A Deep Learning Approach for Face Detection and Deepfake Identification

Abstract

Deepfake media presents a critical threat to digital authenticity, personal identity, and public trust. This project proposes a two-stage Convolutional Neural Network (CNN) pipeline that robustly identifies deepfakes by first detecting whether an image contains a human face and then classifying the authenticity of that face. The first model is a lightweight CNN trained for face vs. non-face classification, ensuring only relevant regions are passed to the next stage. The second model uses a state-of-the-art EfficientNet-B0 backbone with Squeeze-and-Excitation (SE) attention blocks to determine if the detected face is real or AI-generated. This hierarchical approach mimics how real-world systems should operate and significantly improves detection accuracy and reliability. The project showcases how intelligent system design and model integration can strengthen security applications in the era of synthetic media.

1. Introduction

1.1. Why is Face Detection Important?

Face detection plays a foundational role in many computer vision applications such as facial recognition, emotion analysis, identity verification, human-computer interaction, and surveillance. It is often the first critical step in a broader pipeline that enables machines to interpret and respond to human presence. In the context of this project, face detection ensures that the deepfake classifier is only applied to relevant image regions, thereby improving efficiency and accuracy. Without reliable face detection, systems would waste computational resources on non-relevant content and risk misclassifying images that do not contain faces. Moreover, in security-focused applications, face detection helps in real-time identification and access control, making it essential for trustworthy AI systems.

1.2. Why is Deepfake Detection Important?

With the rapid advancement of artificial intelligence, deepfake technology has emerged as a major cybersecurity threat. Deepfakes, generated using Generative Adversarial Networks (GANs) and autoencoders, can manipulate human faces and voices in videos, making it difficult to distinguish real from fake. These forged media can be used for misinformation, fraud, identity theft, and political manipulation, posing serious ethical and security concerns. As deepfake algorithms improve, there is an urgent need for robust detection methods to safeguard digital integrity.

1.3. Why is Our Approach Novel?

Our proposed system simulates a real-world multi-stage security filter. It first ensures that the content being evaluated is indeed a facial image, preventing unnecessary and costly computations on irrelevant images. Then, it applies advanced deep learning techniques to detect signs of manipulation. This separation into two phases allows:

Reduced false positives by verifying input context (face vs. non-face).

Improved deepfake detection via SE-enhanced EfficientNet.

Modular deployment in systems where face validation is already a step (e.g., facial recognition in surveillance).

By combining classic CNN techniques and recent advancements like SE blocks and transfer learning, this system balances efficiency, accuracy, and robustness.

1.4. Mathematics

- **Binary Cross-Entropy Loss (Used in Both Models):**

$$L = -\sum_{i=1}^{N} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right] \quad (1)$$

  This loss function is used for binary classification problems to penalize incorrect predictions.

- **Channel Attention in Squeeze-and-Excitation (SE) Block:**

$$s = \sigma(W_2(\text{ReLU}(W_1 z))) \quad (2)$$

  Where z is the globally pooled feature vector, W_1 and W_2 are the weights of the fully connected layers, and σ is the sigmoid activation function.

- **Accuracy:**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

Measures the overall correctness of the model's predictions.

- **Precision:**

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

Reflects how many predicted positives were actually correct.

- **Recall:**

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

Measures the ability to find all relevant instances (true positives).

- **F1-Score:**

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

Harmonic mean of precision and recall, balancing both metrics.

2. Proposed Method

2.1. Overview

This project implements a two-stage system for robust deepfake detection:

Face Detection Phase — Identifies whether an image contains a face.

Deepfake Classification Phase — Determines whether the detected face is real or AI-generated.

This separation of tasks reduces computational complexity in the later stage and improves the robustness of deepfake classification by removing irrelevant parts of the image.

2.2. Face Detection Network

The face detection model is a compact, custom Convolutional Neural Network designed for binary classification. It distinguishes between face and non-face patches using a combination of convolutional blocks and fully connected layers.

Dataset: Composed of labeled face and non-face image directories.

Input Processing:

Images resized to 48×48 pixels

Data augmentation: horizontal flipping, rotation for generalization

Architecture:

2–3 convolutional layers with ReLU activations

Max pooling for downsampling

Final fully connected classifier with sigmoid activation

Training Setup:

Binary Cross-Entropy loss

Batch size: 32, Epochs: 50

Optimizer: Adam

Output: Classifies input as either face (1) or non-face (0)

2.3. Deepfake Classification Network

Once faces are isolated, they are passed into a highly expressive model built on EfficientNet-B0:

Base Model: EfficientNet-B0 (pretrained on ImageNet)
Enhancement: Squeeze-and-Excitation (SE) block added to improve channel-wise attention
Classifier: Global Average Pooling → Fully Connected Layer (2 output classes: Real/Fake)
Input Size: 256×256

Training Details:
• Fine-tuned the last five layers of EfficientNet-B0
• Optimizer: Adam with learning rate of 0.0001
• Loss Function: CrossEntropyLoss (suitable for multi-class binary classification)

The model outputs class logits for Real and Fake, and applies softmax during evaluation to compute probabilities. Classification is based on the higher probability class.

3. Experimental Analysis

To validate our method, we conducted experiments on a publicly available deepfake and face detection dataset. This section discusses dataset details, evaluation metrics, and experimental results.

3.1. Dataset

For face detection we used a customised dataset containing faces and non faces which were taken from other datasets.
For deepfake detection :
https://www.kaggle.com/datasets/manjilkarki/deepfake-and-real-images

3.2. Evaluation Metrics

Both face detection and classification components were assessed using:
Accuracy
Precision
Recall
F1-score
Confusion Matrix (visualized using Seaborn)
These metrics provide insights into how well the model distinguishes between real and fake faces and whether it maintains a good balance between false positives and false negatives.

3.3. Results

Task	Metric	Value
Face Detection	Test Accuracy	0.9008
	Confusion Matrix	$\begin{bmatrix}1263 & 241\\ 69 & 1553\end{bmatrix}$
Deep Fake Detection	Accuracy	0.98
	Precision	0.98
	Recall	0.98
	F1-score	0.98
	ROC AUC Score	0.9977
	IoU (Jaccard) Score	0.9577

Table 1. Summary of Model Performance for Face Detection and Deep Fake Detection

These results indicate the model is highly effective at distinguishing real from fake faces, especially with a strong balance between precision and recall.

3.4. Visualisaton

The performance of both face detection and deepfake detection models is further evaluated using confusion matrices and training loss graphs. These visualizations highlight the model's classification capabilities and convergence during training.

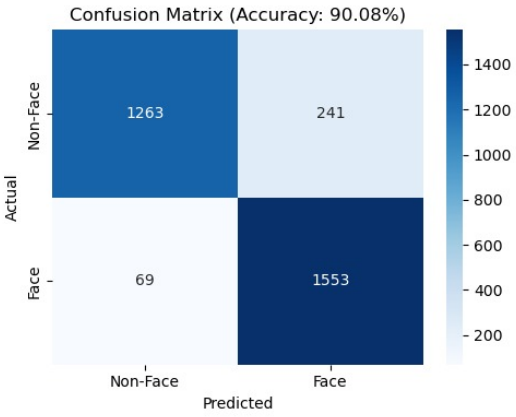

Figure 1. Confusion matrix of face detection.

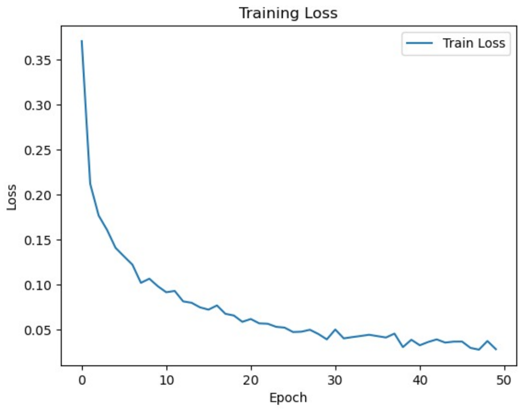

Figure 2. Graph of loss in face detection.

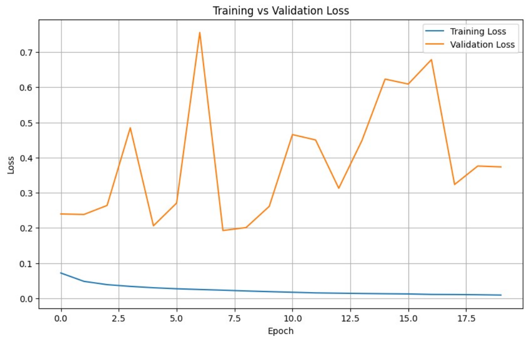

Figure 3. Graph of loss in Deep fake.

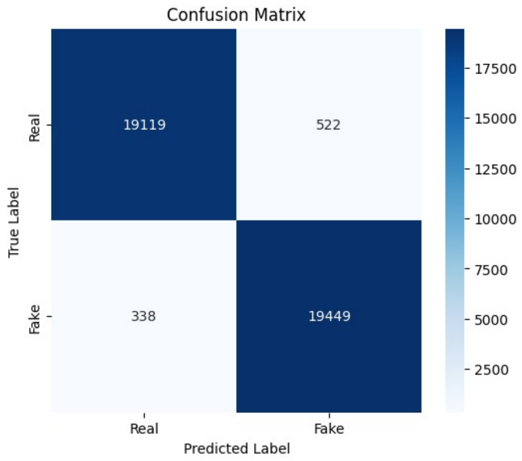

Figure 4. Confusion matrix of Deep fake.

4. Code and Dataset Links

For face detection, we used a customised dataset containing faces and non-faces which were taken from other datasets.

For deepfake detection:
Deepfake and Real Images Dataset

Code link :
Full code link

5. References

1. Face Detection using CNNs:
   Zhang, K., Zhang, Z., Li, Z., Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10), 1499–1503.
   Link to Paper

2. EfficientNet (if used or referenced):
   Tan, M., Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning* (pp. 6105–6114).
   Link to Paper

3. Evaluation Metrics:
   Sokolova, M., Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427–437.
   Link to Paper