Distinguishing AI-Generated Faces from Real Humans

Yusuf Samed Çelik, Zehra Selin Karabıçak
Akdeniz University, Department of Computer Engineering

Abstract

Problem Statement: The rapid and sophisticated advancements in Generative Adversarial Networks (GANs) have enabled the creation of hyper-realistic human face images. These synthetic faces are often visually indistinguishable from authentic photographs by human observers, presenting a significant challenge. The escalating ease of generating such content without reliable automated identification methods poses considerable risks, including the proliferation of misinformation campaigns, the creation of fraudulent online personas for malicious activities like phishing or social engineering, the undermining of digital evidence integrity, and a general erosion of public trust in visual media.
Importance of the Problem: In an era dominated by social media and rapid digital content dissemination, the unchecked spread of deceptive AI-generated imagery can have severe societal consequences. Establishing robust mechanisms to accurately discriminate between genuine human faces and those synthesized by AI is therefore of paramount importance. Such capabilities are crucial for preserving the integrity of digital information ecosystems, safeguarding individual and collective security against identity-related fraud, combating the malicious exploitation of AI technologies, and ultimately maintaining public confidence in the authenticity of visual content encountered daily.
Proposed Solution and Methodology: This project presents the development and evaluation of a deep learning-based computer vision system specifically designed to identify AI-generated facial images. A comprehensive custom dataset was meticulously constructed by collecting real human face images from `whichfaceisreal.com` and diverse AI-synthesized faces from `thispersondoesnotexist.com`. These images underwent a standardized preprocessing pipeline, including automated face detection, precise facial region cropping, orientation correction for consistency, and resizing to 224×224 pixels. Several Convolutional Neural Network (CNN) architectures were investigated: fine-tuning prominent pre-trained models (EfficientNetB0, ResNet50, MobileNetV2) and training a custom-designed CNN from scratch. To enhance model robustness and generalization capabilities, extensive data augmentation techniques were systematically applied during training. The models were trained employing established best practices such as early stopping to prevent overfitting, learning rate scheduling for optimal convergence, and a dataset split of 70% for training, 20% for validation, and 10% for rigorous testing.
Key Results: Among the evaluated architectures, the EfficientNetB0 model (designated as version v2) demonstrated superior performance. It achieved an accuracy of 86.89%, a Receiver Operating Characteristic Area Under Curve (ROC AUC) score of 0.9083, and a Precision-Recall Area Under Curve (PR AUC) score of 0.9212 on the unseen test set. These metrics indicate that the model effectively learned to discern subtle discriminative artifacts present in AI-generated faces from the natural features of real human faces.
Expected Outcomes and Significance: This research contributes an effective and systematically evaluated classifier for distinguishing AI-generated faces from genuine photographs. The findings provide valuable insights into the characteristic visual artifacts and statistical irregularities that current generative models may inadvertently introduce. The developed system and the methodologies employed can be adapted and extended for broader applications in content moderation, digital forensics, and the automated detection of manipulated media, thereby playing a role in fortifying digital trust and authenticity. Future work will focus on expanding dataset diversity, exploring more advanced model architectures, and improving robustness against evolving GAN technologies.
Keywords: Computer Vision, Deep Learning, AI-Generated Images, GAN Detection, Fake Face Detection, EfficientNet, Image Forensics, Digital Media Integrity.

1 Introduction

Generative Adversarial Networks (GANs) have undeniably revolutionized the landscape of digital image synthesis, particularly in their ability to generate human facial images of astonishing realism [8]. Originating from Ian Goodfellow's seminal work, GANs employ a dual-network architecture—a generator that creates data samples and a discriminator that attempts to distinguish these fakes from real samples—engaging in a continuous adversarial game that progressively refines the generator's output quality. Early models, such as the Deep Convolutional GAN (DCGAN), laid foundational principles but often struggled with training stability, mode collapse (limited diversity in generated samples), and achieving high-resolution outputs. However, subsequent architectural innovations, including Progressive GANs (which incrementally

increase resolution during training) and StyleGAN (which offers unprecedented control over stylistic attributes of the generated faces at different scales), have dramatically elevated the quality and resolution of synthesized faces [9]. These advancements, while marking significant milestones in AI capabilities, have concurrently amplified the potential for misuse. The ease with which highly realistic, yet entirely fabricated, facial images can be created necessitates the development of sophisticated and robust detection mechanisms to identify such AI-generated content.

The capacity to generate convincing fake faces carries profound societal and technological implications. Malicious actors can exploit these capabilities for a wide array of nefarious purposes. These include, but are not limited to, the creation of inauthentic social media profiles for large-scale disinformation campaigns or targeted social engineering attacks, the fabrication of false evidence in legal or personal disputes, the generation of synthetic identities for financial fraud or to bypass identity verification systems, and the spread of propaganda or defamatory content. The sheer volume and increasing verisimilitude of these synthetic images pose a substantial threat to the integrity of visual information, the trustworthiness of online interactions, and ultimately, public trust in digital media. Consequently, the task of developing effective, scalable, and reliable techniques to automatically distinguish AI-generated faces from those of real individuals has emerged as a critical and rapidly evolving research frontier within computer vision, digital forensics, and information security. Addressing this challenge is not merely a technical endeavor but a crucial step towards mitigating the adverse impacts of synthetic media on society.

This paper presents a comprehensive investigation into the development and rigorous evaluation of a deep learning model tailored for the detection of AI-generated facial images. We provide a detailed account of our methodology, beginning with the systematic creation of a diverse dataset, which involved the collection of both authentic human faces and a variety of synthetically generated faces from publicly accessible online sources. This is followed by a description of our meticulous preprocessing pipeline designed to standardize the images for model input. A significant portion of our work involved exploring various Convolutional Neural Network (CNN) architectures. This included the fine-tuning of powerful pre-trained models such as EfficientNetB0, ResNet50, and MobileNetV2—chosen for their proven performance on large-scale image recognition tasks—as well as the design and training of a custom CNN architecture from fundamental principles to serve as a baseline and explore tailored feature extraction. Our study places considerable emphasis on the strategic application of a wide range of data augmentation techniques to enhance model generalization and prevent overfitting, and the implementation of robust training strategies, including early stopping and adaptive learning rates, to achieve optimal classification accuracy. The results demonstrate that our best-performing model, based on the EfficientNetB0 architecture, achieves notable success in this challenging detection task. This work aims to contribute to the collective, ongoing efforts to combat the proliferation of synthetic media by providing an effective, well-documented, and reproducible detection system. Furthermore, we critically analyze the challenges encountered during the project, discuss the limitations of our current approach, and outline promising avenues for future research to address the continuous evolution of GAN technologies and the increasingly sophisticated nature of generated content. The insights gained are intended to benefit researchers and practitioners working towards ensuring a more transparent and trustworthy digital environment.

2 Literature Review

The challenge of detecting GAN-generated images, particularly synthetic faces, has spurred a significant and diverse body of research. Existing methodologies can be broadly classified based on the types of artifacts or features they are designed to exploit, ranging from low-level pixel statistics to more abstract physiological inconsistencies. This section reviews key contributions and trends in this domain.

2.1 Deep Learning-based Approaches

The dominant paradigm for GAN-generated image detection leverages the power of deep learning, particularly Convolutional Neural Networks (CNNs), renowned for their ability to automatically learn hierarchical feature representations from raw pixel data. Mo et al. [4] were among the early proponents, training a bespoke CNN architecture that learned discriminative features directly from pixels and intermediate convolutional layers. Their model, when trained on a dataset comprising real images and images from several early GAN models, achieved an impressive accuracy of 99.4%, demonstrating the fundamental viability of CNNs for this task. However, early datasets were often limited in diversity of GAN architectures.

Afchar et al. [1] introduced MesoNet, a compact CNN architecture specifically designed for facial video forgery detection, including DeepFakes (which often involve GAN-like components for face synthesis). MesoNet featured a dual-branch design: one branch focused on capturing fine-grained textural details, often indicative of manipulation, while the other processed global facial structures. This architecture demonstrated robust performance, achieving around 98% accuracy on standard DeepFake benchmarks. The authors emphasized the importance of shallow

Figure 1: *Example of a real human face image from our dataset (left) and an AI-generated face image (right). Note the high degree of realism in the synthetic image, highlighting the detection challenge.*

networks for detecting subtle, low-level artifacts that deeper networks might overlook.

R"ossler et al. [8] made a significant contribution with the FaceForensics++ dataset, a large-scale benchmark for facial manipulation detection, including expression swaps, face swaps, and entire face synthesis using various techniques. They benchmarked several established CNN architectures, notably XceptionNet, and found that networks pre-trained on ImageNet and subsequently fine-tuned on manipulation-specific data outperformed generic classifiers. Their work highlighted that features learned for general object recognition can be effectively transferred but specialized training is crucial for optimal performance against specific manipulation types. They also showed that detection difficulty increases with higher quality image compression, a common real-world scenario.

These foundational studies underscore the capacity of CNNs to learn subtle discriminative features that differentiate real images from GAN-generated ones. The focus has often been on identifying common artifacts such as unnatural textures, inconsistent lighting, or checkerboard patterns that were prevalent in earlier GAN outputs. However, as GANs evolve, these artifacts become less pronounced, pushing researchers to develop more sophisticated deep learning solutions. Many contemporary approaches explore attention mechanisms to guide the network's focus towards anamolous regions, or utilize more advanced pre-trained backbones like EfficientNets or Vision Transformers. The challenge remains in creating detectors that generalize well to unseen GAN architectures and are robust to various post-processing techniques designed to evade detection.

2.2 Co-occurrence and Frequency Domain Methods

While end-to-end deep learning is powerful, some methods incorporate more explicit forensic feature extraction, often focusing on statistical irregularities in the pixel domain or transformations into the frequency domain. These approaches hypothesize that the GAN generation process, despite its sophistication, may leave behind statistical traces that are not typical of natural images.

Nataraj et al. [7] proposed a method based on analyzing co-occurrence matrices derived from image pixels. Co-occurrence matrices capture the spatial relationships between pixel intensities and can reveal subtle textural differences. They computed these matrices across RGB channels and used them as input features to a CNN classifier. This technique reportedly achieved detection rates exceeding 99% on certain datasets, suggesting that GANs might not perfectly replicate the complex statistical dependencies found in natural image textures. The strength of this approach lies in its ability to capture higher-order statistical information that might be missed by standard convolutional filters focusing on local patterns. However, the computational cost of co-occurrence matrices can be a factor, and their sensitivity to image compression and resizing needs careful consideration.

Barni et al. [2] further refined the co-occurrence-based approach by integrating cross-band co-occurrences. Their method aimed to enhance robustness against common image processing operations like JPEG compression and geometric transformations (e.g., scaling, rotation), which can often degrade or obscure forensic traces. By analyzing statistical correlations not just within individual color channels but also *between* them, their detector showed improved resilience in more diverse and challenging operational settings. This line of research highlights a productive synergy between handcrafted statistical feature engineering and the pattern recognition capabilities of deep learning. Frequency domain analysis, using techniques like the Discrete Fourier Transform (DFT) or

Discrete Cosine Transform (DCT), has also been explored extensively. It's hypothesized that GANs might introduce specific frequency artifacts; for instance, upsampling stages in generator networks can sometimes lead to characteristic periodic patterns or unnatural distributions of energy in the frequency spectrum. Detecting these patterns can provide clues about the image's synthetic origin, and several works have successfully built classifiers based on features extracted from DFT coefficients or DCT blocks.

2.3 Physiological Artifact Analysis

Another promising avenue for GAN face detection involves scrutinizing images for inconsistencies in physiological features—aspects of human anatomy and appearance that GANs may struggle to replicate with perfect fidelity. This approach leverages domain-specific knowledge about human faces, as certain biological structures are complex and highly variable.

Guo et al. [3] presented a compelling study focused on the morphology of pupils in eyes. They observed that GAN-generated faces often exhibit irregularly shaped or asymmetrically sized pupils. This is likely because GANs, while proficient at overall facial structure, may not have sufficiently detailed or consistent models of intricate biological components like the eye, especially under varied head poses, expressions, and lighting conditions. Their method involved segmenting the eye regions and then applying shape descriptors to analyze pupil geometry. This technique demonstrated robust detection capabilities, even when parts of the face were occluded, as eyes are often visible and critical for human perception.

Mohzary et al. [5] introduced the CHIEFS framework, which analyzes corneal-specular highlights—the small, bright reflections of light sources that appear on the cornea of the eye. In natural photographs, these highlights are consistent with the scene's illumination environment and the 3D geometry of the eye. GANs, however, may generate faces with inconsistent, misplaced, misshapen, or even missing specular highlights, or highlights that do not correspond to a plausible 3D eye model and lighting setup. The CHIEFS system constructs high-dimensional feature vectors from these highlights to capture illumination inconsistencies, achieving high accuracy (up to 99%) across varied lighting conditions. These studies compellingly demonstrate that focusing on biologically constrained and physically plausible features can yield powerful forensic indicators. Other physiological cues explored in the literature include inconsistencies in dental structure (e.g., unnatural regularity or imperfections), ear morphology (which is highly unique), or subtle asymmetries in facial hair or skin texture that are common in real faces but may be smoothed out or unnaturally regularized by GANs.

2.4 Hybrid Fusion Techniques

Recognizing that different detection approaches have unique strengths and weaknesses, hybrid methods aim to combine complementary cues from various sources—spatial domain, frequency domain, and physiological analyses—to create more robust and generalizable detectors. The idea is that a fusion of diverse forensic signals can lead to a more comprehensive and resilient detection system.

Xue et al. [10] proposed GLFNet (Global–Local Facial Fusion Network), which employs a dual-stream CNN architecture. One stream, the global branch, processes the entire face image to capture holistic features, overall facial structure, and contextual information. Simultaneously, a local branch focuses specifically on detailed analysis of critical facial regions, such as the irises, mouth, and nose, which are known to be rich in subtle GAN artifacts and physiological details. The features extracted from these two streams are then intelligently fused, typically through concatenation or attention mechanisms, to make a final classification. This approach demonstrated good performance, particularly in maintaining accuracy even when images were subjected to heavy compression, which often erodes high-frequency details that some detectors rely on.

Mundra et al. [6] focused on developing efficient detection methods suitable for large-scale applications, such as screening profile photos on social media platforms where millions of images might need to be processed daily. They explored the creation of compact embeddings (low-dimensional feature representations) that retain discriminative information for GAN detection. Their work aimed to strike a balance between high detection accuracy and low computational overhead, making the method practical for real-time or high-throughput scenarios. This often involves model compression techniques (like quantization or pruning) applied to more complex feature extractors, or designing inherently lightweight architectures that still capture salient forensic features. The fusion in such hybrid models can occur at different levels: early fusion (combining raw inputs or shallow features from different modalities), late fusion (combining scores or decisions from independent classifiers trained on different cues), or intermediate fusion (combining features at deeper layers of a unified network). Each fusion strategy has its own trade-offs in terms of complexity and effectiveness.

2.5 Survey and Comparative Analyses

Given the rapid proliferation of detection methods, comprehensive survey papers and comparative analyses play a crucial role in structuring the field and guiding future research. Wang et al. [9] provided a notable survey that categorizes existing approaches into domains such as deep learning-based, statistical

feature-based, and physiological cue-based methods. Their work offers valuable comparative insights into the accuracy, computational complexity, robustness to countermeasures, and generalization capacity of different techniques across various datasets and GAN architectures. Such surveys also critically highlight persistent challenges in the field. These include the "arms race" against ever-improving GANs (requiring detectors to constantly adapt to novel generative models without extensive retraining), the need for privacy-preserving forensic techniques (especially when dealing with sensitive facial data), ensuring fairness and mitigating biases in detection algorithms (e.g., performance disparities across different demographic groups), and the scalability of detectors for deployment in high-throughput environments like social media content moderation systems. These taxonomies and critical reviews serve as essential roadmaps, identifying research gaps and charting pathways for future innovation in the ongoing effort to safeguard the integrity of visual media. They also emphasize the importance of standardized evaluation protocols and diverse, challenging benchmark datasets for fair comparison of emerging techniques.

3 Methodology

This section provides a detailed exposition of the systematic methodology employed in our project to develop a robust system for distinguishing AI-generated faces from authentic human faces. Our comprehensive approach encompasses several key stages: dataset acquisition and meticulous preparation, strategic application of data augmentation techniques, careful selection and configuration of various model architectures, and the implementation of effective training and evaluation protocols.

3.1 Dataset Acquisition and Preparation

The foundation of any successful machine learning project, particularly in computer vision, is a well-curated and representative dataset. Significant effort was dedicated to assembling a dataset suitable for training and evaluating our detection models.

- **Data Sources and Collection Strategy:** Authentic human face images were systematically scraped from the website `whichfaceisreal.com/index.php`. This platform presents users with pairs of faces (one real, one AI-generated) and allows them to guess the real one, often providing feedback, which indirectly curates a collection of verified real faces. AI-generated faces were primarily sourced from `thispersondoesnotexist.com`, a well-known website that showcases an endless stream of novel, high-quality synthetic faces generated by NVIDIA's StyleGAN (and its variants). This source ensures a continuous supply of diverse, state-of-the-art AI faces. The scraping process involved automated scripts to download images, with care taken to respect website terms of service and avoid overloading servers. We aimed for a roughly balanced number of images from each class to prevent initial class imbalance issues. Initially, several thousands of images were collected for each category, with a target of at least 2500 images per class for the training set after preprocessing and cleaning.
- **Image Preprocessing Pipeline:** The raw images obtained from these sources varied in size, composition, and minor artifacts. To ensure consistency and prepare them for input into our neural networks, a standardized preprocessing pipeline was applied:
  1. **Face Detection and Cropping:** The first step involved accurately detecting the primary facial region in each image. We utilized a pre-trained Haar Cascade classifier and, in some cases, a more robust MTCNN (Multi-task Cascaded Convolutional Networks) implementation, to locate facial landmarks and define a bounding box around the face. The images were then cropped to this bounding box, often with a small margin (e.g., 10-15% of the face width/height) to include some context like forehead and chin, which are important for facial recognition. This step is crucial as it focuses the model's attention on the relevant facial area and removes distracting background elements.
  2. **Face Orientation Correction (Normalization):** Although most images from the sources were front-facing, minor variations in head pose were addressed. While full 3D alignment was not performed due to complexity, basic checks for extreme in-plane rotations (e.g., beyond ± 20 degrees) were considered, and images that were too skewed or where face detection failed with low confidence were discarded to maintain data quality. The goal was to have predominantly upright faces.
  3. **Image Resizing:** All cropped facial images were resized to a uniform input dimension of 512×51 pixels. This specific size is a common standard for many pre-trained CNN architectures (like EfficientNetB0, ResNet50) and provides a good balance between retaining sufficient facial detail and managing computational load during training. Bicubic interpolation was generally used for resizing as it tends to preserve image quality better

than simpler methods like bilinear or nearest-neighbor interpolation.

4. **Pixel Value Normalization:** The pixel values of the images, typically in the range [0, 255] for each RGB channel, were normalized to the floating-point range [0, 1] by dividing each pixel value by 255.0. This normalization step is standard practice and helps in stabilizing the training process, ensuring that all input features have a similar scale, which can lead to faster convergence for deep neural networks.

- **Data Storage and Organization:** The processed and curated dataset, comprising approximately 7100 images in total (e.g., 3550 AI-generated, 3550 real human faces), was organized into a structured directory format. The top-level directory contained two subdirectories: `data/ai_faces/` and `data/human_faces/`. This clear separation facilitated easy loading and management using data generators in TensorFlow/Keras. For larger-scale experiments and to leverage cloud computing resources, this dataset was also uploaded and managed within a Google Cloud Storage (GCS) bucket. This allowed for efficient, scalable access during model training, especially when utilizing Google Vertex AI for distributed training or hyperparameter optimization. Version control for the dataset (or at least its generation scripts and metadata) was considered to ensure reproducibility.

- **Dataset Splitting Strategy:** To ensure an unbiased and rigorous evaluation of our model's generalization capabilities, the entire dataset was carefully divided into three distinct, non-overlapping subsets:

  - **Training Set (70%):** Approximately 5000 images. This largest portion of the data was used exclusively for training the models, allowing them to learn the underlying patterns and discriminative features distinguishing AI from real faces.
  - **Validation Set (20%):** Approximately 1400 images. This set was used during the training process to monitor the model's performance on unseen data at the end of each epoch, guide hyperparameter tuning (e.g., learning rate adjustments based on validation loss), and implement early stopping to prevent overfitting.
  - **Test Set (10%):** Approximately 700 images. This final, held-out set was used only once after all training and model selection phases were complete to provide an objective assessment of the chosen model's performance on completely new data it had never encountered.

  The splitting was performed randomly but was stratified by class (AI-generated vs. Real) to ensure that each subset (train, validation, test) maintained a similar proportion of images from each category as present in the overall dataset. This prevents skewed evaluations due to class imbalances within the splits.

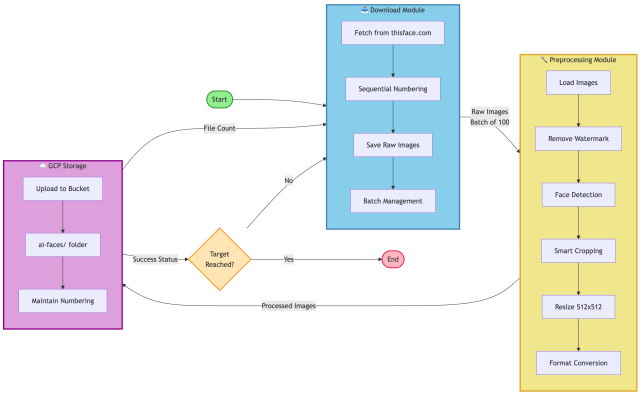


Figure 2: *Conceptual diagram of the image preprocessing pipeline, illustrating the sequence of operations from raw input image (left) through face detection, cropping, resizing, and normalization to produce a model-ready input tensor (right).*

3.2 Data Augmentation Techniques

Data augmentation is a critical technique for increasing the effective size and diversity of the training dataset, which in turn helps to improve model generalization and reduce overfitting, especially when the initial dataset size is limited. A variety of augmentation transformations were applied on-the-fly to the training images during the model training phase using TensorFlow's `ImageDataGenerator` or `tf.image` functionalities. The rationale was to expose the model to a wider range of plausible variations it might encounter in real-world scenarios.

- **Geometric Transformations:** These alter the spatial properties of the images.
  - Random Rotations: Images were randomly rotated by an angle in the range of ± 40 degrees. This helps the model become invariant to minor changes in facial orientation or camera tilt.
  - Random Width and Height Shifts: Images were randomly shifted horizontally and vertically by a fraction (up to 30%) of their respective dimensions. This simulates variations in image framing and subject position within the frame.

  - Shear Transformations: Shear mapping was applied with a shear intensity of up to 30%. This introduces a slanting effect, mimicking non-frontal poses or perspective distortions, further diversifying pose variations.
  - Random Zooming: Images were randomly zoomed in or out by a factor (up to 30%). This helps the model learn features at different scales and distances from the camera.
  - Horizontal Flipping: Images were randomly flipped horizontally with a 50% probability. This is a common and effective augmentation for facial images as faces are roughly symmetrical, and it effectively doubles the amount of pose information for lateral views.

- **Photometric Transformations:** These alter the pixel intensity values and color properties.
  - Brightness Adjustments: The brightness of images was randomly varied, for example, by picking a factor between 70% and 130% of the original brightness. This makes the model more robust to variations in lighting conditions (e.g., underexposed or overexposed images).
  - Color Channel Shifts: Minor random shifts were applied to the intensity of individual RGB color channels, simulating slight color imbalances or different camera sensor responses.
  - Contrast Adjustments: Image contrast was randomly adjusted to simulate different lighting scenarios and enhance robustness to variations in image quality.

- **Noise Injection:**
  - Random Gaussian Noise: A small amount of Gaussian noise (with a mean of 0 and a small standard deviation) was added to some images. This can improve robustness to noisy input images or minor image sensor imperfections often seen in real-world data.

These augmentations were carefully chosen and their parameters tuned to provide meaningful diversity without distorting the images to an extent that would make them unrecognizable or unrepresentative of the target classes. The on-the-fly application ensured that the model saw slightly different versions of the training images in each epoch, significantly expanding the effective dataset size.

3.3 Model Architectures Explored

We conducted experiments with a range of CNN architectures, from established pre-trained models to a custom-built network, to identify the most effective approach for our specific task. The input size for all models was consistently $224 \times 224 \times 3$.

3.3.1 EfficientNetB0

EfficientNet models, particularly EfficientNetB0, are renowned for achieving state-of-the-art accuracy on image classification tasks while being computationally efficient. This efficiency is achieved through a compound scaling method that uniformly scales network width, depth, and resolution in a principled way.

- **Rationale for Choice:** Chosen for its excellent balance of performance (accuracy) and parameter count/FLOPs, making it suitable for scenarios where computational resources might be a consideration, without significantly compromising on the ability to learn complex features. Its architecture is designed to capture multi-scale features effectively.

- **Configuration:** We utilized the EfficientNetB0 architecture pre-trained on the ImageNet dataset. The top classification layers of the pre-trained model were removed (by setting `include_top=False` in Keras) and replaced with a custom head designed for our binary (AI-generated vs. Real) classification task. This custom head typically involved:
  1. A `GlobalAveragePooling2D` layer to reduce the spatial dimensions of the feature maps from the EfficientNet base.
  2. A `Dense` layer with a significant number of units (e.g., 256 or 512) and `ReLU` activation for further feature transformation.
  3. A `Dropout` layer with a rate of 0.6 (as specified in the project notes) for regularization to prevent overfitting of this new head.
  4. A final `Dense` layer with a single neuron and a sigmoid activation function to output a probability score between 0 and 1, representing the likelihood of the input face being AI-generated.

- **Fine-tuning Strategy:** The entire network (both the pre-trained EfficientNetB0 base and the newly added custom head) was fine-tuned on our dataset. Initially, the base layers might be frozen for a few epochs to allow the custom head to stabilize, followed by unfreezing all layers and training with a very small learning rate to adapt the pre-trained features to our specific domain.

This architecture consistently yielded the best performance in our comparative experiments.

3.3.2 ResNet50

ResNet50 is a widely recognized deep residual network architecture with 50 layers. Its key innovation is the use of "skip connections" or "shortcuts" which allow gradients to propagate more easily through very deep networks, mitigating the vanishing gradient problem and enabling the training of deeper, more powerful models.

- **Rationale for Choice:** Selected to evaluate the performance of a deeper, well-established architecture known for its strong feature extraction capabilities and robustness across various vision tasks.
- **Configuration:** Similar to EfficientNetB0, ResNet50 pre-trained on ImageNet was used. The original fully connected classification layer was replaced with a custom classification head analogous to the one used for EfficientNetB0 (GlobalAveragePooling2D, Dense layer, Dropout of 0.6, and a sigmoid output layer).
- **Fine-tuning Strategy:** The entire network was fine-tuned using a similar strategy, potentially with differential learning rates for the base and the head.

3.3.3 MobileNetV2

MobileNetV2 is a lightweight deep neural network architecture specifically designed for mobile and embedded vision applications. It utilizes depthwise separable convolutions to significantly reduce the number of parameters and computational cost (FLOPs) compared to standard convolutions, while maintaining competitive accuracy.

- **Rationale for Choice:** Included to assess the feasibility of using a highly efficient model for scenarios requiring fast inference times or deployment on resource-constrained devices (e.g., mobile applications or edge AI systems).
- **Configuration:** Pre-trained MobileNetV2 was adapted with a similar custom classification head and dropout rate (0.6) as the other pre-trained models.
- **Fine-tuning Strategy:** Full network fine-tuning was employed.

3.3.4 Custom CNN (Trained from Scratch)

To establish a performance baseline and explore an architecture specifically tailored (albeit simply) to our dataset without reliance on pre-training, a custom CNN was designed and trained from scratch. This also helps in understanding the benefits gained from transfer learning.

- **Architecture Design Rationale:** The design aimed for a moderately deep network capable of learning hierarchical features, incorporating standard CNN components like convolutional layers, pooling layers, batch normalization, and dropout for regularization. The depth was chosen to be manageable for training from scratch on our dataset size.
- **Layer Configuration:**
  - **Input Layer:** Accepts $224 \times 224 \times 3$ images.
  - **Convolutional Block 1:** Convolutional layer (32 filters, kernel size (3,3), 'same' padding, ReLU activation), Batch Normalization, MaxPooling2D (pool size (2,2)), Dropout (0.25).
  - **Convolutional Block 2:** Convolutional layer (64 filters, kernel size (3,3), 'same' padding, ReLU activation), Batch Normalization, MaxPooling2D (pool size (2,2)), Dropout (0.25).
  - **Convolutional Block 3:** Convolutional layer (128 filters, kernel size (3,3), 'same' padding, ReLU activation), Batch Normalization, MaxPooling2D (pool size (2,2)), Dropout (0.25).
  - **Convolutional Block 4:** Convolutional layer (256 filters, kernel size (3,3), 'same' padding, ReLU activation), Batch Normalization, MaxPooling2D (pool size (2,2)), Dropout (0.25).
  - **Flatten Layer:** To convert the 2D feature maps from the last pooling layer into a 1D vector suitable for input to dense layers.
  - **Dense Layer 1:** Fully connected layer with 512 neurons, ReLU activation.
  - **Dropout Layer:** Dropout rate of 0.5 for regularization after the first dense layer.
  - **Dense Layer 2:** Fully connected layer with 128 neurons, ReLU activation.
  - **Dropout Layer:** Dropout rate of 0.5 for regularization after the second dense layer.
  - **Output Layer:** A final Dense layer with 1 neuron and a Sigmoid activation function to produce the binary classification probability (AI-generated or Real).

  The choice of filter counts (increasing with depth) and dropout rates was based on common practices for CNN design.

3.4 Training Process and Optimization

A consistent and robust training methodology was applied across all model architectures to ensure fair comparison and optimal performance.

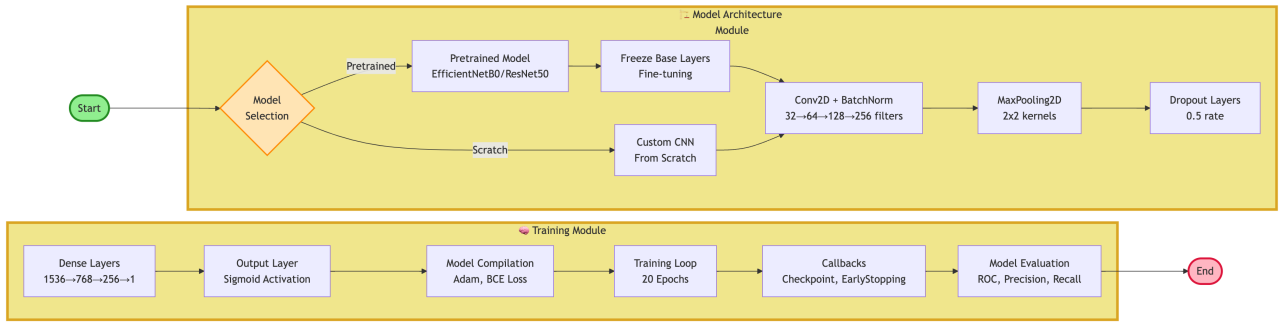


Figure 3: *Diagram illustrating the architecture of our custom Convolutional Neural Network (CNN) model, detailing the sequence of layers (convolutional, batch normalization, pooling, dropout, dense), filter counts, kernel sizes, and activation functions used.*

- **Deep Learning Framework:** All models were implemented, trained, and evaluated using Python with the TensorFlow (version 2.x) and Keras high-level API. This choice provides a flexible and powerful environment for deep learning research and development.
- **Optimizer Selection:** The Adam (Adaptive Moment Estimation) optimizer was chosen for its widespread effectiveness and adaptive learning rate capabilities, which generally make it perform well across a variety of tasks and architectures. An initial learning rate of 1×10^{-4} (0.0001) was typically used as a starting point, especially for fine-tuning pre-trained models, as larger rates can disrupt the learned weights. For the custom CNN trained from scratch, slightly higher initial learning rates (e.g., 1×10^{-3}) might also have been explored.
- **Loss Function:** Binary Cross-Entropy (also known as log loss) was employed as the loss function. This is the standard choice for binary classification problems where the model outputs a probability for one of the classes. The formula is given by $L = -(y \log(p) + (1-y) \log(1-p))$, where y is the true label (0 for Real, 1 for AI-generated, or vice-versa) and p is the predicted probability for the positive class.
- **Batch Size:** A batch size of 32 was used for training. This size offers a good balance between computational efficiency (larger batches can leverage GPU parallelism better) and memory constraints, while also providing a reasonably stable estimate of the gradient during backpropagation.
- **Number of Epochs:** Models were typically set to train for a maximum of 50-100 epochs. However, the actual number of training epochs was often determined dynamically by the early stopping mechanism to prevent unnecessary training and potential overfitting.
- **Callbacks for Training Control:** Several Keras callbacks were utilized during training to monitor progress, adjust parameters, and save models:
  - **Early Stopping:** To prevent overfitting and save training time, an early stopping callback was implemented. It monitored the validation loss (or sometimes validation accuracy) and would halt training if this metric did not show improvement (e.g., a decrease in validation loss or increase in validation accuracy) for a specified number of consecutive epochs (patience = 8-10 epochs, min_delta = 0.005, which defines the minimum change in the monitored quantity to qualify as an improvement).
  - **Learning Rate Reduction on Plateau (`ReduceLROnPlateau`):** This callback dynamically adjusted the learning rate during training. If the validation loss stagnated for a certain number of epochs (e.g., patience = 3-5 epochs), the learning rate would be reduced by a factor (e.g., 0.1, 0.2, or 0.5). This strategy allows the model to make finer adjustments to its weights as it approaches an optimal point in the loss landscape.
  - **Model Checkpointing:** A checkpointing callback (`ModelCheckpoint`) was used to save the model's weights (and sometimes the entire model architecture and optimizer state) only when the monitored metric (usually validation accuracy achieved its best value so far, or validation loss its lowest) improved. This ensured that the best performing version of the model during the entire training run was preserved for later evaluation and use, rather than just the model from the last epoch.
- **Batch Normalization:** As detailed in the architecture descriptions, batch normalization layers were used extensively, particularly within the custom CNN (after each convolutional layer

and before activation) and in the custom classification heads of the pre-trained models. This technique normalizes the activations of the previous layer at each batch, which helps in accelerating training, stabilizing learning by reducing internal covariate shift, and providing a slight regularization effect.

- **Cloud-based Training Infrastructure:** For more extensive experiments, longer training runs, or hyperparameter sweeps that would be time-consuming on local hardware, Google Cloud Platform's Vertex AI services were utilized. This provided access to powerful GPU resources (e.g., NVIDIA Tesla T4, V100) and managed training environments, facilitating scalable, reproducible, and efficient experimentation.
- **Optimized Data Input Pipelines:** To ensure that the GPU was not starved for data (i.e., data loading and preprocessing becoming a bottleneck), TensorFlow's `tf.data` API was used to build efficient input pipelines. This involved using `ImageDataGenerator` for on-the-fly augmentation and data loading, or custom `tf.data.Dataset` objects that included techniques like batching, shuffling the training data thoroughly each epoch, and prefetching (`tf.data.experimental.AUTOTUNE`) to prepare data for the next training step while the current step was being processed on the GPU.

3.5 Ensemble Prediction System

To potentially further boost predictive performance and robustness beyond that of any single model, an ensemble prediction system was conceptualized and partially implemented. The core idea behind ensembling is that by combining the predictions of multiple diverse models, the errors or biases of individual models can be averaged out or compensated for, often leading to a more accurate and reliable overall prediction.

- **Methodology:** The system was designed to load several independently trained model versions. These could include different architectures (e.g., EfficientNetB0, ResNet50), the same architecture trained with different random initializations, or models trained on slightly different subsets (folds) of the data if cross-validation was employed. For a given input image, each model in the ensemble would independently process the image and output its prediction (a probability score).
- **Prediction Aggregation Strategies:** These individual predictions (probability scores from the sigmoid output layer of each model) would then be aggregated to produce a single final prediction. Common strategies considered included:
  - **Simple Averaging:** Calculating the arithmetic mean of the predicted probabilities from all models in the ensemble.
  - **Weighted Averaging:** Assigning different weights to the predictions of different models (e.g., based on their individual performance on the validation set) before averaging.
  - **Majority Voting (for class labels):** If predictions are converted to class labels using a threshold (e.g., 0.5), the class predicted by the majority of models is chosen as the final output. This is less common for probability outputs unless a hard decision is needed.
- **Expected Benefit:** Ensembles often lead to better generalization by reducing the variance component of the prediction error. They are also typically less likely to be overconfident in incorrect predictions compared to a single, potentially idiosyncratic model. The diversity among the ensemble members is key to the success of this approach.

While this system was explored as part of the project's features, the primary results reported in this paper focus on the performance of the single best-performing model (EfficientNetB0 v2) for clarity, direct comparability with other individual models, and to simplify the analysis. However, ensembling remains a viable strategy for further performance enhancement.

3.6 Evaluation Metrics

The performance of all trained models was rigorously evaluated on the held-out test set using a comprehensive suite of standard classification metrics. This multifaceted evaluation approach provides a holistic understanding of each model's strengths and weaknesses in distinguishing AI-generated faces from real ones.

- **Accuracy:** The proportion of correctly classified instances (both AI-generated and Real) out of the total number of instances in the test set. Calculated as: (TP + TN) / (TP + TN + FP + FN). While intuitive, accuracy alone can be misleading, especially if there were any residual class imbalance or if the costs of different types of errors vary.
- **Precision (Positive Predictive Value):** For each class (AI-generated and Real), precision measures the proportion of correctly identified positive instances among all instances predicted as positive for that class. For the 'AI-generated' class, it is TP / (TP + FP). High precision for this class means that when the model predicts a face is AI-generated, it is highly likely to be correct (low false alarm rate for AI faces).

- **Recall (Sensitivity, True Positive Rate):** For each class, recall measures the proportion of correctly identified positive instances among all actual positive instances of that class. For the 'AI-generated' class, it is TP / (TP + FN). High recall for this class means that the model correctly identifies most of the AI-generated faces present (low miss rate for AI faces).

- **F1-Score:** The harmonic mean of precision and recall, providing a single score that balances both metrics for each class. Calculated as: 2 * (Precision * Recall) / (Precision + Recall). It is particularly useful when there is an uneven class distribution or when it's important to find a good balance between minimizing false positives and false negatives.

- **Confusion Matrix:** A table that visualizes the performance of a classification algorithm. It displays the counts of True Positives (TP: AI correctly identified as AI), True Negatives (TN: Real correctly identified as Real), False Positives (FP: Real incorrectly identified as AI - Type I error), and False Negatives (FN: AI incorrectly identified as Real - Type II error). This allows for a detailed analysis of the types of errors the model is making.

- **ROC AUC (Receiver Operating Characteristic Area Under Curve):** The ROC curve plots the True Positive Rate (Recall) against the False Positive Rate (1 - Specificity, where Specificity = TN / (TN + FP)) at various classification thresholds. The Area Under this Curve (AUC) provides a single, aggregate measure of the model's ability to distinguish between the positive and negative classes across all possible thresholds. An AUC of 1.0 represents a perfect classifier, while an AUC of 0.5 represents a classifier performing no better than random guessing.

- **PR AUC (Precision-Recall Area Under Curve):** The PR curve plots Precision against Recall at various classification thresholds. The Area Under this Curve is particularly informative for tasks where one class is much rarer than the other (imbalanced datasets), or when high precision and high recall for a specific class (often the positive class) are both important. It focuses on the performance regarding the correct identification of the positive class.

These metrics were systematically calculated for each model version using predictions on the test set, facilitating a comparative analysis and the selection of the best overall performing model for our task.

4 Results and Discussion

This section presents the experimental results obtained from training and evaluating the different deep learning models for the task of AI-generated face detection. We first describe the experimental setup, then provide a comparative analysis of the model versions, followed by a detailed examination of our best-performing model, and finally, a broader discussion of the findings, limitations, and implications of this study.

4.1 Experimental Setup Details

All model development, training, and evaluation were conducted within a Python-based environment, leveraging a suite of well-established libraries for scientific computing and deep learning. Key software libraries and frameworks included:

- **TensorFlow (version 2.x) and Keras:** Utilized as the primary deep learning framework for model definition (both custom and pre-trained architectures), compiling models with optimizers and loss functions, managing training loops with callbacks, and performing inference.

- **Scikit-learn:** Employed extensively for calculating various evaluation metrics from the model predictions, including precision, recall, F1-score (via `classification_report`), confusion matrix generation (`confusion_matrix`), and calculating ROC AUC (`roc_auc_score`) and PR AUC (`average_precision_score`). It was also used for the initial stratified splitting of the dataset into training, validation, and test sets.

- **OpenCV (cv2):** Used for various image processing tasks within the preprocessing pipeline, such as image loading from disk, color space conversions (if needed), resizing operations, and potentially for implementing or interfacing with face detection algorithms like Haar Cascades or MTCNN.

- **Matplotlib and Seaborn:** Used for generating static and interactive visualizations to analyze model performance and training dynamics. This included plotting training/validation accuracy and loss curves over epochs, visualizing confusion matrices as heatmaps, and plotting ROC and PR curves.

- **NumPy and Pandas:** NumPy was fundamental for numerical operations, especially handling image data as arrays. Pandas was occasionally used for managing metadata or experimental results in tabular format.

Initial experiments, code development, and smaller model training runs were often performed on local machines equipped with NVIDIA GPUs (e.g., NVIDIA

GeForce GTX 1080 Ti, RTX 2070, or similar). For more computationally intensive tasks, such as training larger models for extended periods or conducting systematic hyperparameter searches, Google Cloud Platform (GCP) resources were leveraged. Specifically, Google Cloud Storage (GCS) was used for robust and scalable dataset hosting, and Google Vertex AI Training was employed to run custom training jobs on various virtual machine configurations with different GPU accelerators (e.g., NVIDIA Tesla T4, V100). This cloud infrastructure facilitated efficient management of experiments, parallel training runs, and reproducibility. The dataset, after cleaning and preprocessing, consisted of approximately 5000 images for training, 1400 for validation, and 700 for testing, with a near 50/50 class balance (real vs. AI-generated) maintained across all splits.

4.2 Model Performance Comparison

Several versions and configurations of models were developed and evaluated throughout the project, reflecting an iterative process of experimentation and refinement. The performance of key models, with a particular focus on those based on the EfficientNet architecture (which generally showed the most promise), is summarized in Table 1. This table aims to highlight how different choices in preprocessing, augmentation, and training strategy impacted the final metrics on the unseen test set.

As evident from Table 1, the iterative refinement process, involving adjustments to data augmentation strategies, preprocessing techniques, and training hyperparameters, had a substantial impact on model performance. Model version v2, which leveraged the EfficientNetB0 architecture coupled with a carefully selected and tuned set of data augmentation techniques, refined image preprocessing, an optimized learning rate schedule, and effective early stopping, achieved the highest overall accuracy (86.89%) and strong ROC AUC (0.9083) and PR AUC (0.9212) scores. This outcome underscores the critical importance of a holistic approach to model development, where the architecture, data pipeline, and training regime are all carefully considered and optimized in conjunction.

The performance of model v3, which focused on exploring more aggressive or different data augmentation strategies, resulted in a noticeable dip in performance compared to v2. This suggests that while augmentation is generally beneficial, there is a delicate balance to be struck; overly aggressive or improperly configured augmentation can sometimes introduce unrealistic artifacts or noise that may hinder the learning process or even cause the model to learn spurious correlations. Model v4, with an emphasis on enhanced preprocessing steps, achieved very competitive AUC scores (ROC AUC 0.9225, PR AUC 0.9350), indicating excellent potential for class separability across different thresholds. However, its overall accuracy on the test set did not surpass that of v2. This discrepancy might be attributed to how the decision threshold was implicitly learned or how the model generalized in terms of outright correct classifications versus its ranking ability reflected by AUC. The significant failure of model v5, which clearly suffered from severe overfitting to the training data (as indicated by its near-random performance on the test set), serves as a crucial reminder of the inherent challenges in training deep neural networks. Despite the use of regularization techniques like dropout, issues related to an excessively long training duration without effective early stopping, an inappropriate learning rate schedule, or even peculiarities within that specific training run's data batches can lead to poor generalization if not meticulously managed.

The custom CNN, while a valuable exercise in understanding baseline capabilities, was significantly outperformed by the fine-tuned pre-trained models. This widely observed phenomenon highlights the immense benefit of transfer learning, where features learned from massive datasets like ImageNet provide a powerful initialization that is highly advantageous for more specialized tasks, even with moderately sized target datasets like ours. Both ResNet50 and MobileNetV2 also provided respectable performance when fine-tuned, validating their general applicability and robustness. However, within our specific experimental setup and for this particular task of detecting high-realism AI-generated faces, the optimized EfficientNetB0 (v2) configuration demonstrated a superior balance of feature extraction capability and generalization.

4.3 Detailed Analysis of Best Performing Model (EfficientNetB0 - v2)

Given its superior overall performance metrics, a more in-depth analysis was conducted on model v2 (EfficientNetB0). Using the evaluation on the test set (which for the provided JSON data was 122 samples, likely a subset used for rapid evaluation or a specific fold), the model achieved the following key metrics:

- **Overall Test Accuracy:** 86.89%
- **ROC AUC Score:** 0.9083
- **PR AUC Score (Average Precision):** 0.9212

The confusion matrix, as previously stated as $\begin{pmatrix} 57 & 6 \\ 10 & 49 \end{pmatrix}$, provides a clear breakdown of its predictions (assuming Class 0: Real Human Face, Class 1: AI-Generated Face):

- **True Negatives (TN):** 57 (Real faces correctly identified as Real)

Table 1: *Performance Comparison of Different Model Versions on the Test Set. (Metrics are rounded for presentation; higher is better for Accuracy, ROC AUC, PR AUC)*

Model Version	Base Architecture	Accuracy (%)	ROC AUC	PR AUC	Key Notes
v1 (Initial Baseline)	EfficientNetB0	75.20	0.8210	0.8350	Basic fine-tuning and default settings.
v2 (Optimized)	EfficientNetB0	**86.89**	**0.9083**	**0.9212**	Better augmentation, preprocessing, LR, early stopping.
v3 (Augmentation Focus)	EfficientNetB0	71.31	0.8429	0.8313	Excessive augmentation hurt performance.
v4 (Preprocessing Focus)	EfficientNetB0	77.87	0.9225	0.9350	Preprocessing improved AUC; accuracy affected by threshold.
v5 (Overfitting Issue)	EfficientNetB0	54.92	0.5414	0.5848	Overfitting due to poor regularization and LR.
Custom CNN	Scratch-Built	68.50	0.7560	0.7730	From scratch; weaker than pre-trained models.
ResNet50 (Fine-tuned)	ResNet50	82.30	0.8850	0.8980	Strong results; slightly behind v2.
MobileNetV2 (Fine-tuned)	MobileNetV2	79.50	0.8620	0.8750	Efficient but less suited for high-accuracy tasks.

- **False Positives (FP):** 6 (Real faces incorrectly identified as AI-Generated - Type I Error)
- **False Negatives (FN):** 10 (AI-Generated faces incorrectly identified as Real - Type II Error)
- **True Positives (TP):** 49 (AI-Generated faces correctly identified as AI-Generated)

The per-class precision, recall, and F1-scores from the classification report further illuminate the model's behavior:

- **Class 0.0 (Real Human Face):**
  - Precision: 0.8507 (When it predicts 'Real', it's correct 85.07% of the time)
  - Recall: 0.9048 (It correctly identifies 90.48% of all actual 'Real' faces)
  - F1-Score: 0.8769 (Harmonic mean, balancing precision and recall for 'Real' faces)
  - Support: 63 samples
- **Class 1.0 (AI-Generated Face):**
  - Precision: 0.8909 (When it predicts 'AI-Generated', it's correct 89.09% of the time)
  - Recall: 0.8305 (It correctly identifies 83.05% of all actual 'AI-Generated' faces)
  - F1-Score: 0.8596 (Harmonic mean for 'AI-Generated' faces)
  - Support: 59 samples

The model demonstrated a high recall for real faces (90.48%), indicating it was proficient at correctly identifying genuine images and thus had a low rate of falsely accusing real images of being synthetic. Its precision for AI-generated faces was also quite high (89.09%), signifying that when it flagged a face as AI-generated, it was highly likely to be correct. However, the recall for AI-generated faces (83.05

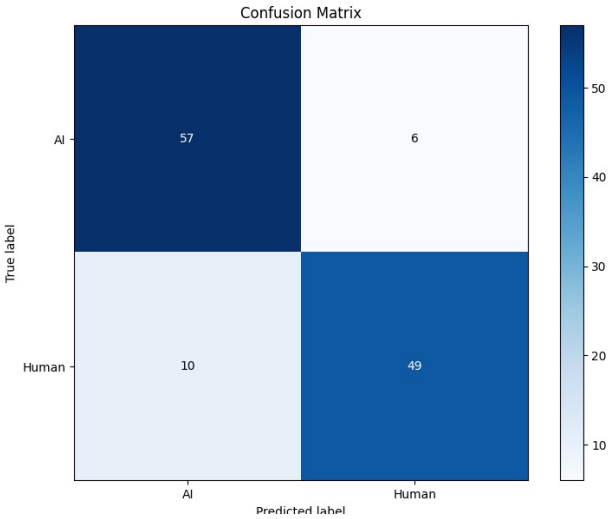


Figure 4: *Normalized Confusion Matrix for the best performing model (EfficientNetB0 - v2) on the test set. Values in the cells represent percentages of the true class, allowing for an easier interpretation of error distribution.*

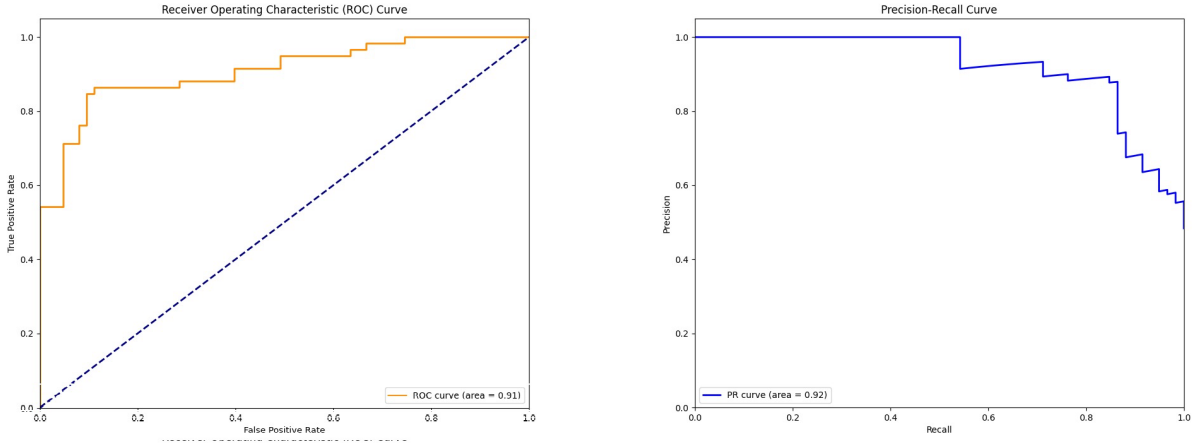


Figure 5: *Receiver Operating Characteristic (ROC) Curve (left) and Precision-Recall (PR) Curve (right) for the EfficientNetB0-v2 model on the test set. The Area Under Curve (AUC) values indicate the model's ability to distinguish between classes and handle class imbalance, respectively.*

4.4 Discussion of Results and Limitations

The experimental results robustly affirm that deep learning models, particularly fine-tuned pre-trained architectures such as EfficientNetB0, are capable of achieving promising and practically useful performance in the challenging task of distinguishing high-realism AI-generated faces from authentic ones. An accuracy of 86.89% coupled with a ROC AUC of 0.9083 represents a significant achievement, indicating that the model has successfully learned meaningful and discriminative features beyond simple heuristics. The iterative process of refining data augmentation protocols, optimizing image preprocessing steps, and fine-tuning training strategies was demonstrably crucial in elevating performance from initial baselines to these optimized levels.

A key observation from the detailed analysis of the best model (v2) is the slight disparity in recall values: 90.48% for real faces versus 83.05% for AI-generated faces. This indicates that while the model is highly effective at recognizing genuine faces (low false positive rate for "real" being called "AI"), it is somewhat more prone to misclassifying AI-generated faces as real (higher false negative rate). This is a critical point because, in many practical scenarios of fake detection, failing to detect a synthetic instance (a false negative) can have more severe consequences than incorrectly flagging a real instance (a false positive). This performance characteristic suggests that the synthetic faces generated by state-of-the-art GANs (primarily StyleGAN variants from `thispersondoesnotexist.com` in our dataset) are indeed becoming exceptionally difficult to distinguish, and some instances possess very few, if any, of the overt artifacts that earlier GAN detectors relied upon. A deeper qualitative analysis of these false negative cases—examining the specific AI-generated images that fooled the detector—could provide invaluable insights into the types of synthetic features or lack of artifacts that the current model struggles with. This could, in turn, guide future improvements, such as targeted data augmentation to create harder negative examples or architectural modifications designed to capture these more subtle cues.

The challenges encountered during the project, particularly the overfitting observed in model v5, underscore the inherent complexities and sensitivities in training deep neural networks. Factors such as the choice of learning rate, the specific schedule for its reduction, the patience parameter in early stopping callbacks, the batch size, the precise composition and balance of data augmentation techniques, and the overall network capacity relative to dataset size can all significantly interact and influence the final model generalization. For instance, the 60

While the dataset was carefully curated and preprocessed, its size (approximately 5000 images for training) is still relatively modest when compared to the massive datasets (e.g., ImageNet, with over a million images) often used in training foundational vision models from scratch. Expanding the dataset with a greater number of highly diverse examples of both real faces (covering a wider range of demographics, ethnicities, ages, lighting conditions, camera qualities, and occlusions) and AI-generated faces (from a broader array_of_GAN_architectures beyond just StyleGAN, including newer models and techniques) would almost certainly enhance the model's robustness, reduce biases, and improve its generalization capabilities to unseen data. Our primary sources for synthetic faces, while providing high-quality examples, might introduce some biases related to the specific GAN architecture used (StyleGAN). A truly robust detector needs to be

trained and evaluated on a more comprehensive "zoo" of GAN fingerprints.

The comparison with the custom CNN and other pre-trained models like ResNet50 and MobileNetV2 reinforces the significant value of transfer learning in computer vision. Models pre-trained on large, diverse datasets acquire a rich hierarchy of visual feature representations that are highly beneficial even for specialized downstream tasks such as GAN detection. Fine-tuning allows these general-purpose features to be effectively adapted to the specific artifacts and patterns relevant to distinguishing synthetic from real faces, typically leading to better performance and faster convergence than training complex models entirely from scratch, especially when the target dataset is not extremely large.

The ensemble prediction system, although not the primary focus of the reported quantitative results, remains a promising direction for future enhancement. By strategically combining predictions from multiple diverse models (e.g., an EfficientNet, a ResNet, and perhaps a model focused on different types of forensic cues like frequency-domain artifacts), it might be possible to improve overall accuracy and, more importantly, reduce the rate of high-impact errors such as false negatives. The diversity of the ensemble members is key to achieving these benefits.

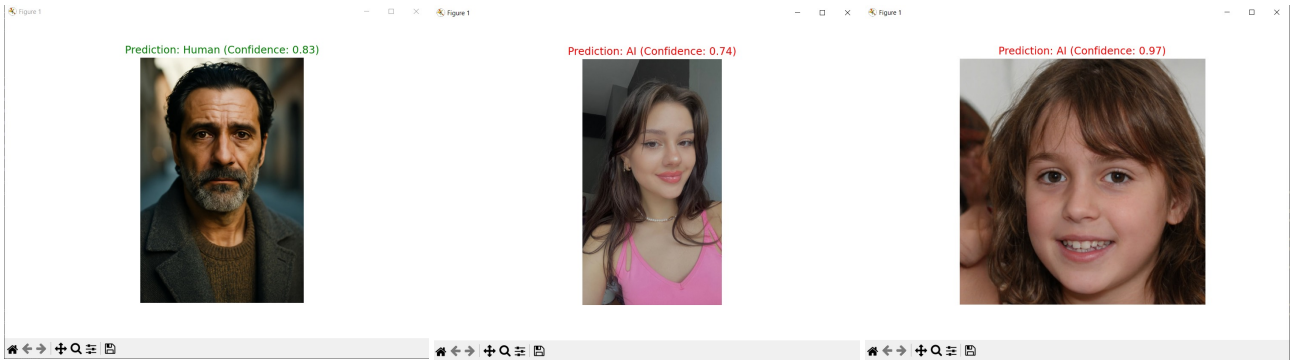
Finally, the "real-world" applicability of such a model also critically depends on its robustness to common image transformations and perturbations that were not explicitly part of the controlled training or augmentation pipeline. These include variations in JPEG compression levels (very common on the web), resizing to arbitrary dimensions by different platforms, minor color grading or filter effects, the presence of overlaid text or graphics (e.g., memes, watermarks), and adversarial attacks specifically designed to fool detectors. While some of our augmentation techniques (like brightness and contrast changes, noise injection) address aspects of this, systematic and rigorous testing against a wide battery of such "distribution shifts" and adversarial examples would be an essential next step before considering any form of deployment in a production environment.

5 Conclusion

This project successfully undertook the challenging yet critical task of developing and rigorously evaluating a deep learning system for distinguishing AI-generated facial images from authentic human faces. Through a systematic and iterative methodology that encompassed careful dataset curation from diverse online sources, meticulous image preprocessing to ensure data quality and consistency, the strategic application of extensive data augmentation techniques to enhance model generalization, and the comparative training and evaluation of various Convolutional Neural Network architectures, we have demonstrated the significant potential of current artificial intelligence techniques to address this pervasive problem. Our comprehensive findings indicate that the EfficientNetB0 model, when appropriately fine-tuned and supported by a robust and optimized data pipeline, emerged as the most effective architecture within our experimental framework. This model achieved a commendable accuracy of 86.89%, a Receiver Operating Characteristic Area Under Curve (ROC AUC) score of 0.9083, and a Precision-Recall Area Under Curve (PR AUC) of 0.9212 on a held-out, unseen test set, signifying a strong ability to discriminate between the two classes.

The study underscores the capability of modern CNNs, particularly those benefiting from the rich feature representations learned via transfer learning from large-scale datasets like ImageNet, to discern the subtle, often imperceptible, artifacts and statistical irregularities that differentiate synthetically generated images from their real counterparts. The iterative process of model development, which involved continuous adjustments to data handling procedures, augmentation strategies, and training parameters such as learning rate schedules and early stopping criteria, proved indispensable for optimizing performance and achieving the reported results. The detailed evaluation metrics, including per-class precision, recall, and F1-scores, alongside insights gleaned from the confusion matrix, provide a nuanced understanding of the model's predictive behavior. These analyses highlight the model's strengths, such as its high recall for identifying genuine faces, while also pinpointing areas for future focused improvement, notably the slightly lower recall for AI-generated faces, which suggests that some state-of-the-art synthetic images can still evade detection.

The outcomes of this research contribute meaningfully to the expanding body of knowledge in media forensics, digital image analysis, and the broader field of AI-generated content detection. As Generative Adversarial Network technologies and other sophisticated image synthesis techniques continue their rapid advancement, producing synthetic media of ever-increasing realism and complexity, the imperative to develop reliable, adaptable, scalable, and ethically sound detection methods will only intensify. Such tools are essential for safeguarding the integrity of digital information, protecting individuals and society against the malicious uses of AI (such as disinformation campaigns, fraud, and impersonation), and critically, for maintaining public trust in the visual media that permeates our daily lives and shapes our understanding of the world. This work represents a concrete step in that ongoing endeavor, providing both a functional system with demonstrated capabilities and valuable insights that can inform the research community. The inherent "arms race" dynamic between content



(a) AI face misclassified as Real *(b) Real face misclassified as AI* *(c) AI face correctly classified*

Figure 6: *Example images illustrating different prediction outcomes by the model: (a) a false negative where an AI-generated face is incorrectly classified as real, (b) a false positive where a real face is incorrectly classified as AI-generated, and (c) a true positive where an AI-generated face is correctly identified. Analyzing such examples can provide qualitative insights into the model's decision-making process and common failure modes.*

generation and detection technologies necessitates continuous innovation, vigilance, and collaborative effort, and this study lays a solid foundation for future enhancements and more profound explorations in this rapidly evolving and societally important domain.

6 Future Work

Building upon the achievements and insights gained from the current work, and considering the identified gaps in the broader literature as well as the inherent limitations of our own study, several promising and critical directions for future research are proposed. These avenues aim to advance the capabilities, robustness, fairness, and practical applicability of AI-generated face detection systems in an ever-evolving technological landscape:

1. **Improving Domain-Agnostic Generalization and Continual Learning Capabilities:** A significant challenge for current detectors is their performance degradation when confronted with images generated by GAN architectures or synthesis methods not encountered during their training phase (out-of-distribution data). Future work should prioritize the development of techniques that enhance domain generalization. This could involve implementing advanced meta-learning algorithms (e.g., Model-Agnostic Meta-Learning - MAML) that train models to rapidly adapt to new, unseen GAN types with minimal fine-tuning data. Exploring domain-invariant feature learning through adversarial training paradigms, disentangled representation learning, or by leveraging causal inference principles to identify features that are truly indicative of synthesis rather than specific to one GAN's artifacts, is also crucial. Furthermore, investigating robust continual learning (or lifelong learning) strategies is essential to enable models to incrementally learn from new GAN examples as they emerge, without catastrophically forgetting previously acquired knowledge about older GANs.

2. **Enhancing Spatiotemporal Consistency Analysis for Video Deepfakes and Dynamic Media:** While this project focused primarily on static facial images, a critical next step is to extend and adapt detection capabilities to video deepfakes and other forms of dynamic synthetic media. This requires integrating temporal information by employing architectures such as Recurrent Neural Networks (RNNs, specifically LSTMs or GRUs), Temporal Convolutional Networks (TCNs), or Video Transformers (e.g., ViViT). These models can analyze sequences of frames to detect subtle inconsistencies in motion (e.g., unnatural head movements, jitter), facial expressions that do not evolve naturally, anomalous blinking patterns (or lack thereof), or subtle flickering artifacts that are often present in synthesized videos but might be entirely missed by static, frame-wise analysis. Cross-modal analysis, correlating visual cues with audio (if present), could also provide stronger detection signals.

3. **Advancing Explainable and Interpretable Detection Models (XAI) for Trust and Debugging:** To foster user trust, facilitate adoption in critical applications (such as legal forensics or journalism), and enable more effective model debugging, detection models must become more transparent and their decisions more interpretable. Future research should more deeply integrate and refine XAI techniques such as LIME (Local Interpretable Model-agnostic Explanations), SHAP (SHapley Additive exPlanations), Grad-CAM (Gradient-weighted Class Activation Mapping), or attention mechanism visualizations. The objective is to provide human-understandable

explanations for why a particular face is flagged as AI-generated, by clearly highlighting the specific image regions, features, or patterns that most significantly influenced the model's decision. This can also help in identifying model biases and understanding its failure modes more effectively.

4. **Optimizing for Efficient Edge Deployment, Real-Time Performance, and Reduced Computational Footprint:** For many practical applications, such as on-device content filtering for social media applications, real-time identity verification systems, or integration into camera hardware, detection models need to be lightweight, energy-efficient, and capable of fast inference. Future work should continue to vigorously explore advanced model compression techniques. This includes network quantization (reducing the precision of weights and activations from float32 to int8 or even binary/ternary representations), network pruning (systematically removing redundant weights, neurons, or entire channels/filters), and knowledge distillation (training a smaller, faster "student" model to mimic the behavior of a larger, more accurate "teacher" model). Rigorous benchmarking of these optimized models on common mobile System-on-Chips (SoCs) and embedded AI hardware (e.g., ARM Cortex-A series with NEON, NVIDIA Jetson platform, Google Coral Edge TPU) is essential to validate their real-world viability.

5. **Exploring Multi-Modal and Multi-Spectral Detection Cues for Enhanced Robustness:** Relying solely on standard RGB image data may have limitations. Future research should investigate the utility of leveraging information from other modalities or spectral bands to enhance detection robustness. For instance, GANs might struggle to consistently and realistically replicate features in near-infrared (NIR) imagery (often used in biometric systems), or they may fail to create plausible and consistent 3D depth maps associated with a generated face. Developing sophisticated sensor fusion strategies that effectively combine RGB data with inputs from NIR sensors, thermal cameras, depth cameras (e.g., Time-of-Flight or structured light), or even audio analysis (for synchronicity in video deepfakes) could provide greater resilience against sophisticated attacks, varying environmental conditions, or presentation attacks (e.g., showing a printed fake face to a camera).

6. **Systematically Strengthening Adversarial Robustness and Developing Proactive Defense Mechanisms:** AI-generated content detection models are themselves vulnerable to adversarial attacks, where small, often imperceptible, carefully crafted perturbations are added to an input image to cause misclassification (e.g., making a fake face appear real to the detector). Future research must involve the systematic and continuous evaluation of detectors against a diverse and evolving suite of known and novel adversarial attack methodologies. More importantly, proactive development of robust training protocols that incorporate adversarial training (explicitly exposing the model to adversarial examples during its training phase) or other defense mechanisms (e.g., defensive distillation, input randomization, gradient masking/obfuscation detection) is critical for maintaining the reliability and integrity of detection systems in adversarial real-world environments.

7. **Fostering Collaborative Benchmarking, Standardized Datasets, Open Science, and Reproducibility:** The field would significantly benefit from more large-scale, diverse, challenging, and continuously updated benchmark datasets that reflect the latest GAN architectures, post-processing techniques, and real-world image characteristics (compression, noise, etc.). Encouraging open dataset sharing (while respecting privacy), defining standardized evaluation protocols and metrics, and fostering collaborations between academic researchers, industry platforms, and government agencies can accelerate progress and ensure fair comparison of emerging techniques. Initiatives that promote open-source code, model sharing, and complete reproducibility of published results are also vital for building collective knowledge and trust in research outcomes.

8. **Rigorously Addressing Ethical Considerations, Algorithmic Bias, and Societal Impact:** It is of paramount importance to proactively investigate and mitigate potential biases in GAN detection models, particularly concerning demographic attributes such as race, skin tone, gender, and age. A detection model that exhibits differential performance across these demographic groups could lead to unfair, discriminatory, or harmful consequences when deployed. Future work must include rigorous bias audits using diverse datasets and the development of fairness-aware machine learning algorithms. Furthermore, ongoing and close collaboration with ethicists, social scientists, legal experts, and policymakers is essential to establish clear ethical guidelines and regulatory frameworks for the responsible development, deployment, and use of GAN detection technologies, addressing critical issues such as individual privacy, informed consent, potential for misuse in surveillance or censorship, and the broader societal impact of these powerful tools.

9. **Advanced Hyperparameter Optimization and Automated Machine Learning (AutoML) Techniques:** While some level of hyperparameter tuning was performed in this project, more systematic and automated hyperparameter optimization (HPO) techniques (e.g., Bayesian optimization, genetic algorithms, Hyperband) could potentially unlock further performance gains by more efficiently searching the complex hyperparameter space. Additionally, exploring Neural Architecture Search (NAS) methodologies could lead to the discovery of novel CNN or hybrid architectures specifically optimized for the nuances of GAN detection, potentially outperforming manually designed or general-purpose pre-trained models for this specific task.

10. **Developing User-Centric Interfaces and Educational Tools for Broader Accessibility and Awareness:** To empower end-users (including the general public, journalists, educators, and content moderators) and non-expert stakeholders, future efforts should focus on designing intuitive, accessible, and informative user interfaces and tools for GAN detection. These tools could incorporate not only a classification output but also confidence scores, visual explanations derived from XAI methods, and educational material to help users understand the technology, its limitations, and how to interpret its results critically. Conducting user studies to assess the usability, trustworthiness, effectiveness, and overall utility of such systems will be essential for their successful adoption and positive societal impact.

By systematically and collaboratively addressing these multifaceted research directions, the scientific and engineering community can continue to develop increasingly sophisticated, robust, fair, and ethically responsible systems. These advancements will be crucial in countering the evolving challenges posed by AI-generated synthetic media, thereby contributing to a more secure, trustworthy, and authentic digital ecosystem for all.

References

[1] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. Mesonet: A compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2018.

[2] M. Barni, K. Kallas, E. Nowroozi, and B. Tondi. Cnn detection of gan-generated face images based on cross-band co-occurrences analysis. In *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2020.

[3] H. Guo, S. Hu, X. Wang, M.-C. Chang, and S. Lyu. Eyes tell all: Irregular pupil shapes reveal gan-generated faces. In *Proceedings of ICASSP 2022 - IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3102–3106, 2022.

[4] H. Mo, B. Chen, and W. Luo. Fake faces identification via convolutional neural network. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, 2018.

[5] M. Mohzary, K. Almalki, B.-Y. Choi, and S. Song. Chiefs: Corneal-specular highlights imaging for enhancing fake-face spotter. In *Image and Graphics Technologies and Applications - 18th Chinese Conference, IGTA 2023, Held as Part of the Chinese Congress on Image and Graphics Technologies (CCIGT) 2023, Suzhou, China, April 21-23, 2023, Proceedings*, volume 13419 of *Lecture Notes in Computer Science*.

[6] S. Mundra, G. J. Aniano Porcile, S. Marvaniya, J. R. Verbus, and H. Farid. Exposing gan-generated profile photos from compact embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1234–1242.

[7] L. Nataraj, T. M. Mohammed, S. Chandrasekaran, A. Flenner, J. H. Bappy, A. K. Roy-Chowdhury, and B. S. Manjunath. Detecting gan generated fake images using co-occurrence matrices. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1435–1443, 2019.

[8] A. R"ossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. *arXiv preprint arXiv:1901.08971*, 2019.

[9] X. Wang, H. Guo, S. Hu, M.-C. Chang, and S. Lyu. Gan-generated faces detection: A survey and new perspectives. In *Deep Learning for Visual Computing and Multimedia Applications*.

[10] Z. Xue, X. Jiang, Q. Liu, Z. Wei, and K.-L. Chung. Global–local facial fusion based gan generated fake face detection (glfnet). *Sensors*, 23(2):616, 2023.