we observed a significant drop on our rigorous Holdout Set (unseen domains like Midjourney 6, HeyGen). ResNet34 accuracy dropped to 35.55%, and ViT to 55.39%, indicating domain overfitting.

5.1 Multi-Source Domain Generalization (MSDG)

To address this, we adopted a Multi-Source Domain Generalization approach using Domain-Adversarial Training (DANN). We added a Domain Classifier and a Gradient Reversal Layer (GRL) to force the feature extractor to learn domain-invariant representations.

$$L_{total} = L_{class}(y, \hat{y}) - \lambda L_{domain}(d, \hat{d}) \tag{1}$$

MSDG improved Holdout accuracy from 35.55% to 56.11%, showing promise, though a generalization gap remains due to architectural novelty in the holdout generators.

5.2 Critical Analysis: The Technological Temporal Shift

Although MSDG improved accuracy on the unseen Holdout set from 35.55% to 56.11%, a substantial generalization gap (approx. 40%) remains. We hypothesize that this limitation stems from a fundamental **"Technological Generation Gap"** between our training and holdout distributions.

- **Training Distribution (2019–2023 Era):** Our source domains represent a specific snapshot of deepfake technology, primarily relying on standard GANs (StyleGAN2) and early Diffusion models (Stable Diffusion v1.4/1.5).
- **Holdout Distribution (2024 Era):** The holdout set includes state-of-the-art generators such as *Midjourney v6*, *HeyGen*, and advanced implementations of *DALL-E 3*.

Why Domain Adaptation Failed to Extrapolate: Domain adversarial training effectively aligns feature distributions *within* the observed technological paradigms. However, the newest generators introduce novel architectures (e.g., Transformers integrated with Latent Diffusion) and possess significantly higher image fidelity that lacks the specific artifacts (like background warping or eyes asymmetry) present in the older training data. The model, therefore, struggles to extrapolate to these "future" domains that lie completely outside the manifold of the learned source domains.

6 Conclusion and Future Work

6.1 Conclusion

We developed an end-to-end Explainable AI deepfake detection system and investigated domain generalization strategies for cross-generator robustness. Our baseline models achieve nearly perfect detection rates (93-95%) on held-out test data from the training distribution. The integration of a Streamlit interface with LLM-based explanations represents a significant step towards user-centric AI safety tools. To address cross-generator generalization, we implemented Multi-Source Domain Generalization (MSDG) using domain adversarial training across three source domains (video-based, GAN-based, and diffusion-based deepfakes). MSDG achieved 96.92% accuracy on in-distribution test data and improved holdout performance by 20.56 percentage points over baseline (56.11% vs. 35.6%). However, substantial challenges remain: our holdout set—containing eight diverse generators, including recent methods (Midjourney 6, HeyGen) and novel techniques (StyleCLIP, CollabDiff)—revealed a 40.81% generalization gap, highlighting the difficulty of detecting generators that differ significantly from training data in architecture or quality.

6.2 Future Work

To address the generalization gap observed in the holdout set, future research will focus on:

1. **Expanded Training Coverage**: Include recent generators (Midjourney 6, HeyGen, DALL-E 3) and novel techniques (text-guided manipulation, collaborative generation) in training to reduce the architecture gap.
2. **Foundation Model Features**: Replace CNN backbones with pre-trained vision-language models (CLIP, DINOv2) that may capture more universal visual patterns beyond generator-specific artifacts.
3. **Multi-Modal Detection**: Combine spatial features with frequency-domain analysis (FFT/DCT) to detect structural anomalies that may be more robust across generator evolution.
4. **Continuous Learning Pipeline**: Establish automated retraining workflows that incorporate new generators as they emerge, with monitoring for distribution drift in deployment.
5. **Hybrid Approaches**: Explore complementary strategies combining learned features with physics-based forensics (lighting consistency, physiological plausibility) and content provenance systems (C2PA standards).

7 Team Contributions

The specific contributions of each team member to this project are outlined below:

Table 4: Project Contributions and Responsibilities

Member	**Primary Role**	**Key Tasks**
Ching-Heng Huang	Data processing & Model Training & Streamlit POC & Slide	Curated the dataset (FFHQ, CelebDF); Implemented the data preprocessing pipeline; Trained and Finetuned EfficientNet-B4 baselines; Constructed Streamlit base POC; Created project proposal and slide.
Chung-Yeh Yang	System Implementation & Explainability (XAI)	Trained and Finetuned ResNet34 baselines; Designed the XAI(Grad-cam) Architecture; Converted the notebook into structured and unified pipeline; Revised and finalized the Streamlit Demo.
Liang-Jie Chiu	Domain Generalization & Model Training & Evaluation	Trained the ViT baselines; Implemented the Attention Rollout; Designed and trained the MSDG (Domain Adversarial) model; Conducted the Holdout set analysis; Wrote the Discussion and Conclusion and Future Work sections.

References

[1] Y. Li et al., "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics," CVPR, 2020.

[2] T. Karras et al., "A Style-Based Generator Architecture for Generative Adversarial Networks," CVPR, 2019.

[3] K. He et al., "Deep Residual Learning for Image Recognition," CVPR, 2016.

[4] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," ICML, 2019.

[5] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," ICLR, 2021.

[6] R. R. Selvaraju et al., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," ICCV, 2017.

[7] Y. Ganin et al., "Domain-Adversarial Training of Neural Networks," JMLR, 2016.

[8] Z. Yan et al., "DF40: Toward Next-Generation Deepfake Detection," arXiv, 2024.

[9] S. Abnar et al., "Quantifying Attention Flow in Transformers," ACL, 2020.

A Appendix

A.1 Training Hyperparameters

Table 5 details the specific hyperparameters used for training the baseline models.

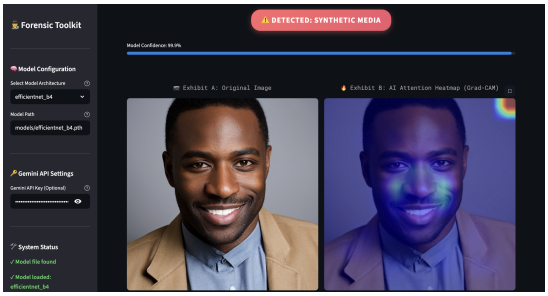
Table 5: Training Hyperparameters

Hyperparameter	**ResNet/EffNet**	**ViT**
Epochs	5	5
Batch Size	32	16
Learning Rate	1e-4	2e-5
Weight Decay	0.01	0.01
Optimizer	AdamW	AdamW
Warmup Ratio	0.1	0.1
Patience	3	3

A.2 Streamlit POC Implementation

The Proof-of-Concept tool was built using `Streamlit`. It allows users to upload an image and select a model backend. The explanation generation utilizes the `Google GenAI SDK` to prompt Gemini-Flash-2.5 with the generated heatmap and the original image, requesting a layman-friendly analysis of the artifacts.

(a) Step 1: Uploading image



(b) Step 2: Real-time Heatmap

(c) Step 3: Generated explanation by Gemini

Figure 5: Streamlit POC System Workflow: From input selection to final explanation.

A.3 MSDG Implementation

Table 6 details the specific configurations and hyperparameters used for training MSDG.

Table 6: MSDG Training and Implementation Parameters

Hyperparameter	**Value**
Feature Extractor	ResNet34 (ImageNet Pretrained)
Epochs	30
Batch Size	32
Optimizer	Adam
Learning Rate	1e-4
Domain Labels	
Domain 0	Celeb-DF + YouTube-Real (video-based)
Domain 1	FFHQ-Real + StyleGAN (GAN-based)
Domain 2	FFHQ-Real (reused) + Stable Diffusion (diffusion-based)