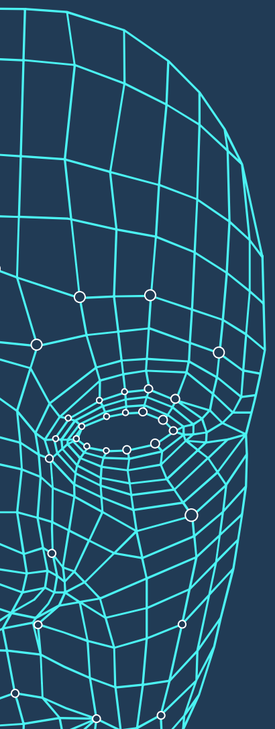
جامعة الملك سعود
King Saud University

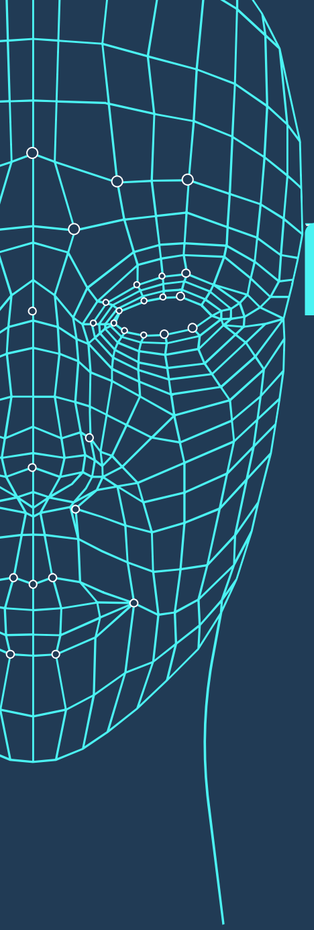
DEEPDETECT

IT461-2025



01

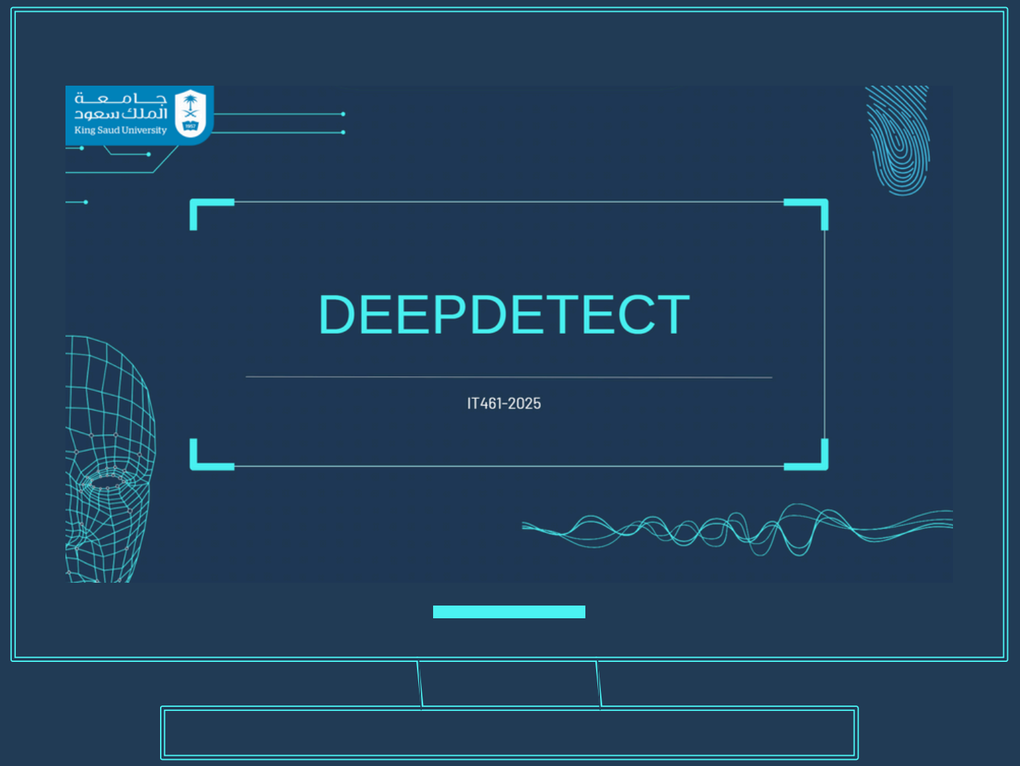
INTRODUCTION



The rise of deepfake technology has made it difficult to distinguish authentic facial images from manipulated ones, creating risks to privacy, digital trust, and the spread of misinformation. To address this problem, our project develops a machine-learning solution capable of detecting subtle signs of facial manipulation that are often invisible to the human eye. The system takes a cropped facial image as the input and produces a binary output indicating whether the image is “Real” or “Fake” This solution aims to provide a reliable automated method for identifying manipulated visual content.

02

Related works

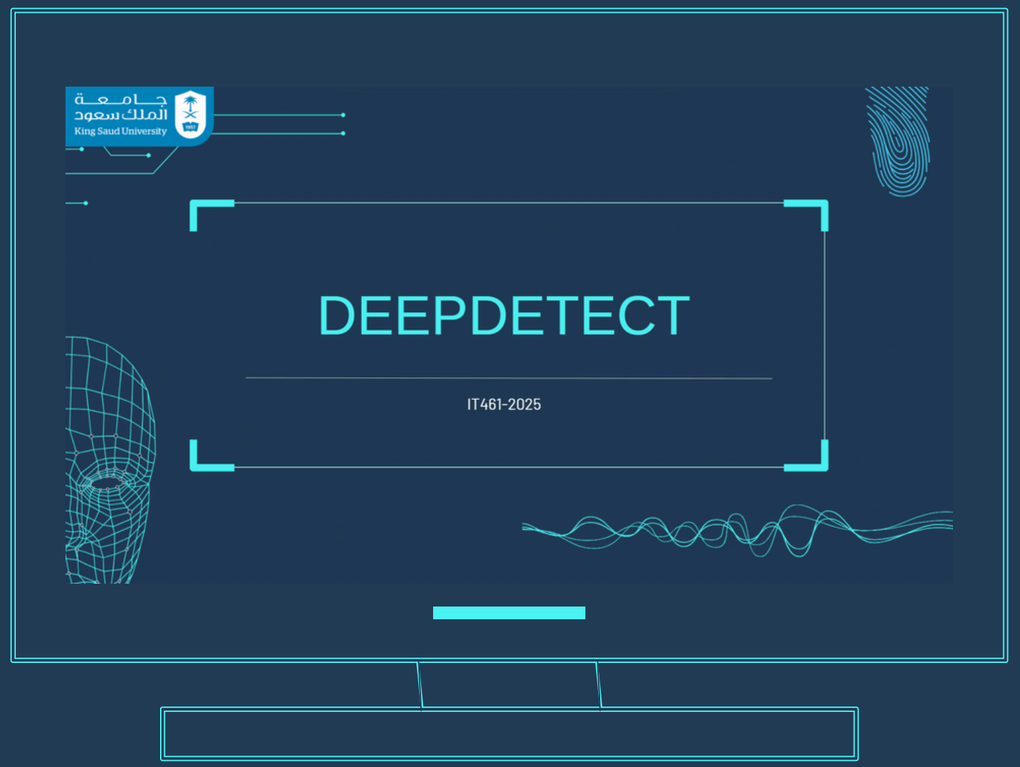


These studies introduced the main datasets and methods used in deepfake detection. They compare different models (CNNs, Capsule Networks, pixel-level analysis) and show how texture and artifact patterns help distinguish real and fake images.

Study	Dataset	Method	Performance	Relevance to DeepDetect
Rössler et al., 2018	FaceForensics++	CNN	~93%	Provides the baseline dataset and CNN foundations we build on.
Yu et al., 2019	DeepFake-TIMIT	Capsule Network	~94%	Shows alternative feature-based architectures beyond CNNs.
Dolhansky et al., 2020	DFDC	CNN + Texture	92–95%	Highlights robustness and real-world variability—aligned with our goal of generalization.
Zhang et al., 2020	Celeb-DF	Pixel-level CNN	~96%	Supports our pixel-focused models (EfficientNet/Xception).

03

Dataset


جامعة الملك سعود
King Saud University
DEEPDETECT
IT461-2025

Dataset Description

- Goal: binary-classification task to detect whether a face image is Real or Fake.
- Source: Kaggle dataset — Deepfake and Real Images by Manjil Karki (2025).
- Samples:
  - 70,000 real images.
  - 70,000 fake images.
  - ~10,000 test images.
- Type: RGB images (256×256 pixels).
- Why this dataset?
  - Large and balanced.
  - Multiple fake generation methods.
  - Public and reliable source.

Examples & Feature Extraction

- Dataset Split:
  - Training: 140,002 images.
  - Validation: 39,428 images.
  - Testing: 10,905 images.

- Feature Extraction:
  - Pre-trained models were used to extract high-level visual features while the convolutional layers were frozen.
  - Only the classifier was trained in the first stage, and selected deeper layers were later fine-tuned to adapt the models to deepfake-specific patterns.

Fake face (AI-generated)

Real face (Authentic)

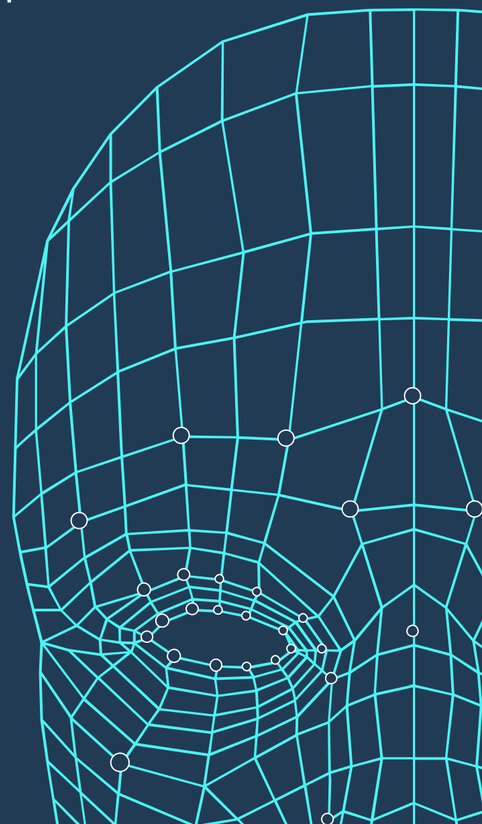
04

Methods

Methods: Models & Preprocessing

- **Three CNN architectures used:**
  - ResNet50: Strong in extracting deep features; reduces vanishing-gradient issues.
  - Xception: Excellent for fine-grained detail analysis using depthwise-separable convolutions.
  - EfficientNetB0: High accuracy with fewer parameters; balanced scaling of depth, width, and resolution.

- **Preprocessing & Augmentation:**
  - Image resizing (ResNet50: 256×256, Xception/EfficientNetB0: 224×224).
  - Normalization aligned with ImageNet standards.
  - Augmentation: horizontal flip, 5–10% rotation, 10–20% zoom.
  - Regularization: Global Average Pooling + Dropout (0.3) to reduce overfitting.



Methods: Training & Fine-Tuning

- Two-stage training approach:

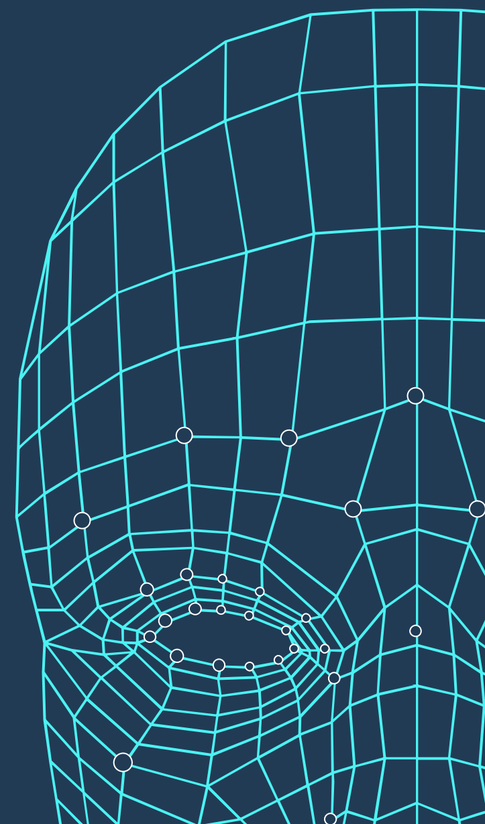
  a. Freeze the backbone + train classifier layers.
  b. Fine-tune top layers for each model

    - ResNet50: top 100/80 layers
    - Xception: top 80 layers
    - EfficientNetB0: last convolution block

- Optimizer & Loss: Adam + Binary Cross-Entropy.
- Callbacks: EarlyStopping, ReduceLROnPlateau, ModelCheckpoint.

Justification:

- Transfer learning boosts performance with limited data.
- Xception and EfficientNetB0 capture subtle artifacts in fake images.
- Outcome: stable training and strong models with reduced overfitting



04

Experiments

Dataset & Preprocessing

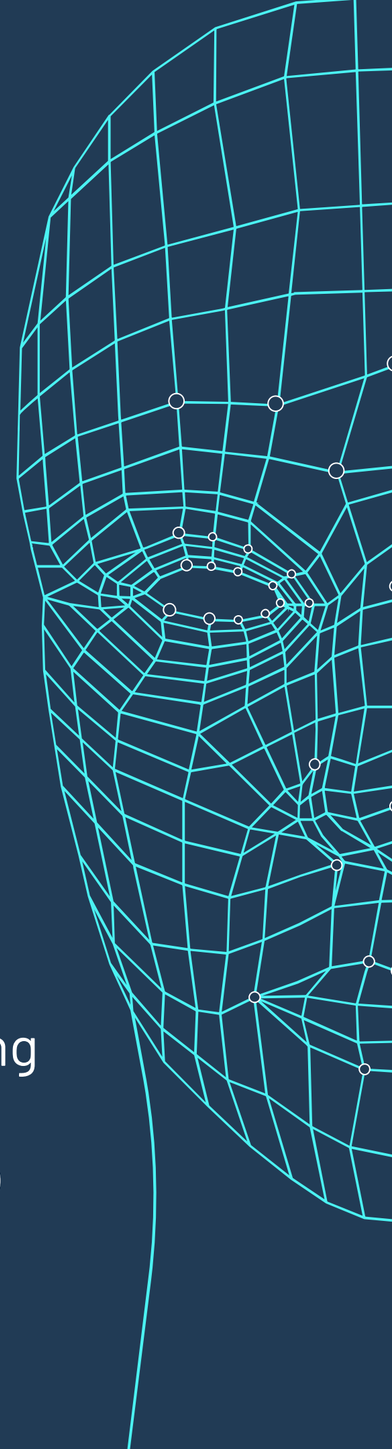
- Used 140K+ Real/Fake face images.
- Split into 70% training, 20% validation, and 10% testing.
- Models trained directly on pixels (no handcrafted features).
- Applied normalization and resizing according to each model's expected input size.

Data Augmentation & Regularization

- Light augmentation: horizontal flip, small rotations & zoom.
- Regularization included dropout and early stopping to improve generalization.

Model Structure (High-Level)

- All models follow a similar flow:
  Input → Augmentation → Normalization → CNN Backbone → Global Average Pooling → Dropout → Sigmoid Output
- Fine-tuning applied on the upper layers of each pretrained model to adapt features to deepfake artifacts.



Hyperparameter Tuning

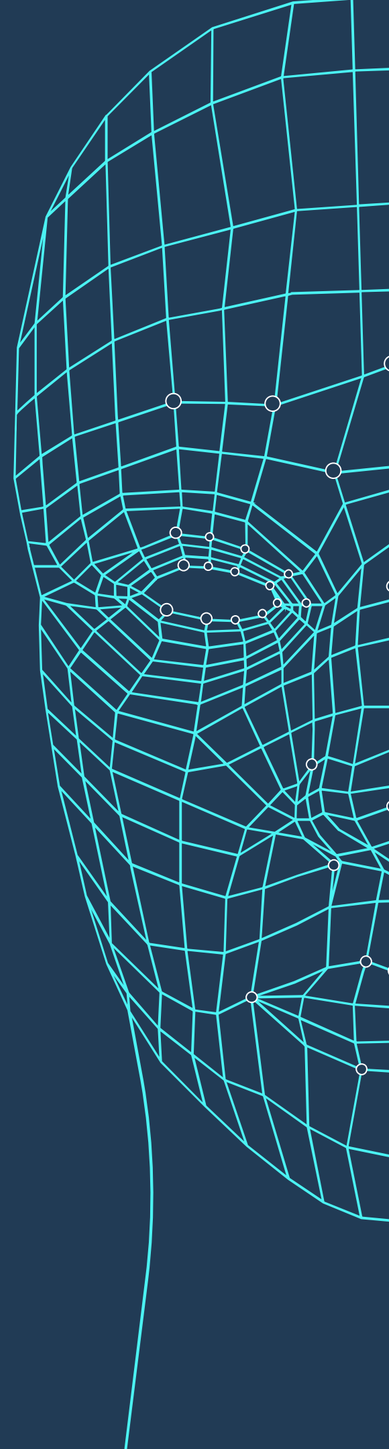
- Tuned:
- Learning rate, batch size, fine-tuned layers, augmentation strength, steps per epoch.
- Explored LR range from 1e-3 → 1e-5 and batch sizes from 2–8 to reach stable convergence.
- Optimal values selected based on highest validation F1 and lowest overfitting.

Model Evaluation

- Performance measured using:
- Accuracy – Precision – Recall – F1-Score
- Confusion matrices used to analyze false positives/negatives.

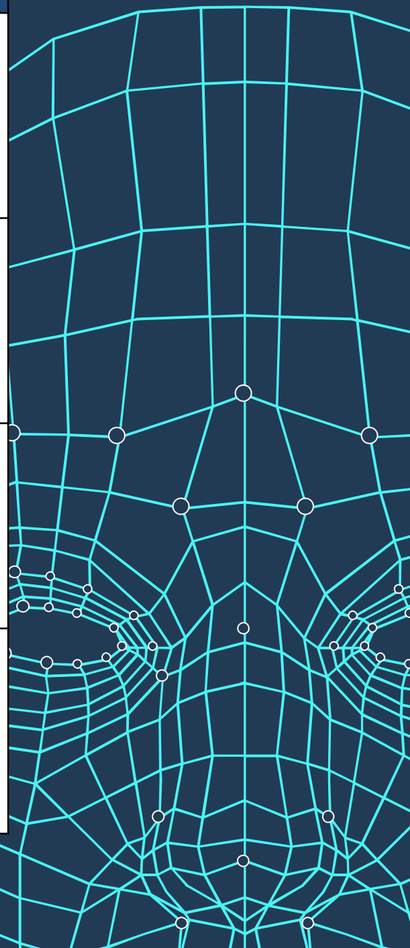
Libraries & Computational Resources

- Libraries: TensorFlow/Keras, NumPy, scikit-learn, Matplotlib.
- Hardware: training on NVIDIA T4, fine-tuning on NVIDIA A100 for higher speed and memory.



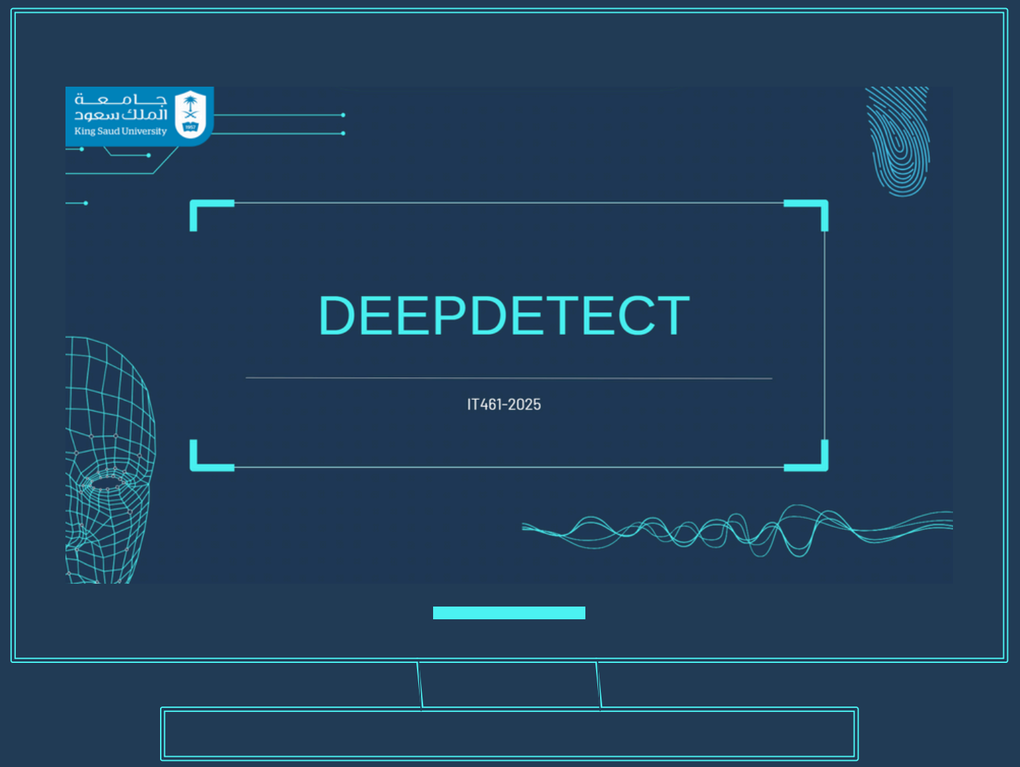
Hyperparameter Tuning Process

Model	LR Stage 1	LR FT Stage	Epochs
ResNet50 v1	**0.0001**	**0.00001**	**20 + FT 8**
ResNet50 v2	**0.0001**	**0.00003**	**15+ FT 4**
EfficientNetB0	**0.0001**	**0.00003**	**15+ FT 4**
Xception	**0.0001**	**0.00003**	**15+ FT 4**



06

Results and Discussion



Deep Learning Models Evaluated

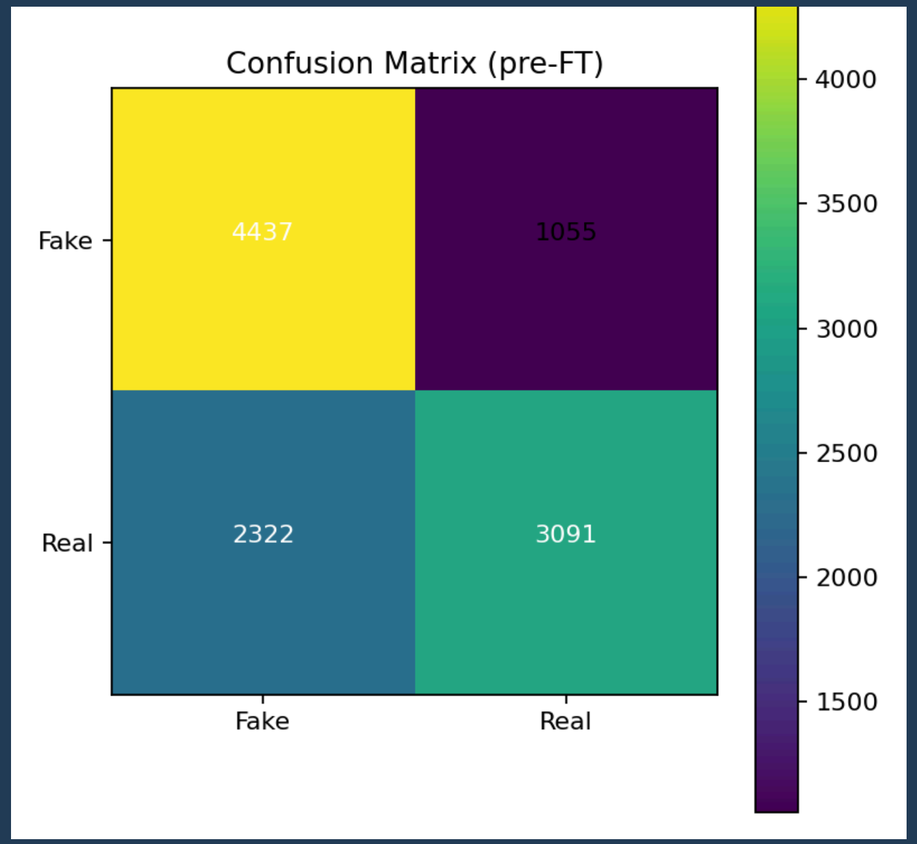
- Xception
- EfficientNetB0
- ResNet50

All models were trained using transfer learning and fine-tuned to improve performance

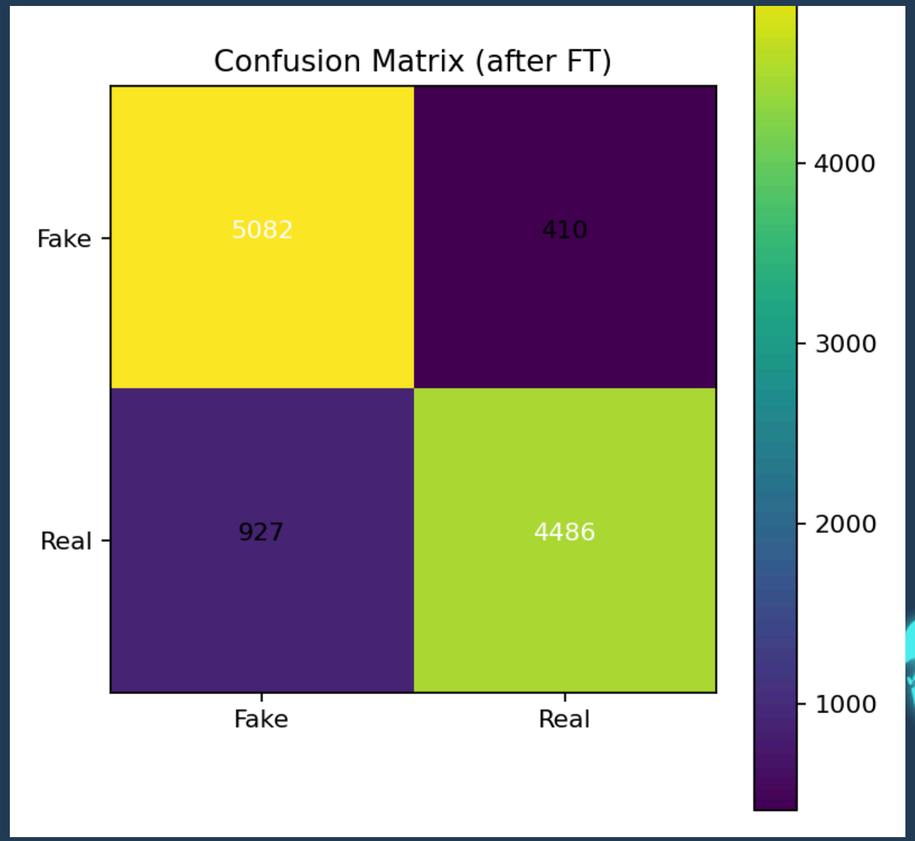
Why These Models?

- Xception: excels at capturing subtle texture differences using depthwise separable convolutions.
- EfficientNetB0: provides strong accuracy with minimal computation due to compound scaling.
- ResNet50: deep residual layers help detect complex manipulation artifacts.

Xception

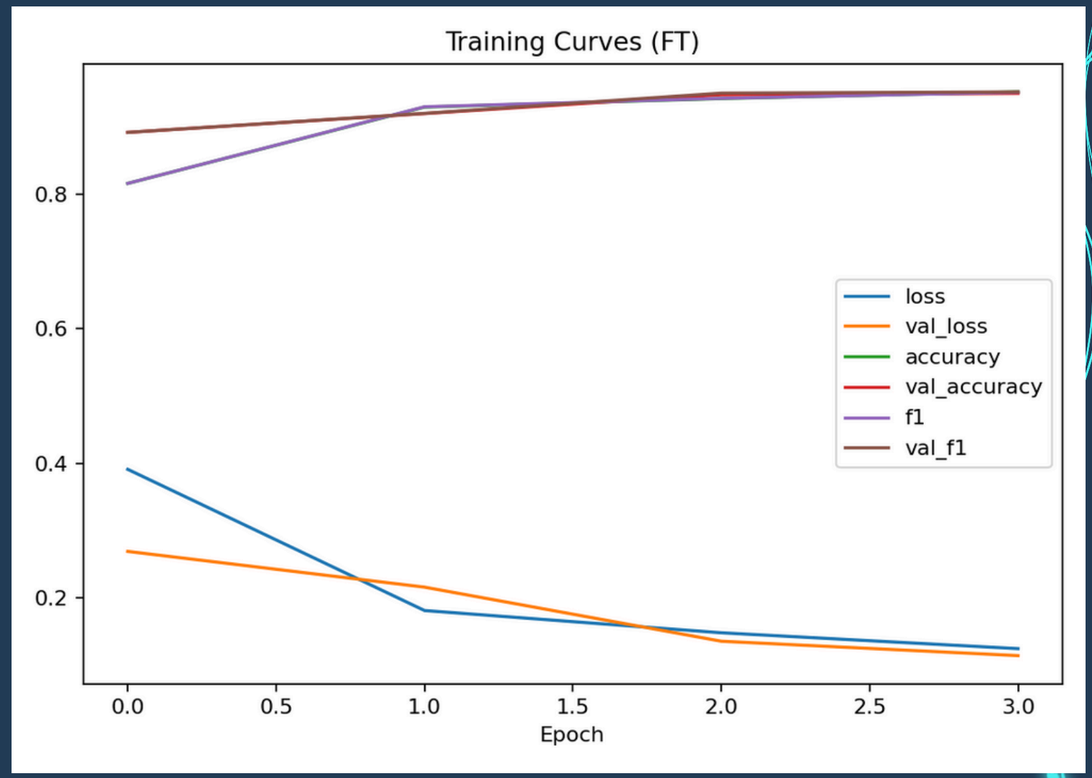


- True Fake : 4437
- False Real: 1055
- False Fake: 2322
- True Real : 3091
- Total : 7528 / 10905 ≈ 68.99%



- True Fake: 5082
- False Real: 410
- False Fake: 927
- True Real: 4486
- Total : 9568 / 10905 ≈ 87.74%

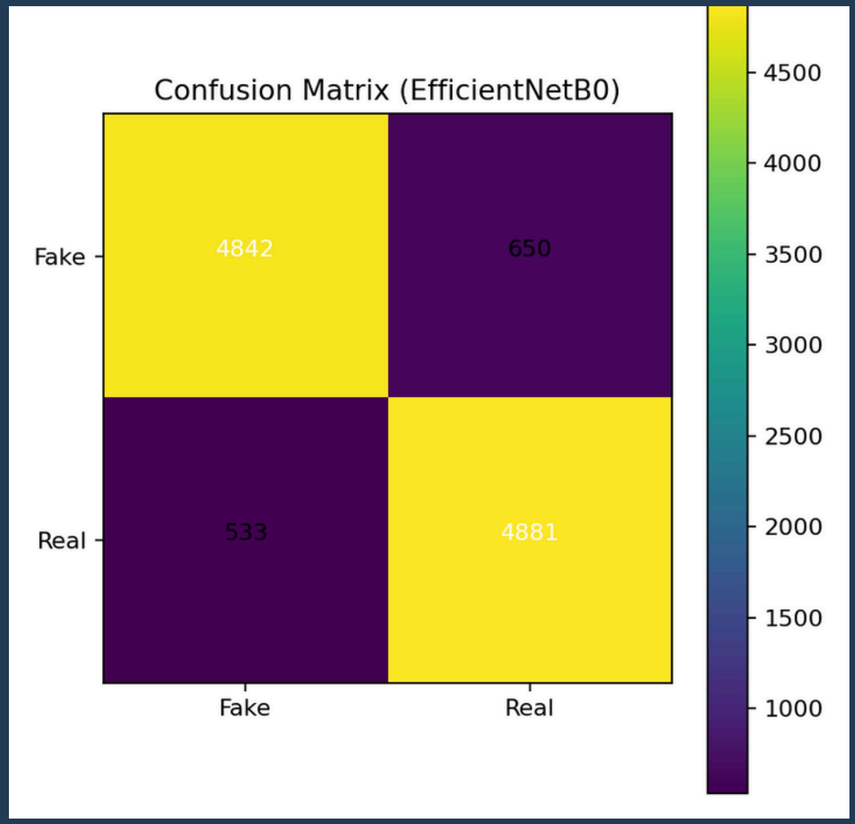
Xception



- Accuracy (T): ~0.73
- Accuracy (V):~0.75
- Training F1:~0.74
- Validation F1: ~0.72
- Training Loss: ~0.53
- Validation Loss: ~0.50

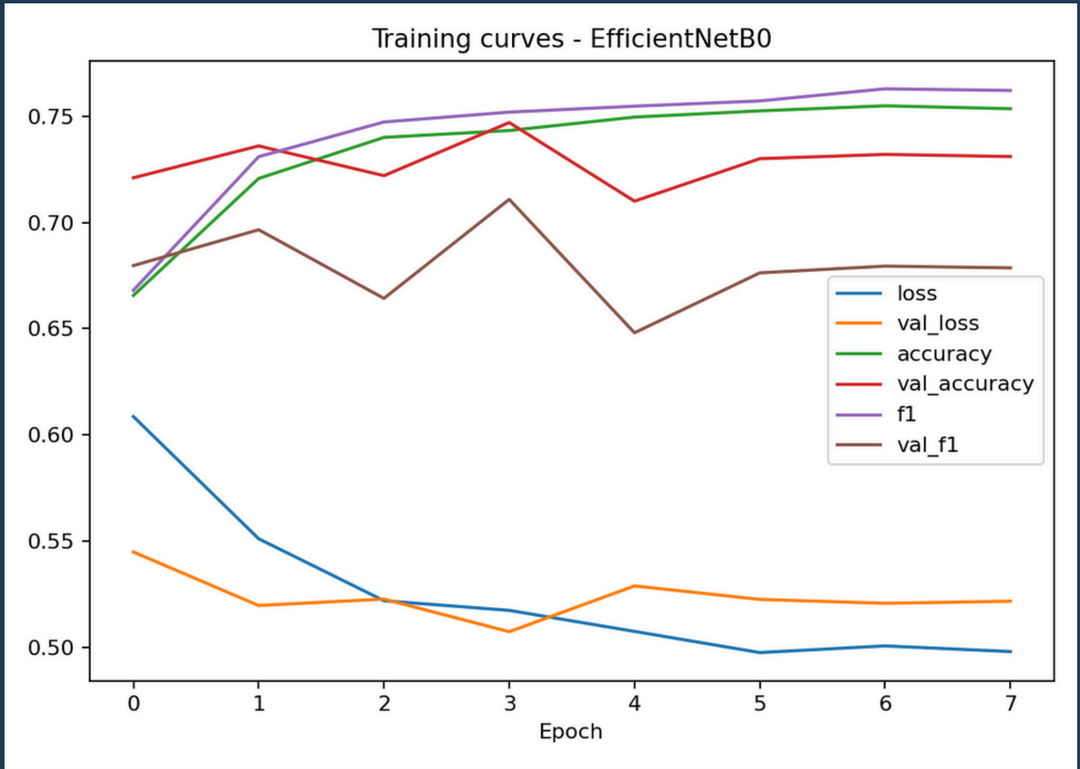
- Accuracy (T): ~0.92
- Accuracy (V):~0.92
- Training F1:~0.92
- Validation F1: ~0.93
- Training Loss: ~0.13
- Validation Loss: ~0.12

EfficientNetB0

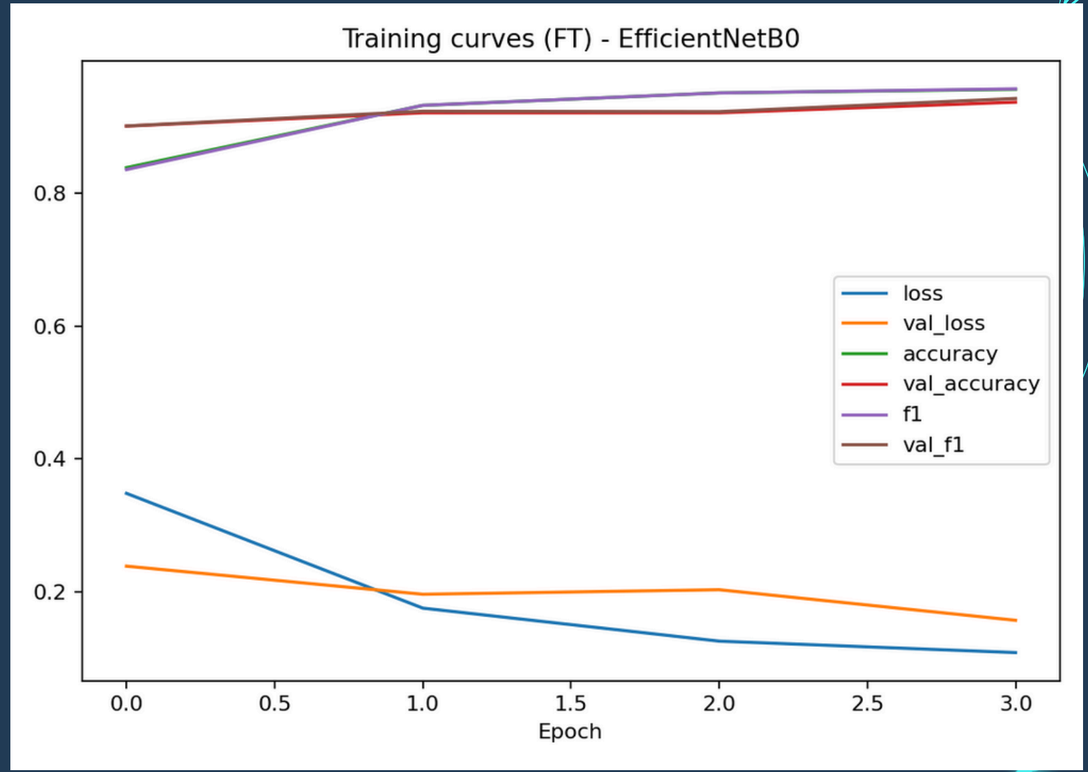


- True Fake : 4842
- False Real : 650
- False Fake : 533
- True Real : 4881
- Total : 9723 / 10906 ≈ 89.15%

EfficientNetB0

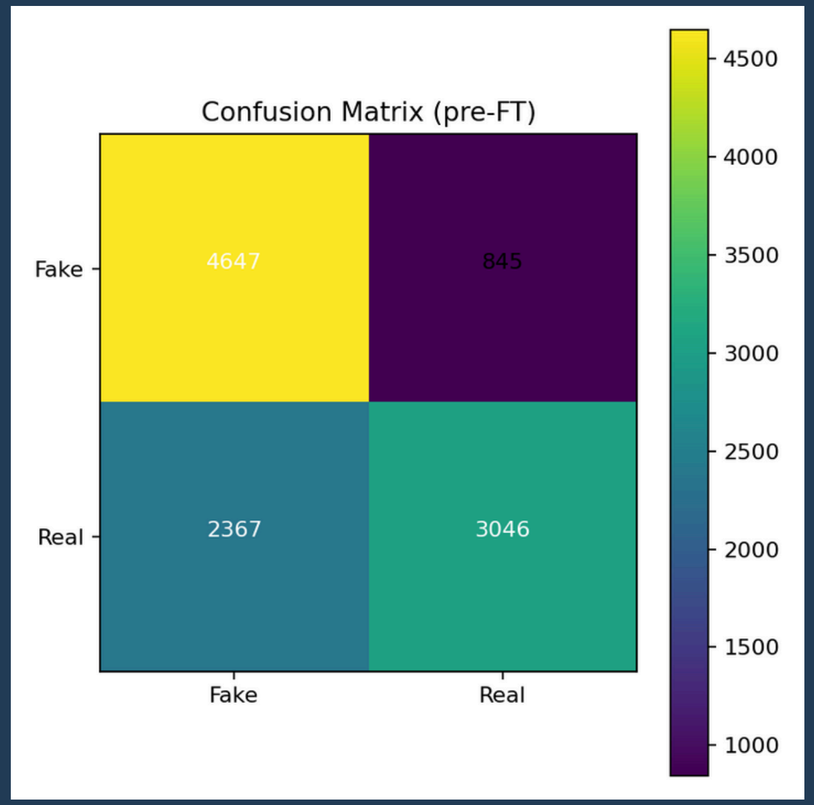


- Train Accuracy : 0.755
- Validation Accuracy : 0.732
- Train F1 : 0.765
- Validation F1 : 0.678
- Train Loss : 0.498
- Validation Loss : 0.521

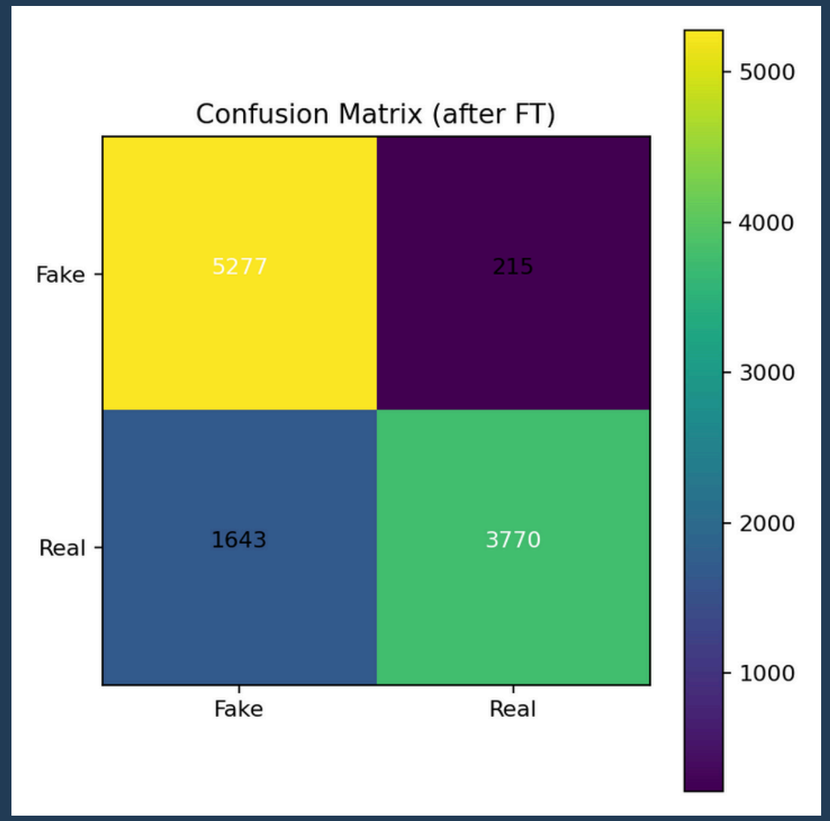


- Train Accuracy : 0.93
- Validation Accuracy : 0.935
- Train F1 : 0.94
- Validation F1 : 0.945
- Train Loss : 0.11
- Validation Loss : 0.155

ResNet50

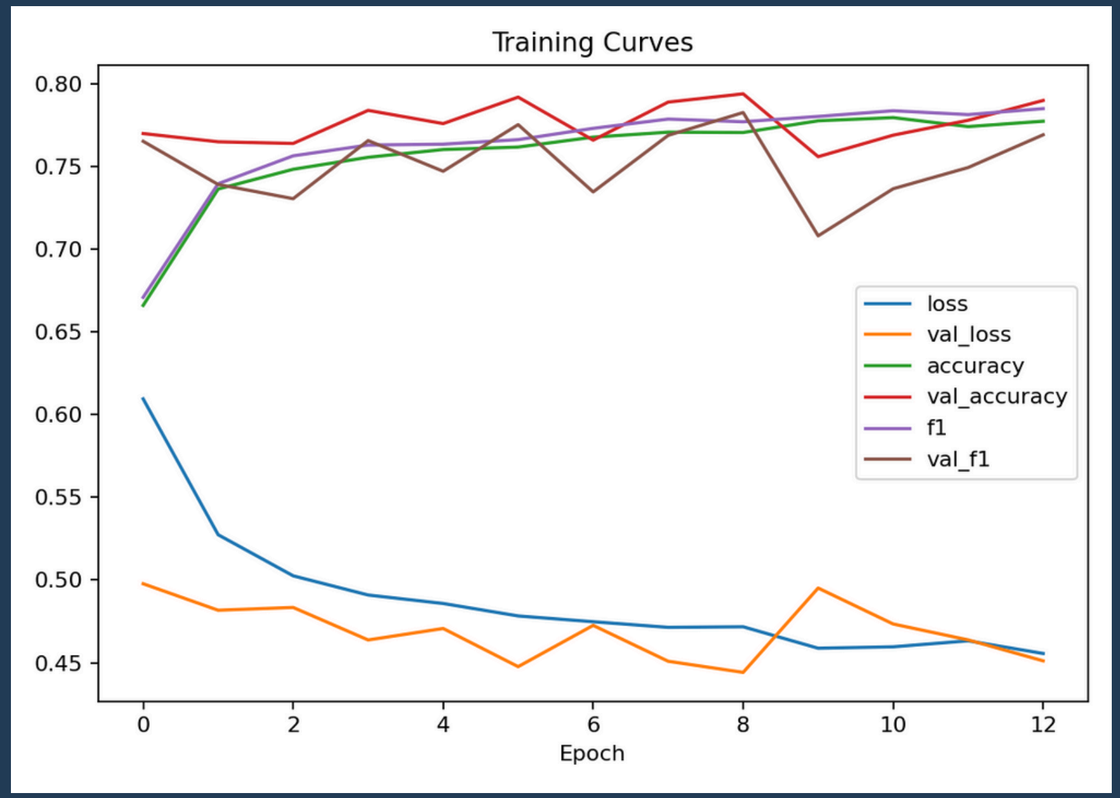


- True Fake: 4647
- False Real: 845
- False Fake: 2367
- True Real: 3046
- Total Accuracy: (4647 + 3046) / 10905 ≈ 70.7%

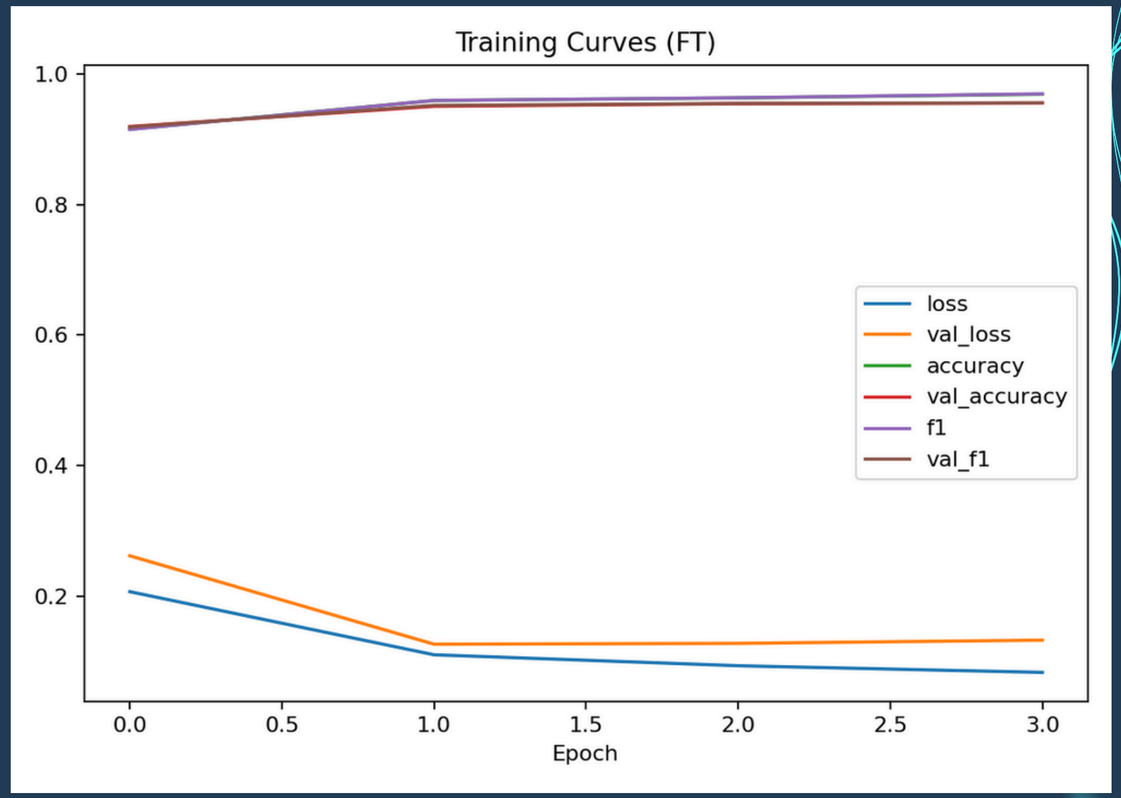


- True Fake: 5277
- False Real: 215
- False Fake: 1643
- True Real: 3770
- Total Accuracy: (5277 + 3770) / 10905 ≈ 82.6%

ResNet50



- Train Accuracy: 0.755
- Validation Accuracy: 0.732
- Train F1: 0.765
- Validation F1: 0.678
- Train Loss: 0.498
- Validation Loss: 0.521



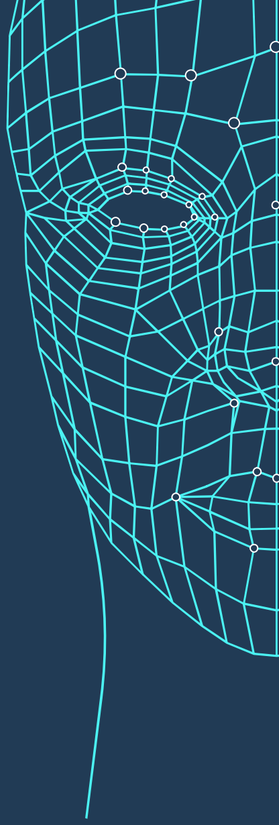
- Train Accuracy: 0.93
- Validation Accuracy: 0.935
- Train F1: 0.94
- Validation F1: 0.945
- Train Loss: 0.11
- Validation Loss: 0.155

Final Comparison Summary

Model	Summary
EfficientNetB0	Best overall & most balanced
Xception	Stable & reliable
ResNet50	Strongest Fake detection

Discussion

- Fine-tuning significantly improved all models, with the largest impact on ResNet50.
- EfficientNetB0 delivered the strongest and most balanced overall performance.
- Xception maintained smooth convergence and strong generalization.
- ResNet50 became highly sensitive to Fake patterns after fine-tuning.



07

Conclusion

Conclusion

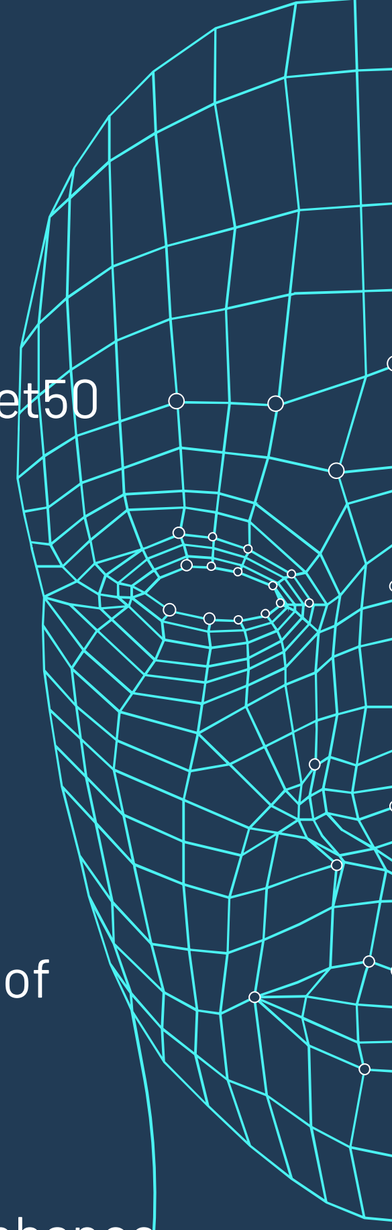
- The system achieved strong performance in detecting real vs fake images.
- EfficientNetB0 delivered the best overall results, followed by Xception, while ResNet50 showed lower generalization

Challenges

- Handling a large dataset.
- Sensitivity to augmentation settings.
- Risk of model overfitting.

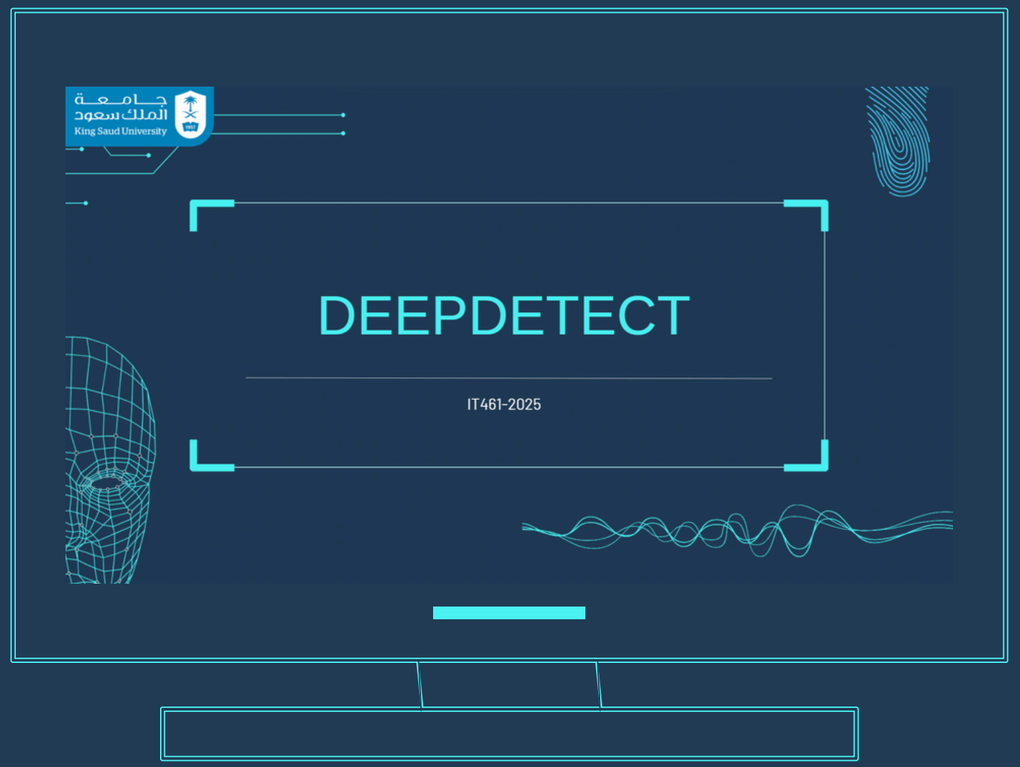
Future Work

- Explore advanced architectures such as Vision Transformers to improve detection of subtle manipulations.
- Incorporate frequency-domain features to reveal hidden deepfake artifacts.
- Extend the system to video-based detection for stronger temporal consistency.
- Increase dataset diversity and test against emerging manipulation techniques to enhance robustnes



08

DEMO


جامعة الملك سعود
King Saud University
DEEPDETECT
IT461-2025

THANKS!

Do you have any questions?

Rawan Albatati

Hissah Alotaibi

Norah Alfaheed

Nora Alasmari

Supervised by
Dr. Luluah Alhusain