Deep-Fake Video Detection Using CNNs and Transfer Learning

Dorukhan YILDIZ
Student ID: 202011003
Department of Computer Engineering
Çankaya University

Hikmet Berkin BULUT
Student ID: 202111057
Department of Computer Engineering
Çankaya University

***Abstract*—This paper presents a deepfake video detection pipeline based on CNNs and transfer learning, leveraging pre-processed face frames available in a subset of the Deepfake Detection Challenge (DFDC) dataset. EfficientNet-B0 was used as the base architecture, trained on face-cropped video frames. The system achieved an AUC-ROC of 0.82 with moderate computational requirements. Ethical considerations, limitations, and avenues for future work are also discussed.**

I. Introduction

Deepfake technology enables the creation of highly realistic synthetic videos, which can undermine trust in digital media. Detection systems based on deep neural networks offer promising results, especially when fine-tuned on domain-specific datasets. This work presents a CNN-based approach trained on a subset of DFDC data, optimized for limited computational resources.

We used a Kaggle dataset curated specifically for the DFDC challenge [1], in which each video had 5–10 pre-extracted face frames. This greatly reduced preprocessing time and allowed us to focus on model design and evaluation.

Using a fixed seed value (`random_seed = 42`), we ensured the reproducibility of experiments, including frame selection and train-validation-test splits.

The subset statistics were as follows:

- Total video entries: 2,658
- REAL videos: 416
- FAKE videos: 2,242

This alignment between fakes and their sources enabled detailed pairwise comparisons and consistent training conditions across runs.

II. Model Architecture and Hyperparameters

We employed EfficientNet-B0 as the backbone, pre-trained on ImageNet. The network head consists of a global average pooling layer, followed by a dense layer with 128 units, a 0.5 dropout, and a final sigmoid layer for binary classification.

Details:

- Input Size: 224x224 RGB
- Optimizer: Adam (learning rate = 1e-4)
- Loss: Binary Cross-Entropy
- Augmentations: Resize, ToTensor
- Regularization: Dropout (0.5)

The model was trained with the backbone frozen.

III. Learning Curves

Training progress was monitored using TensorBoard and stored logs. The model showed consistent improvements in accuracy and loss during the early epochs. To provide a detailed view of training dynamics, we include the following visualizations.

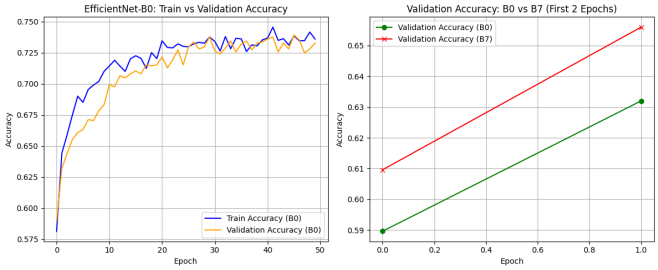

Fig. 1. Left: EfficientNet-B0 training vs validation accuracy across 50 epochs. Right: Comparison of B0 and B7 on the first 2 validation epochs.

As seen in the left panel, EfficientNet-B0 demonstrates stable convergence and generalization across epochs. In the right panel, EfficientNet-B7—despite running only for 2 epochs due to its complexity—achieves better validation accuracy compared to B0 at the same point, indicating potential if further training were feasible.

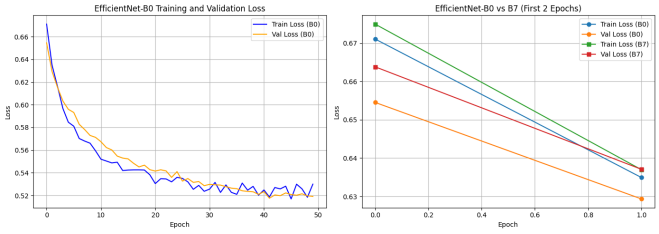

Fig. 2. Left: EfficientNet-B0 training vs validation loss. Right: Comparison of B0 and B7 loss trends for the first 2 epochs.

Both B0 and B7 show rapid early loss reduction. B7 starts with lower validation and training loss, but due to its size (around 66M parameters), it was not feasible to train it fully within time and hardware constraints. Still, its promising initial trend encourages future exploration.

IV. Evaluation

The model was evaluated on a held-out validation set using AUC-ROC, accuracy, and F1-score. The best model was saved at epoch 46 out of 50, based on validation performance.

- Validation Accuracy: 73.92%
- F1 Score: 0.7631
- AUC-ROC: 0.8212

At epoch 46, training loss was 0.5170 with an accuracy of 73.85%. Validation loss at the same epoch was 0.5208, indicating stable generalization performance.

We also attempted to train EfficientNet-B7 for comparison. Despite only completing 2 epochs due to computational constraints, it showed a promising start with a validation AUC-ROC of 0.70 and validation accuracy of 65.6%. Compared to EfficientNet-B0's early performance, B7 appeared to converge faster, but its higher parameter count significantly increased training time (7+ minutes per epoch), making it less feasible for extensive training under our hardware limitations.

False positives often involved low-light or blurry real faces, while false negatives came from high-quality fakes.

A. Error Analysis

To better understand the model's misclassifications, we manually examined typical examples of false positives and false negatives.

Fig. 3. False Positives: REAL faces predicted as FAKE.

Figure 3 illustrates several false positive cases. Most of these frames suffer from compression artifacts, low-light conditions, or facial blurriness. These issues may lead the model to detect non-existent manipulation cues and wrongly classify genuine content as fake.

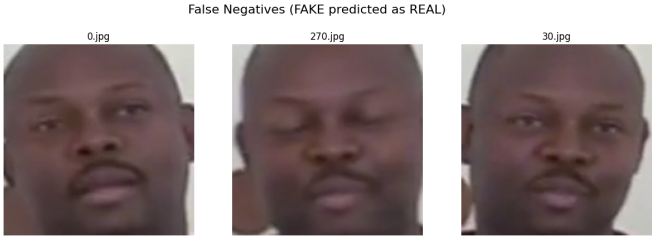

Fig. 4. False Negatives: FAKE faces predicted as REAL.

In contrast, Figure 4 presents examples of false negatives. These fake frames are visually clean and exhibit few signs of manipulation, such as flickering or distortions. In some cases, closed eyes or frontal lighting helped the fakes pass as real, highlighting the model's limitations in subtle forgery detection.

These findings suggest that real-world robustness requires not just spatial-level feature modeling, but possibly temporal and motion-aware analysis to distinguish authentic vs manipulated content more reliably.

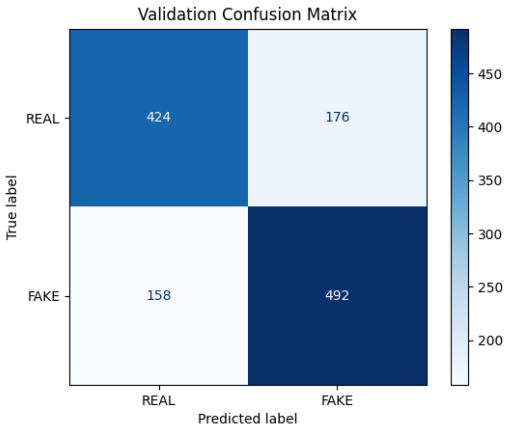

Fig. 5. Confusion matrix. (At epoch=49. You can view logs for more.)

V. Ethical Discussion and Future Work

A. Ethical Impact

Automated deepfake detection tools can protect individuals and institutions from misinformation. However, their misuse or failure may result in false accusations or discrimination. Ethical deployment requires human oversight and fairness audits.

Additionally, the public availability of detection systems may enable malicious actors to reverse-engineer or adapt their generation techniques to evade detection, further escalating the arms race between fake generation and detection. This could ultimately make synthetic content harder to identify and increase societal harm if not properly regulated.

B. Limitations

- Ignores temporal consistency
- Limited generalization to new deepfake methods
- Dataset bias toward certain demographics

C. Future Work

- Explore more advanced backbones beyond EfficientNet-B0, such as EfficientNet-B4/B7 or other transformer-based architectures.
- Train heavier EfficientNet variants with longer epochs if computational resources permit.
- Integrate 3D CNNs for capturing temporal dynamics (e.g., X3D).
- Use self-supervised pretraining for better feature representations.
- Apply adversarial training techniques to improve model robustness.

- Evaluate generalizability on cross-domain and real-world datasets.

VI. Conclusion

This project demonstrates the feasibility of detecting deepfakes using transfer learning on pre-extracted face frames from the DFDC dataset. We employed a lightweight EfficientNet-B0 architecture to achieve high validation performance under limited resources. Preliminary tests with EfficientNet-B7 indicate that more powerful models may offer better convergence, though at significantly higher computational cost. Future work should explore temporal modeling and scale up model architectures to improve generalization and robustness in more realistic settings.

References

[1] K. U. greatgamedota, "Dfdc part 34," https://www.kaggle.com/datasets/greatgamedota/dfdc-part-34, 2020, accessed: 2025-05-01.