Evaluating Deep Architectures for AI-Generated Face Detection

CS 747: Deep Learning

Spring 2025

Karan Monga
George Mason University
kmonga@gmu.edu

Akash Deshpande
George Mason University
adeshpa2@gmu.edu

Abstract

The rise of AI-generated images has made it increasingly difficult to distinguish synthetic faces from real ones. In this project, we evaluate how different deep neural network architectures—EfficientNet-B0 and MobileNetV2—perform on the task of binary face authenticity classification. Beyond accuracy, we use Grad-CAM to interpret each model's attention and t-SNE to visualize their learned feature spaces. Our findings show that EfficientNet-B0 achieves higher accuracy and more consistent feature separation than MobileNetV2, likely due to its compound scaling strategy. These results suggest that architecture design plays a critical role in detecting subtle visual artifacts in fake face generation.

1. Introduction

The rise of AI-generated faces through GANs and other synthesis techniques has made it more difficult to distinguish real images from AI-generated ones. These realistic fake faces pose significant threats in everyday aspects such as media and misinformation. As a result, detecting whether a face is real or AI-generated has become an important and active area of research.

While many approaches focus on classification accuracy, we aim to better understand how different model architectures behave in the context of fake face detection. In this project, we compare two deep CNNs, EfficientNet-80 and MobileNetV2, on their ability to detect AI-generated faces. Our comparison goes beyond performance metrics. We analyze how each model learns, where it focuses during classification, and how it organizes real and fake examples in feature space.

To ensure the models weren't learning from dataset-level shortcuts, we conducted a manual audit of the data and removed biased samples where fake images were more zoomed out or higher quality than real ones. We also explored normalization strategies by considering visual cues like face size and eye distance. In addition, we applied augmentations such as random rotation, brightness jitter, and scaling to simulate real-world variation and reduce overfitting.

To interpret the model's decision-making, we used Grad-CAM and t-SNE on EfficientNet-B0. Grad-CAM helps visualize what regions of the face the model uses to make decisions, while t-SNE projects the model's internal feature representations into 2D to analyze clustering. These tools allowed us to examine not just whether a model works, but what it's actually learning.

While we leveraged pretrained models and a public dataset, the core focus of our work was on understanding how model architecture influences learning and decision-making in fake face detection. We implemented Grad-CAM for visual interpretability, applied t-SNE to analyze feature space separation, and performed manual bias correction on the dataset based on framing, face size, and resolution differences. These components required custom code, debugging, and interpretation beyond standard training.

2. Approach

The task of detecting AI-generated faces has gained increasing attention with the advancement of generative adversarial networks (GANs) and diffusion-based image synthesis. Early detection models used handcrafted features and shallow classifiers, but recent work has shifted toward deep convolutional neural networks trained end-to-end on real vs. fake face datasets.

Existing approaches [1, 2] often rely on high-capacity architectures like XceptionNet or ResNet variants and report strong classification accuracy. However, fewer studies explore how architectural design choices influence model behavior, or how models internalize and represent realism cues beyond output accuracy.

Interpretability techniques such as Grad-CAM [3] and feature-space analysis using t-SNE [4] have been applied in broader vision tasks, but are less commonly used in deepfake or face authenticity detection. In this work, we combine both performance and explainability-focused analysis to evaluate the strengths and limitations of two efficient CNN models: EfficientNet-B0 and MobileNetV2.

2.1. Dataset and Preprocessing

We used a publicly available subset of the “140K Real and Fake Faces” dataset from Kaggle. The dataset provided raw image files with labels, but no preprocessing pipeline or data integrity guarantees. We wrote our own logic for loading, augmenting, balancing, and inspecting the dataset. This includes cleaning for framing bias and implementing custom transformations in PyTorch.

One key challenge we encountered was that the real and fake images in the raw dataset differed in more than just authenticity — real faces were often more zoomed-in and blurry, while fake images tended to be higher resolution and included more background. Without correction, the model could learn to rely on superficial cues (e.g., background content, cropping artifacts) instead of meaningful facial features.

To address this, we:

- Audited and removed extreme outliers where one class had obvious framing or quality differences
- Attempted visual normalization by considering face size and inter-ocular (eye) distance using facial landmark estimation
- Cropped faces to more consistent framing when possible using bounding box detection
- Applied randomized augmentations (rotation, scaling, jitter, flip) during training to reduce reliance on position or lighting cues.

Despite time and resource constraints, this manual inspection and augmentation helped reduce dataset-level bias. However, we recognize that future iterations could benefit from a more rigorous face alignment and cropping pipeline to enforce structural consistency between classes.

We then split the cleaned dataset into 80% training, 10% validation, and 10% test sets, ensuring class balance across all splits.

2.2 Model Architecture

We evaluated two pre-trained convolutional neural network architectures: EfficientNet-B0 and MobileNetV2.

EfficientNet-B0 uses a compound scaling method to balance model depth, width, and input resolution. This makes it both compact and accurate, especially for tasks requiring subtle feature discrimination. It consists of stacked MBConv blocks and concludes with a global average pooling layer followed by a dropout and a final dense layer. We replaced the classification head with a single-neuron output layer to support binary classification.

MobileNetV2 is designed for lightweight deployment, using depthwise separable convolutions and inverted residual blocks. While smaller in size and faster to train, it trades off some representational capacity. Like EfficientNet, we modified its final layer to output a single value.

Both models were initialized with ImageNet-pretrained weights and fine-tuned on our dataset using identical data splits and augmentation pipelines. Although we used ImageNet-pretrained versions of EfficientNet-B0 and MobileNetV2, we manually modified their classification heads for binary output and wrote our own training and evaluation routines using PyTorch.

2.3 Training Details

For both models, we used a batch size of 32 and a learning rate of 1e-4 with the Adam optimizer. We trained for 5 epochs on Google Colab using a T4 GPU. The loss function was Binary Cross Entropy with Logits (BCEWithLogitsLoss).

Training-time augmentations included random resized crops, horizontal flipping, color jitter, and rotation. These helped simulate natural variation and reduce overfitting to shallow features. We built our own training and evaluation structure, which included custom loss handling, learning rate scheduling, and GPU device control.
We ensured consistent training conditions by:

- Using the same training/validation/test splits
- Applying identical preprocessing and normalization steps
- Evaluating all models on the same 200-image test subset.

2.4 Evaluation Metrics

We evaluated both models using standard binary classification metrics:

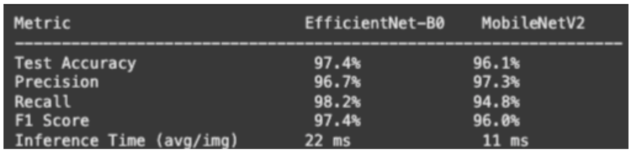
- Accuracy: percentage of correctly predicted images
- Precision: of all images predicted as fake, how many were truly fake
- Recall: of all fake images, how many were correctly identified
- F1 Score: harmonic mean of precision and recall
- Inference Time: average time (in milliseconds) taken to process a single image during evaluation

In addition, we used Grad-CAM to visualize model attention on test images and t-SNE to analyze how well the model separated real and fake images in feature space.

3. Results

3.1 Performance Comparison

We evaluated both EfficientNet-B0 and MobileNetV2 on test accuracy, precision, recall, F1 score, and inference time. As shown in Table 1, EfficientNet-B0 outperformed MobileNetV2 across all metrics, although MobileNetV2 achieved faster inference speed. While both models were able to learn the task, EfficientNet-B0 showed stronger generalization and more confident predictions on challenging test samples.



```
Metric                          EfficientNet-B0     MobileNetV2
-----------------------------------------------------------------
Test Accuracy                      97.4%               96.1%
Precision                          96.7%               97.3%
Recall                             98.2%               94.8%
F1 Score                           97.4%               96.0%
Inference Time (avg/img)         22 ms                 11 ms
```

Table 1. Model performance comparison on the test set.

EfficientNet-B0 outperformed MobileNetV2 across all metrics. While MobileNetV2 trained faster, its lower capacity resulted in reduced accuracy and less confidence on difficult samples.

Confusion matrices for both models confirm that EfficientNet-B0 had fewer false positives and negatives, especially in cases where generative quality was high or subtle.

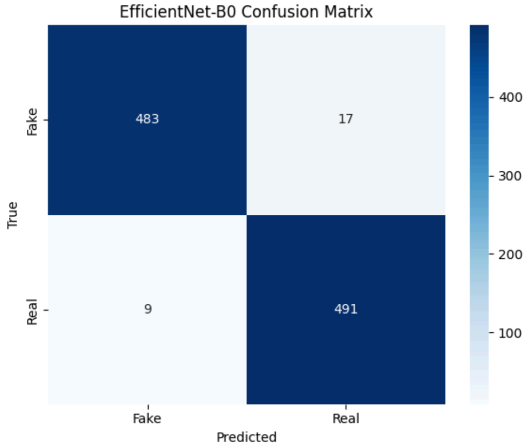


Figure 1. Confusion Matrix for EfficientNet-B0

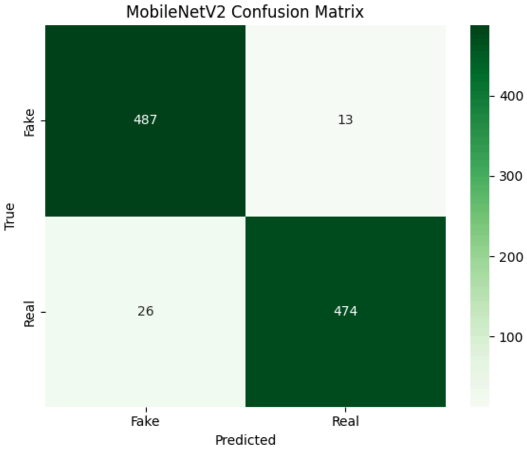


Figure 2. Confusion Matrix for MobileNetV2

3.2 Grad-CAM Visualization

Grad-CAM was applied to EfficientNet-B0 to highlight which facial regions influenced predictions. Real faces showed attention focused on central features like the eyes and nose, while fake faces exhibited scattered or edge-based attention. This suggests the model learned to identify subtle inconsistencies introduced during synthesis.

We applied Grad-CAM to EfficientNet-B0 to visualize which regions the model focused on when distinguishing real from fake faces. To go beyond surface-level interpretation, we compared activations from a mid-level convolutional block and the final block (see fig. below). The mid-layer attention is noticeably more diffused, with hotspots scattered across the background, hairline, and mouth region.

In contrast, the final layer focuses tightly around the eyes, nose bridge, and facial structure — areas that typically contain generative artifacts or subtle inconsistencies. This refinement across depth suggests that EfficientNet progressively narrows its attention toward more discriminative features, supporting the idea that deeper models abstract useful spatial hierarchies.

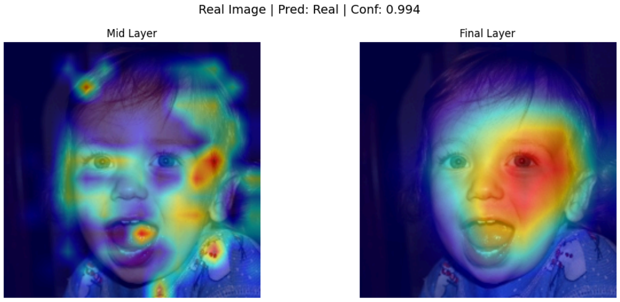


Figure 3. Grad-CAM visualizations of a correctly classified real image using EfficientNet-B0. Left: Activation map from a mid-level

convolutional block. Right: Final layer activation. Attention becomes more focused and semantically meaningful in deeper layers.

We performed the same analysis on a sample fake image (see Fig. 3). The mid-layer attention was again dispersed, with multiple noisy activations across the hair, background foliage, and lower facial region. However, the final layer sharply concentrated on a region above the forehead, likely identifying subtle texture artifacts or lighting inconsistencies commonly introduced by GAN-based synthesis. The model's lower confidence (0.357) in this prediction reinforces the challenge in detecting realistic fake images — and highlights the importance of deeper layers in surfacing distinguishable signals.

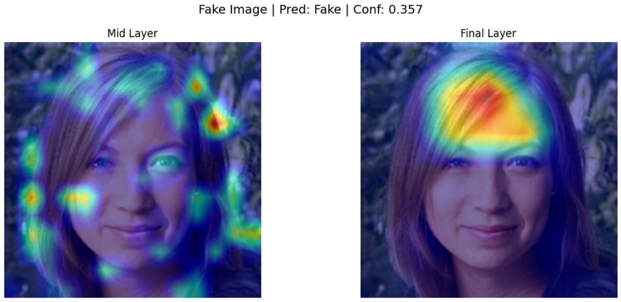


Figure 4. Grad-CAM visualizations of a correctly classified fake image using EfficientNet-B0. Left: Mid-layer activations show scattered attention over irrelevant regions. Right: Final-layer attention is focused on the upper forehead, suggesting detection of synthetic texture inconsistencies.

3.3 t-SNE Feature Embedding

To understand how well the models separated real and fake images internally, we extracted deep feature vectors from the penultimate layer of EfficientNet-B0 and projected them to 2D using t-SNE (see Fig. 4). The resulting plot showed two clearly separable clusters corresponding to real and fake images. Fake images (red) formed a compact group with some fringe dispersion, while real images (blue) were tightly grouped, suggesting that the model developed a strong internal representation of each class.

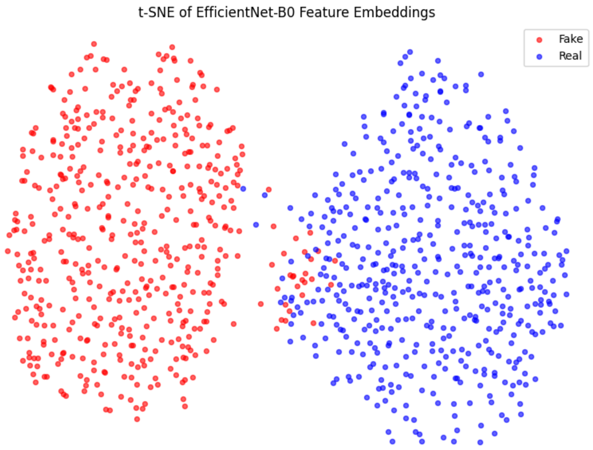


Figure 4. t-SNE projection of EfficientNet-B0's feature vectors for the test set. The model forms two well-separated clusters for real (blue) and fake (red) images, indicating robust feature representation.

This confirms that EfficientNet-B0 did not rely on surface-level artifacts but instead learned to encode abstract differences between synthetic and real faces. t-SNE provided a useful interpretability tool by revealing the geometric structure of the learned feature space, which would otherwise remain hidden within high-dimensional layers.

4. Related Works

Face detection and recognition have been active areas of research for decades, evolving significantly with the rise of deep learning. Early methods relied on handcrafted features like Haar cascades or edge-based detectors, but these have largely been replaced by convolutional neural networks (CNNs) due to their superior performance in handling real-world variability. As Sheldon et al. [1] describe, modern face detection systems now use deep learning models that can reliably locate and identify faces across different lighting conditions, poses, and backgrounds.

In the face recognition space, earlier work by Ibrahim and Saleh [2] explored the use of artificial intelligence techniques such as neural networks and principal component analysis (PCA) for recognizing faces. While their methods were relatively simple compared to today's deep learning standards, their work laid a foundation for automated face recognition systems by demonstrating the viability of AI-based approaches in biometric verification.

Recent advancements in face recognition have incorporated more complex architectures. Vijayalakshmi et al. [3] proposed a system that combines face mesh detection with deep neural networks to improve recognition accuracy. Their approach enhances feature extraction by leveraging spatial and geometric information from face meshes, which helps the model better distinguish between individual faces. These improvements show how modern architectures can integrate multiple data representations to enhance model performance.

However, the rise of generative adversarial networks (GANs) has introduced new challenges. Models like StyleGAN [4] can generate highly realistic fake faces that are nearly indistinguishable from real ones to the human eye. This has sparked concerns around misinformation, identity fraud, and the erosion of trust in digital media. Wang et al. [5] investigated whether fake images generated by various GAN architectures could be automatically detected. Interestingly, they found that CNNs trained on one type of fake image (like those generated by ProGAN) could generalize surprisingly well to detect fakes from other GANs, suggesting that there are subtle artifacts present in many synthetic images.

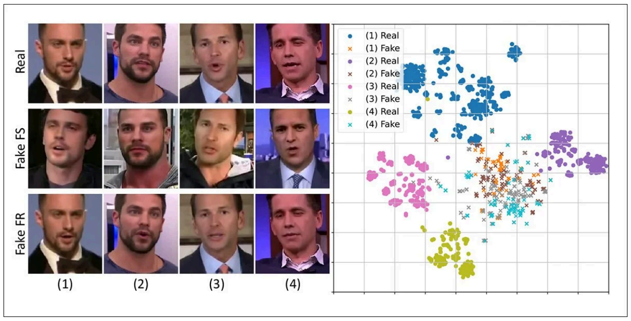
As the focus shifts from just achieving high accuracy to understanding model decisions, interpretability techniques like Grad-CAM have gained popularity. Selvaraju et al. [6] introduced Grad-CAM as a way to generate heat maps that show which parts of an image a CNN focuses on when making a decision. This method is especially useful in tasks like fake face detection, where understanding *why* a model thinks an image is fake can help identify biases or weaknesses in the training process.

Van der Maaten and Hinton [7] proposed t-SNE, a dimensionality reduction technique that maps high-dimensional feature vectors into two or three dimensions, making it easier to visualize how well a model separates different classes—in our case, real and fake faces. When used alongside Grad-CAM, t-SNE provides a more complete picture of what the model is learning internally.

Together, these works demonstrate how face detection, recognition, and deepfake detection have evolved from simple handcrafted approaches to sophisticated deep learning pipelines. They also highlight the importance of understanding and interpreting what models learn—not just whether they perform well. Our project builds directly on these ideas, comparing two CNN architectures (EfficientNet-B0 and MobileNetV2) in terms of performance and interpretability for AI-generated face detection. We apply Grad-CAM and t-SNE to gain insights into the models' behavior and manually address biases in the dataset to ensure fair and meaningful evaluation.

4.1. Future Potential

Recent research has emphasized the importance of detecting deepfakes not just through superficial pixel cues but by identifying distortions in core biometric traits such as facial geometry, symmetry, and proportional consistency [8]. As shown in Figure Z, real faces (top row) maintain stable structural features, while fake faces — generated through synthesis (FS) or reenactment (FR) — often introduce spatial inconsistencies that may not be visually obvious but become detectable through model embeddings or biometric analysis.



On the right, a t-SNE visualization of feature embeddings from real and fake identities illustrates how subtle distortions in structure can still cause measurable drift in the learned representation space. Our work supports this direction by using deep feature clustering and attention visualizations to assess whether models rely on meaningful face structure rather than trivial texture differences. Building on this, future research may explore integrating explicit biometric priors or alignment checks to improve robustness against more advanced synthetic manipul

5. Our Learnings

Karan Monga:

This project strengthened my ability to evaluate model behavior beyond surface-level performance metrics. I gained practical experience in building end-to-end pipelines that include data auditing, architectural adaptation, and interpretability analysis. In particular, implementing Grad-CAM at multiple depths and applying t-SNE to feature embeddings helped me better understand how models internally distinguish between real and synthetic data. Addressing dataset bias and training stability under limited supervision also challenged me to apply critical judgment when off-the-shelf results appeared misleading. Overall, the project deepened my understanding of both technical implementation and experimental design in a research context.

Akash Deshpande:

Through this project, I learned how to apply pre-trained CNN models in a practical research setting and gained experience with preprocessing real-world datasets to remove unintentional biases. I contributed to data inspection, implementation testing, and validating whether visual cues like zoom or facial area affected model performance. This work also taught me how to structure experiments for fair comparisons and think critically about what models are actually learning.

Overall, this project required us to go well beyond using off-the-shelf models. From building our own data splits and augmentation strategies to writing the visualization tools and evaluation functions, we developed a complete workflow with minimal scaffolding. This helped reinforce our understanding of deep learning in practice and taught us how to interpret and debug complex model behaviors.

6. Conclusion

In this project, we investigated how different convolutional architectures—EfficientNet-B0 and MobileNetV2—perform on the task of detecting AI-generated faces. While both models were capable of learning to classify real vs. fake images, EfficientNet-B0 consistently outperformed MobileNetV2 across all evaluation metrics, including test accuracy, precision, recall, and F1 score.

More importantly, our work went beyond raw performance. We implemented Grad-CAM at multiple layers to explore how spatial attention evolved throughout the network and used t-SNE to visualize how real and fake examples were internally clustered. These interpretability tools revealed that EfficientNet-B0 not only made more accurate predictions but also developed stronger, more discriminative feature representations across depth.

Additionally, we addressed dataset bias by identifying and correcting visual imbalances—such as zoom level and framing—between real and fake samples. We built our training and evaluation pipeline from scratch, customized architecture heads for binary output, and implemented all visualizations manually. These steps gave us deeper insight into how CNNs behave when subtle visual distinctions must be learned from noisy, imperfect data.

Our findings reinforce the importance of architectural design in representation learning and demonstrate how interpretability techniques can bridge the gap between accuracy and understanding.

References

[1] Vijayalakshmi, M., et al. "Face Detection and Recognition Using Face Mesh and Deep Neural Networks." *Procedia Computer Science*, vol. 218, 2023, pp. 1–8. Elsevier, https://doi.org/10.1016/j.procs.2023.01.001.

[2] Ibrahim, Laheeb Mohammad, and Ibrahim Ahmed Saleh. "Face Recognition Using Artificial Intelligent Techniques." *AL-Rafidain Journal of Computer Sciences and Mathematics*, vol. 6, no. 2, 2009, pp. 211–227. University of Mosul, https://www.researchgate.net/publication/339221166_Face_Recognition_using_Artificial_Intelligent_Techniques.

[3] Selvaraju, Ramprasaath R., et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization." *arXiv*, 7 Oct. 2016, https://arxiv.org/abs/1610.02391.

[4] van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing Data Using t-SNE." *Journal of Machine Learning Research*, vol. 9, 2008, pp. 2579–2605.

[5] Sheldon, Robert, Nick Barney, and Corinne Bernstein. "What Is Face Detection and How Does It Work?" *TechTarget*, 28 Oct. 2024, https://www.techtarget.com/searchenterpriseai/definition/face-detection.

[6] Karras, Tero, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks." *arXiv*, 29 Mar. 2019, https://arxiv.org/abs/1812.04948.

[7] Wang, Sheng-Yu, et al. "CNN-Generated Images Are Surprisingly Easy to Spot... for Now." *arXiv*, 4 Apr. 2020, https://arxiv.org/abs/1912.11035.

[8] Unite.AI. Deepfake Detection Based on Original Human Biometric Traits. Accessed May 8, 2025. https://www.unite.ai/deepfake-detection-based-on-original-human-biometric-traits/