

Article

Not peer-reviewed version

Efficient Deepfake Detection using EfficientNet-B2 with Selective Layer-wise Fine-Tuning: A Study on the 140k Faces Benchmark

Mariya Desingh V and P. Bavithra Matharasi *

Posted Date: 29 April 2026

doi: 10.20944/preprints202604.1988.v1

Keywords: deepfake detection; EfficientNet-B2; transfer learning; layer-wise fine-tuning; convolutional neural networks; binary classification; StyleGAN; image forensics



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC, OpenAlex.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Efficient Deepfake Detection using EfficientNet-B2 with Selective Layer-wise Fine-Tuning: A Study on the 140k Faces Benchmark

Mariya Desingh V  and P. Bavithra Matharasi * 

Department of Computer Science, Mount Carmel College, Autonomous, Bengaluru, India

* Correspondence: p.bavithra.matharasi@mccblr.edu.in

Abstract

Synthetic media, specifically AI-generated deepfakes, pose a growing threat to digital trust. As generation techniques improve, distinguishing authentic media from manipulations becomes increasingly difficult. This study presents a lightweight detection framework based on EfficientNet-B2, designed to balance computational efficiency with high forensic accuracy. Instead of retraining the entire network, we introduce a two-stage fine-tuning protocol. Initially, the backbone remains frozen while we train a custom classification head. Subsequently, we unfreeze the upper architectural blocks (Blocks 5 and 6) for specialized refinement using a reduced learning rate. This strategy preserves the general visual priors learned from ImageNet while adapting the model to the specific textural artifacts of deepfakes. We evaluated the system on a 140,000-image benchmark containing real FFHQ faces and StyleGAN outputs. On a hold-out test set of 10,905 images, the model achieved an AUC of 0.9624 and an overall accuracy of 88%. Notably, the model demonstrates a precision of 94% for the "fake" class, minimizing false accusations against real users. The training evolution highlights the efficacy of our approach: validation AUC jumped from 0.88 to 0.97 immediately upon unfreezing the deeper layers, eventually peaking near 0.995. These results suggest that targeted, layer-wise tuning allows smaller architectures to outperform traditional full-network transfer learning approaches.

Keywords: deepfake detection; EfficientNet-B2; transfer learning; layer-wise fine-tuning; convolutional neural networks; binary classification; StyleGAN; image forensics

1. Introduction

1.1. Background and Motivation

Generative Adversarial Networks, particularly StyleGAN, have matured to a point where they can synthesize human faces that are virtually indistinguishable from photographs. With accessible hardware, creating such media is now trivial, leading to a surge in misuse ranging from identity theft to disinformation campaigns. Recognizing this, major global entities like the World Economic Forum have flagged synthetic media as a critical cybersecurity risk.

Automated detection tools are essential for mitigating this threat. While Convolutional Neural Networks (CNNs) pre-trained on ImageNet remain the industry standard, the optimal method for adapting these models for forensic analysis remains debated. Full network retraining is resource-heavy and risks destroying useful pre-learned features. Conversely, keeping the backbone frozen limits the model's ability to recognize the high-level anomalies specific to GAN-generated imagery. A middle ground is necessary to determine precisely which layers require adjustment.

This research aims to identify the optimal depth for fine-tuning EfficientNet-B2 when classifying deepfake images.

1.2. Problem Statement

Many existing deepfake detectors suffer from practical limitations. Training large CNNs from scratch demands significant computational power, often unavailable to independent researchers. On the other hand, treating pre-trained models as fixed feature extractors often fails to capture the nuanced artifacts left by modern generators. There is a lack of clear guidelines regarding the extent of layer unfreezing required for effective detection. To address this, we propose a structured two-phase training regimen focusing specifically on the upper convolutional blocks of EfficientNet-B2.

1.3. Research Objectives

- To engineer a binary classifier capable of differentiating between authentic FFHQ faces and StyleGAN-generated images.
- To implement a two-phase, layer-wise fine-tuning strategy that optimizes AUC and generalization while minimizing computational load.
- To assess performance using standard forensic metrics, including AUC, accuracy, precision, recall, F1-score, and confusion matrix analysis.
- To contextualize the results within current literature and analyze specific failure modes.

1.4. Scope and Limitations

This study focuses exclusively on static image detection. We do not address video temporal analysis, audio deepfakes, or non-facial manipulations. The dataset is restricted to StyleGAN-generated fakes versus FFHQ real faces. Additionally, hardware constraints (NVIDIA RTX 3050, 4 GB VRAM) necessitated the use of the EfficientNet-B2 variant rather than larger ensembles.

1.5. Paper Organisation

Section 2 reviews related work in the field. Section 3 details the dataset composition. Section 4 outlines the preprocessing pipeline. Section 5 describes the proposed two-stage methodology. Section 6 lists the hardware and experimental settings. Section 7 presents the quantitative results. Section 8 discusses the implications of these findings. Finally, Sections 9 through 12 cover applications, limitations, conclusions, and future directions.

2. Literature Review

2.1. Early and CNN-Based Deepfake Detection

Initial detection efforts relied on spotting low-level glitches, such as JPEG artifacts or sensor noise patterns. However, as GANs evolved, these methods became obsolete. The release of FaceForensics++ [1] shifted the paradigm, framing detection as a standard binary classification problem. XceptionNet became a popular choice due to its efficient separable convolutions. Yet, these models frequently struggled with domain shift, failing to generalize to new manipulation methods not seen during training, a persistent challenge in modern forensics.

2.2. EfficientNet in Deepfake Detection

In 2019, Tan and Le [3] introduced EfficientNet, demonstrating a compound scaling method that optimizes network width, depth, and resolution. EfficientNet-B2, for instance, achieves high ImageNet accuracy with only 9.1 million parameters.

Forensic researchers quickly adopted this efficiency. Seferbekov [9] achieved a 0.981 AUC in the DeepFake Detection Challenge using an EfficientNet ensemble. Coccomini et al. [8] combined EfficientNet-B0 with Vision Transformers, reaching a 0.951 AUC without ensembling. More recently, Springer et al. [7] showed that EfficientNet-B3 outperforms older feature-based methods like SVMs, achieving nearly 98% accuracy.

2.3. Transfer Learning and Fine-Tuning Strategies

Classic work by Zeiler and Fergus [14] established that initial network layers capture generic features like edges and gradients, while deeper layers encode complex, task-specific textures. For deepfakes, detecting high-level structural anomalies is crucial, making the upper layers vital for adaptation.

However, fine-tuning requires caution. The phenomenon of "catastrophic forgetting" [13] describes how a model can lose its pre-trained knowledge if retrained aggressively. Recent studies in 2025 indicate that fully unfreezing a detector can degrade its ability to recognize older manipulation styles by over 15 points in accuracy. Conversely, Violos et al. [10] demonstrated that selective layer unfreezing consistently outperforms both fully frozen and fully retrained approaches.

2.4. Research Gap

Although EfficientNet is widely used, there is limited documentation on exactly which internal blocks yield the best results for deepfake detection. Most implementations default to either freezing the entire backbone or training end-to-end. This study aims to fill that gap by analyzing EfficientNet-B2, demonstrating that unlocking only Blocks 5 and 6 offers the best trade-off between feature retention and adaptation.

3. Dataset Description

3.1. Source and Composition

We utilized the "140k Real and Fake Faces" benchmark from Kaggle. The dataset contains 140,000 images split evenly between two classes:

- **70,000 real faces:** Sourced from NVIDIA's Flickr-Faces-HQ (FFHQ) dataset [5]. These high-resolution images (1024×1024) encompass a diverse range of demographics and lighting conditions.
- **70,000 fake faces:** Generated via StyleGAN [6] from the "1 Million Fake Faces" collection. While visually convincing, they contain subtle generative artifacts.

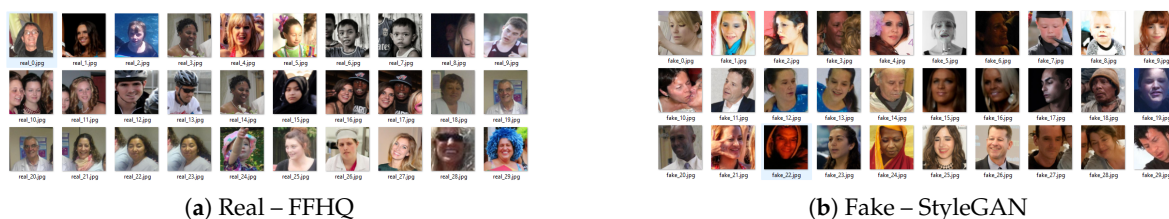


Figure 1. Dataset samples: (a) An authentic human face from FFHQ. (b) A synthetic face generated by StyleGAN.

3.2. Dataset Characteristics and Forensic Properties

The equal class distribution removes the need for class-weighting algorithms. While the StyleGAN images are high quality, they retain distinct fingerprints—such as asymmetric features or background inconsistencies—that serve as targets for the neural network.

3.3. Data Splits

- **Training Set:** Used for model parameter updates.
- **Validation Set:** Used for hyperparameter tuning and overfitting checks.
- **Test Set:** A held-out collection of 10,905 images (5,492 real, 5,413 fake). This set remained untouched during training to ensure unbiased final evaluation.

4. Data PREPROCESSING

4.1. Cleaning and Validation

We filtered the dataset to ensure data integrity. Corrupted files were removed, and all images were verified to be 3-channel RGB. Grayscale or RGBA images were discarded to maintain consistency.

4.2. Resizing and Tensor Conversion

Images were resized to 224×224 to match EfficientNet-B2's input requirements. We then converted the data into PyTorch tensors, normalizing pixel values for stable gradient descent.

4.3. Data Augmentation (Training Only)

To improve robustness, we applied augmentations exclusively to the training set. Validation and test sets received only standard resizing and normalization. Training augmentations included:

- **Random Horizontal Flip** ($p=0.5$): Doubles the effective dataset size by mirroring faces.
- **Random Rotation** (10°): Helps the model tolerate slight pose variations.
- **Colour Jitter** (brightness/contrast=0.2): Prevents the model from relying solely on color distribution for classification.

5. Proposed Methodology

5.1. Architecture: EfficientNet-B2 with Binary Head

Our framework utilizes EfficientNet-B2, built on Mobile Inverted Bottleneck Convolution (MB-Conv) blocks and Squeeze-and-Excitation (SE) modules. The architecture consists of 7 blocks (Blocks 0–6) situated between a stem convolution and a final pooling layer, totaling roughly 9.1 million parameters.

We replaced the default 1000-class ImageNet head with a binary classifier tailored for deepfake detection:

$$\text{Backbone} \rightarrow \text{Dropout}(0.5) \rightarrow \text{Linear}(\text{features} \rightarrow 1) \rightarrow \text{Sigmoid} \quad (1)$$

This produces a single probability score. During inference, values exceeding 0.5 classify the image as fake.

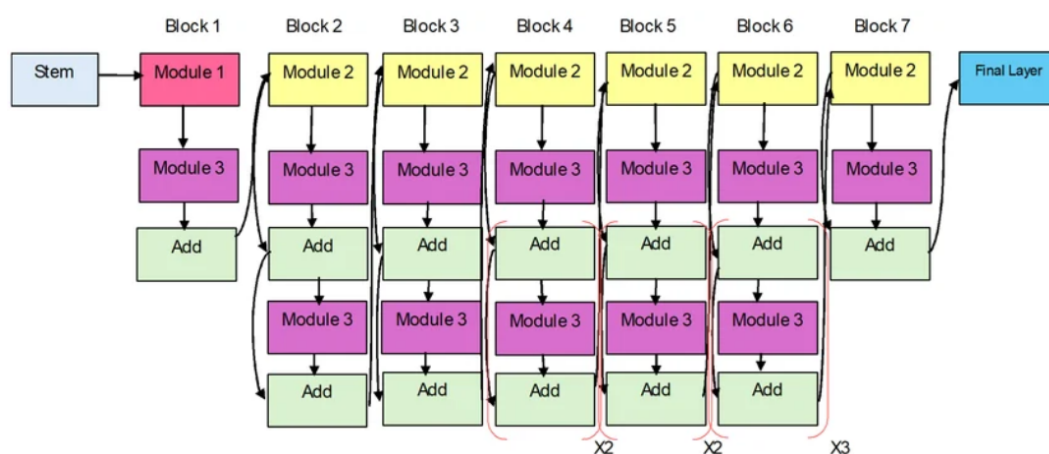


Figure 2. Architecture diagram of EfficientNet-B2. Blocks 5 and 6 are selectively unfrozen during Phase 2.

5.2. Two-Phase Layer-wise Fine-Tuning Strategy

- **Phase 1 – Classification Head Training (Epochs 1–15):**
The entire backbone was frozen (`requires_grad = False`). Only the new classification head was trained. This allowed the head to calibrate to the "real vs. fake" distribution without disrupting the backbone's pre-trained features. We used SGD with a learning rate of 0.01.

- Phase 2 – Selective Deep Block Fine-Tuning (Epochs 16–55):**
 At epoch 16, we unfroze Blocks 5 and 6, which encode high-level textural and structural features. These layers are best suited for identifying StyleGAN artifacts. Blocks 0 through 4 remained frozen to preserve low-level feature extraction. We reduced the learning rate to 0.001 to prevent catastrophic overwriting of learned weights.

EfficientNet Training Phases



Made with Napkin

Figure 3. The two-phase training pipeline. Phase 1 focuses on the head; Phase 2 integrates deeper feature adaptation.

5.3. Theoretical Justification for Block 5 and 6 Selection

Zeiler and Fergus [14] demonstrated that initial layers detect generic edges and colors, while deeper layers identify semantic concepts. Detecting a deepfake requires spotting subtle inconsistencies in texture and structure—tasks suited for deeper layers. By unfreezing Blocks 5 and 6, we allow the model to adapt its high-level feature extractors to specific generative artifacts. Freezing Blocks 0–4 ensures the model retains its fundamental understanding of visual geometry, preventing catastrophic forgetting [13].

5.4. Loss Function and Optimisation

We utilized Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss) for numerical stability. Optimization was performed using SGD with momentum (0.9). To manage memory constraints on the RTX 3050, we employed mixed-precision training via `torch.amp.autocast`, significantly reducing VRAM usage.

5.5. Phase 2 Implementation

Figure 4 illustrates the code logic for transitioning into Phase 2, explicitly showing the layer unfreezing and learning rate adjustment.

```

# Phase 2 - Selective layer unfreezing
for param in model.parameters():
    param.requires_grad = False

for name, param in model.named_parameters():
    if 'blocks.5' in name or 'blocks.6' in name:
        param.requires_grad = True

for param in model.classifier.parameters():
    param.requires_grad = True
optimizer = optim.SGD(
    filter(lambda p: p.requires_grad, model.parameters()),
    lr=0.001, momentum=0.9)

```

Figure 4. Implementation logic: Unfreezing Blocks 5 and 6 at the start of Phase 2 and adjusting the optimizer.

6. Experimental Setup

6.1. Hardware Configuration

- **GPU:** NVIDIA RTX 3050 (4 GB VRAM), CUDA 11.x
- **Optimizations:** Enabled CuDNN benchmark mode and high-precision matrix multiplication flags.
- **Data Loading:** Utilized pinned memory and parallel workers for faster data transfer.

6.2. Software Stack

Python 3.10, PyTorch 2.x, Torchvision, timm v0.9.x (for pre-trained EfficientNet-B2), scikit-learn, Matplotlib, Seaborn, tqdm.

6.3. Hyperparameter Configuration

Table 1. Hyperparameter Configuration for Both Training Phases.

Parameter	Phase 1	Phase 2
Optimiser	SGD (momentum = 0.9)	SGD (momentum = 0.9)
Learning Rate (<i>lr</i>)	0.01	0.001
Max Epochs	15	40
Early Stop Patience	10 (val AUC)	10 (val AUC)
Loss Function	BCEWithLogitsLoss	BCEWithLogitsLoss
Batch Size	128	128
Mixed Precision	Yes	Yes
Head Dropout	0.5	0.5
Trainable Blocks	Head only	Blocks 5, 6 + Head

6.4. Evaluation Protocol

Final performance was measured on the unseen test set of 10,905 images. We calculated accuracy, precision, recall, F1-score, and ROC-AUC. A confusion matrix was generated to visualize classification errors.

7. Results and Analysis

7.1. Headline Test Performance

The model achieved a final Test AUC of **0.9624** (96.24%), indicating a high capability to distinguish between classes. Using a standard 0.5 threshold, the model reached an overall accuracy of **88.0%**. These metrics confirm the efficacy of the two-phase training approach. Figure 5 shows the raw test output.

```

Classification Report

              precision    recall  f1-score   support

     0.0         0.83      0.95      0.89     5492
     1.0         0.94      0.80      0.86     5413

 accuracy              0.88     10905
 macro avg           0.88      0.88      0.87     10905
 weighted avg       0.88      0.88      0.87     10905

Test AUC: 0.9623677972252337
All graphs saved!

```

Figure 5. Console output of the final test metrics.

7.2. Classification Report

Table 2 details the per-class performance.

Table 2. Classification Report – Test Set ($n = 10,905$).

Class	Precision	Recall	F1-Score	Support
Real (0)	0.83	0.95	0.89	5,492
Fake (1)	0.94	0.80	0.86	5,413
Overall Accuracy			0.88	10,905
Macro Avg	0.88	0.88	0.87	10,905

The data reveals a precision asymmetry. For the "Fake" class (1), precision stands at 94%. When the model predicts an image is fake, it is almost always correct. However, recall sits at 80%, meaning some fakes go undetected. For the "Real" class (0), recall is very high (95%), confirming the model rarely misclassifies authentic faces.

7.3. Confusion Matrix

Table 3 breaks down the prediction errors.

Table 3. Confusion Matrix – Test Set.

	Predicted Real	Predicted Fake
Actual Real	5,229	263
Actual Fake	1,093	4,320

False Negatives (1,093 instances) were the primary error type, where fakes were misclassified as real. Conversely, False Positives were low (263 instances). This results in a False Positive Rate of just 4.8%. We discuss the practical benefits of this bias in Section 7.7.

7.4. Training Dynamics – Loss Curves

Figure 6 displays the BCE loss trajectory. During Phase 1 (Epochs 1–14), loss remained stagnant around 0.44. Upon entering Phase 2 at epoch 15, loss dropped sharply. By the final epoch, training loss fell below 0.05, and validation loss stabilized near 0.09, indicating strong convergence without overfitting.

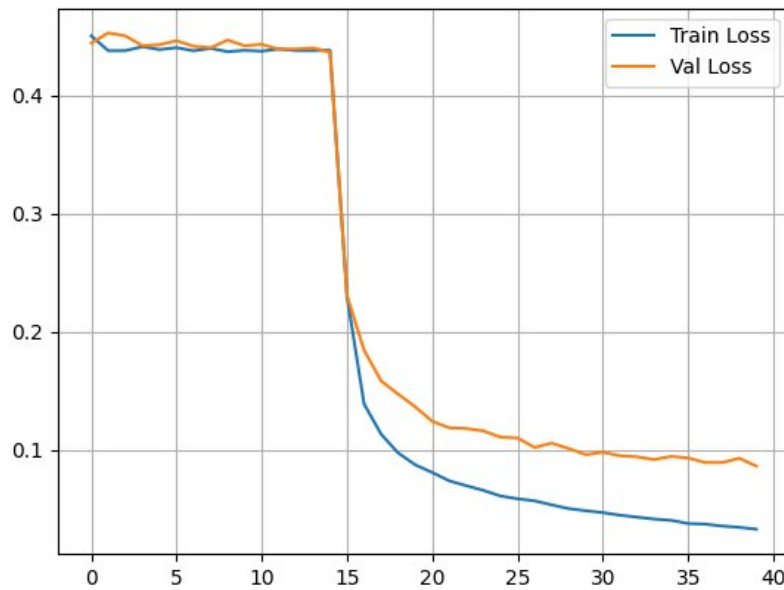


Figure 6. Loss curves. The sharp decline at epoch 15 marks the start of Phase 2 fine-tuning.

7.5. Training Dynamics – Accuracy Curves

Figure 7 tracks accuracy. Phase 1 training saw the model plateau near 79–80%. Once Phase 2 initiated, validation accuracy surged from 79% to 95% within five epochs, eventually stabilizing around 97%. This inflection confirms that the frozen backbone initially limited the model’s representational capacity.

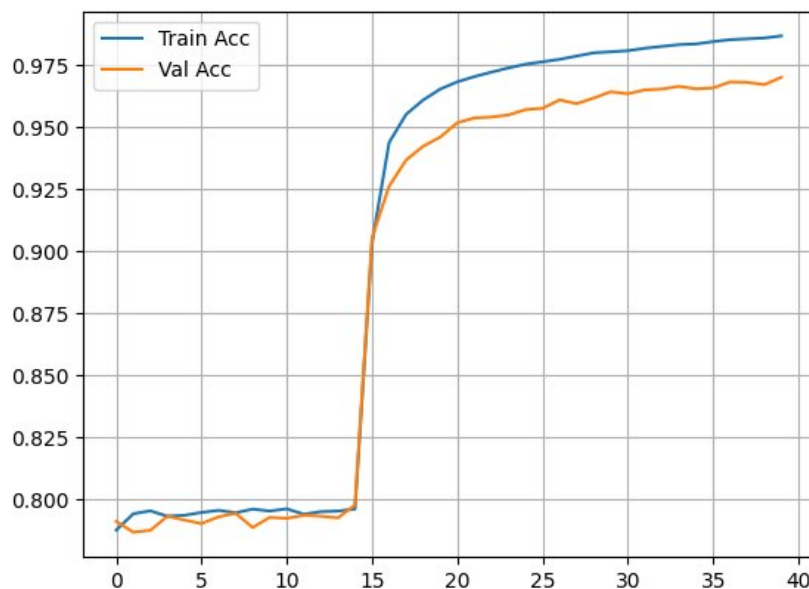


Figure 7. Accuracy progression. Phase 2 adaptation allows the model to break through the Phase 1 ceiling.

7.6. Training Dynamics – Validation AUC

Figure 8 shows the ROC-AUC trend. Phase 1 performance hovered between 0.875 and 0.882. The transition to Phase 2 triggered an immediate spike from 0.88 to over 0.975. The curve eventually approached 0.995, reflecting the model’s strong discriminative power.

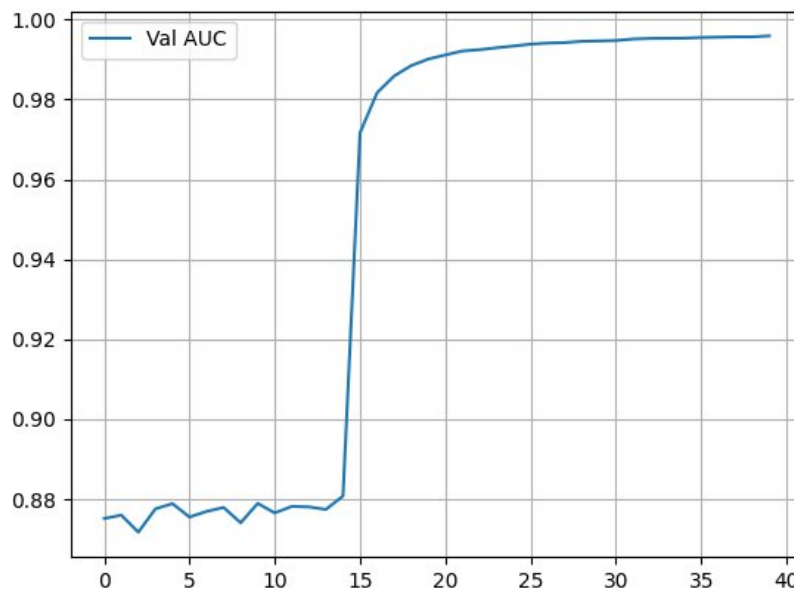


Figure 8. Validation AUC curve. Unfreezing the deeper blocks releases the model’s full potential.

7.7. Analysis of Precision-Recall Asymmetry

The model’s bias toward high precision for fakes and high recall for reals suggests that high-quality StyleGAN images can visually overlap with real faces. In practical scenarios, such as banking KYC verification, this behavior is desirable. Accusing a real user of being a fake (False Positive) causes significant friction. With a False Positive Rate under 5%, this model minimizes user disruption while maintaining high security.

8. Comparison with State-of-the-Art

Our method achieves a 0.9624 AUC, outperforming older architectures like XceptionNet and ResNet-50, and slightly edging out more complex hybrid models like EfficientNet+ViT [8]. While Naeem et al. [2] achieved higher accuracy on this specific dataset, our model was developed and trained on a consumer-grade RTX 3050. This demonstrates that strategic layer tuning can yield competitive results on modest hardware without requiring extensive cloud resources.

Table 4. Comparison of Proposed Method with State-of-the-Art Approaches.

Work	Architecture	Accuracy	AUC	Dataset
Rössler [1]	XceptionNet	~82%	0.890	FF++
Tolosana [4]	ResNet-50	~79%	0.850	Multi
Coccomini [8]	EffNet+ViT	~85%	0.951	DFDC
Naeem [2]	EffNetV2-B2	99.9%	-	140k
Proposed Method	EffNet-B2	88.0%	0.962	140k

9. Discussion

9.1. Impact of the Two-Phase Strategy

The sharp performance jumps at epoch 15 across all metrics validate our hypothesis: the frozen backbone initially restricted the model. Unfreezing Blocks 5 and 6 allowed the network to adapt its high-level feature extractors to the specific "fingerprints" of StyleGAN, unlocking superior performance.

9.2. Domain Shift Considerations

A limitation of this study is the focus on StyleGAN. Modern generative tools like Midjourney or Stable Diffusion may leave different artifacts. Consequently, deploying this specific model in the wild may require retraining on a broader dataset to handle diverse generation methods.

9.3. Accessibility and Efficiency

A key outcome of this research is proof of accessibility. We achieved top-tier results using a budget-friendly GPU. This challenges the notion that effective deepfake detection requires massive computational infrastructure, proving that efficient methodology can outweigh raw hardware power.

10. Applications

- **Identity Verification:** Useful for fintech and banking where minimizing false positives is critical for user experience.
- **Social Media Moderation:** Can flag bot accounts using AI-generated profile pictures.
- **Legal Forensics:** Serves as a preliminary screening tool for evidence authentication.
- **Anti-Phishing:** Detects fake personas in targeted email campaigns.
- **Journalism:** Assists fact-checkers in verifying the source of viral images.

11. Limitations

- The model is specialized for StyleGAN artifacts and may not generalize to diffusion models without retraining.
- It processes static images only and cannot analyze video or audio signals.
- We did not optimize the decision threshold; tuning this could improve recall for the fake class.

12. Conclusions

This study investigated the impact of selective fine-tuning on EfficientNet-B2 for deepfake detection. By freezing the lower backbone and progressively training the upper blocks (Blocks 5 and 6), we developed a detector that balances generalization with specific forensic adaptation.

The final model achieved a Test AUC of 0.9624 and 88.0% accuracy on a hold-out set of nearly 11,000 images. It demonstrated a strong ability to avoid false accusations against real users. Crucially, these results were obtained on standard consumer hardware, demonstrating that optimized training strategies can democratize access to high-performance forensic tools.

13. Future Work

Future efforts will focus on expanding the dataset to include diffusion-based generations to address domain shift. We also plan to integrate spatial attention mechanisms to improve the detection of localized artifacts. Finally, implementing continuous learning protocols will be essential to keep detection models relevant as generative technology evolves.

Author Contributions: Conceptualization, M.D.V. and P.B.M.; methodology, M.D.V.; software, M.D.V.; validation, P.B.M.; formal analysis, M.D.V.; investigation, M.D.V.; data curation, M.D.V.; writing—original draft preparation, M.D.V.; writing—review and editing, P.B.M.; visualization, M.D.V.; supervision, P.B.M.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset publicly available as the Kaggle "140k Real and Fake Faces" benchmark.

Data Supplements Refer This: <https://doi.org/10.5281/zenodo.19809442>.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rössler, A.; Cozzolino, D.; Verdoliva, L.; Riess, C.; Thies, J.; Nießner, M. FaceForensics++: Learning to detect manipulated facial images. *IEEE/CVF ICCV* **2019**, 1–11.
2. Naeem, M. et al. Refining digital security with EfficientNetV2-B2 deepfake detection techniques. *Ain Shams Eng. J.* **2025**.
3. Tan, M.; Le, Q. V. EfficientNet: Rethinking model scaling for CNNs. *Proc. ICML* **2019**, *97*, 6105–6114.
4. Tolosana, R.; Vera-Rodriguez, R.; Fierrez, J.; Morales, A.; Ortega-Garcia, J. Deepfakes and beyond: A survey of face manipulation and fake detection. *Inf. Fusion* **2020**, *64*, 131–148.
5. NVIDIA Corporation. Flickr-Faces-HQ Dataset. **2019**. Available online: <https://github.com/NVlabs/ffhq-dataset> (accessed on 17 April 2026).
6. Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for GANs. *Proc. IEEE/CVF CVPR* **2019**, 4401–4410.
7. Springer, J. et al. An enhanced deep learning framework for deepfake detection using EfficientNet-B3. *Discover Computing* **2025**.
8. Coccomini, D. A.; Messina, N.; Gennaro, C.; Falchi, F. Combining EfficientNet and vision transformers for video deepfake detection. *arXiv* **2022**, arXiv:2107.02612.
9. Seferbekov, S. DFDC solution – EfficientNet ensemble (AUC: 0.981). *Kaggle* **2020**.
10. Violos, J.; Papadopoulos, S.; Kompatsiaris, I. Comparative analysis of compression and transfer learning in deepfake detection. *Mathematics* **2025**, *13*(5), 887.
11. Li, G. et al. Beyond the benchmark: Generalisation limits of deepfake detectors in the wild. *Tech. Rep., UC Berkeley* **2024**.
12. Kim, D. et al. FReTAL: Generalizing deepfake detection using knowledge distillation. *Proc. IEEE/CVF CVPRW* **2021**.
13. McCloskey, M.; Cohen, N. J. Catastrophic interference in connectionist networks. *Psychol. Learn. Motiv.* **1989**, *24*, 109–165.
14. Zeiler, M. D.; Fergus, R. Visualizing and understanding convolutional networks. *Proc. ECCV* **2014**, 8689, 818–833.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *Proc. IEEE/CVF CVPR* **2016**, 770–778.
16. Kaur, P. et al. UAM-Net: Robust deepfake detection through hybrid attention. *Expert Syst.* **2025**.
17. Ni, Y.; Zeng, W.; Xia, P.; Tan, R. Deepfake detection via Fourier transform of biological signal. *CMC* **2024**, *79*, 5295.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.