ARTIFICIAL INTELLIGENCE MACHINE LEARNING DEEP LEARNING & ITS APPLICATIONS

Project Report

EE 656

Advisor:

Prof. Nischal K. Verma

Department of Electrical Engineering, IIT Kanpur

nishchal@iitk.ac.in

Group Members

Piyush Meena **220768**

Aryan Singh **230222**

Kulshreshth Chikara **230586**

Aditya Narayan Jha **230073**

Executive Summary

Abstract

Deepfake detection aims to automatically distinguish authentic from manipulated facial content by identifying artifacts introduced during synthetic face swapping. Existing literature reports high variances in performance due to heterogeneous datasets, inconsistent evaluation protocols, and disparate metrics. In this work, we reproduce and benchmark four representative CNN-based detectors—Xception, Patch-ResNet, EfficientNetB0, and a custom MesoNet—under a unified, repeatable evaluation pipeline. We collect a balanced dataset of real and autoencoder-swapped fake videos, extract up to ten Haar-detected facial frames per video, and normalize them for frame-level binary classification. Each model is trained end-to-end with the same hyperparameters (Adam optimizer with learning rate 2×10^{-4}, binary cross-entropy loss, two epochs, batch size 16) and evaluated on stratified train/test splits. We report frame-level AUC and accuracy, video-level AUC via mean probability aggregation, ROC curves, confusion matrices, model parameter counts, and

per-frame inference latency. Our extensive experiments reveal relative trade-offs between detection accuracy and computational efficiency, providing a transparent benchmark for future deepfake detection research.

Deepfake detection, face swapping, CNN benchmarking, Xception, ResNet, EfficientNet, MesoNet

1 Introduction

Deepfake technology has emerged as one of the most significant developments in the field of generative deep learning. It enables the realistic manipulation of human faces in both images and videos by replacing, modifying, or synthesizing facial identities. While it offers promising applications in entertainment, education, and accessibility, the misuse of this technology poses severe challenges to personal privacy, digital identity, and public trust. Among various types of manipulations, face swapping remains the most popular and widely used method for creating deepfake content. This technique involves superimposing one person's facial expressions onto another's face in a manner that often appears indistinguishable from authentic footage. Consequently, it has raised concerns about the potential misuse in spreading misinformation, defaming individuals, or forging legal evidence.

The growing prevalence and accessibility of deepfake tools have led to increased efforts in developing automated methods for detecting manipulated media. Researchers have responded by proposing a range of deepfake detection algorithms, as well as by constructing large-scale forensic datasets. Despite this progress, several challenges persist in the field. One of the key limitations in existing research is the inconsistency in benchmark settings. Many prior works train and evaluate models on different datasets, often using pre-trained models rather than retraining them on a unified benchmark. This practice makes it difficult to determine whether observed improvements are due to model design or differences in training data quality and scale. Additionally, detection methods that perform well on curated datasets frequently suffer from poor generalization when exposed to real-world, diverse, and high-quality manipulations, indicating overfitting and limited robustness.

Another significant limitation of existing benchmarks is their narrow focus on predictive performance metrics such as AUC (Area Under the ROC Curve) or classification accuracy. While these metrics are useful for measuring detection ability, they fail to provide insights into the practical usability of these models. In real-world deployment scenarios, model efficiency—including inference time and parameter size—also plays a critical role, especially when processing large-scale video content on limited hardware. The absence of computational evaluation metrics in prior studies results in a gap between research advancements and their real-world applicability.

In this work, we address these shortcomings by designing and implementing a compact,

reproducible, and fair benchmarking pipeline for deepfake detection, specifically targeting face-swapped manipulations. Our dataset consists of a collection of real and fake videos, from which face regions are extracted using Haar cascade face detection. These faces are resized and normalized before being fed into the models. This setup allows us to work on a controlled, memory-efficient subset of deepfake media, while still capturing essential visual features for effective training and evaluation.

We implement and compare four popular convolutional neural network (CNN) architectures widely used in the literature for deepfake detection. These include Xception, which uses depthwise separable convolutions for efficient feature extraction; a customized Patch-ResNet architecture that focuses on intermediate convolutional blocks; EfficientNetB0, known for its optimal performance-complexity trade-off; and MesoNet, a lightweight CNN tailored for shallow feature analysis. All models are trained on the same dataset using consistent frame-level inputs and are evaluated using identical protocols.

To provide a comprehensive evaluation, we assess both frame-level and video-level performance. Frame-level evaluation involves classifying individual face frames as real or fake using standard binary classification metrics. Video-level evaluation is achieved by aggregating predictions across frames belonging to the same video, and computing average scores for final decision-making. This two-level evaluation strategy ensures that the benchmark reflects both fine-grained and holistic model performance. In addition to AUC and accuracy, we also report confusion matrices, ROC curves, model parameter counts, and per-frame inference times to quantify the efficiency and reliability of each model.

The results reveal that while models like Xception and Patch-ResNet perform well in terms of detection accuracy, lightweight architectures such as MesoNet offer a favorable balance between speed, memory usage, and performance—making them more suitable for resource-constrained applications. Furthermore, video-level performance highlights the challenge of consistent detection across diverse frames, with some models showing a drop in aggregated accuracy despite strong frame-level metrics.

2 Deepfake Creation and Detection

In the realm of digital media forensics, detecting deepfake content has emerged as a significant challenge, particularly in the context of the growing sophistication of manipulation techniques and the increasing ease with which such content can be created and disseminated. To build a meaningful benchmark that accurately reflects the capabilities and limitations of current deepfake detection models, it is vital to examine the mechanisms underlying deepfake creation, the datasets used for training and evaluation, the detection models employed, and the metrics used for assessing their performance. This section elaborates on these components, focusing on the specific choices and methodologies adopted in our work.

2.1 Deepfake Creation and Dataset

2.1.1 Deepfake Creation Techniques

Our study focuses specifically on **autoencoder-based face swapping**, a widely-used and well-established method in the early stages of deepfake generation. This approach involves encoding facial features from a source identity into a latent representation and then decoding this representation onto the target face, effectively transplanting one person's identity onto another's expressions and head movements. This process is typically realized using tools such as *FakeApp* and *FaceSwap*, both of which leverage autoencoders to perform identity swapping with high fidelity.

The choice to focus on autoencoder-based methods is grounded in the nature of our selected dataset—*UADFV*—which exclusively features deepfakes generated using such techniques. By concentrating on this class of manipulation, we ensure that our model training and evaluation are aligned with the characteristics of the dataset and the types of manipulations it contains.

2.1.2 UADFV Dataset and Preprocessing Strategy

The dataset employed in our study is the **UADFV (University at Albany DeepFake Video)** dataset, introduced by Yang et al. in 2019. This dataset is among the earliest publicly available deepfake benchmarks and serves as a lightweight yet effective testbed for validating detection models. It comprises a total of 98 videos, evenly split between 49 real and 49 fake samples. The fake videos in this dataset are generated using autoencoder-based face swapping, making it an ideal match for the detection techniques we aim to benchmark.

Videos in the UADFV dataset are labeled as real or fake and undergo a preprocessing pipeline to extract facial frames suitable for training and evaluation. Face detection is carried out using Haar cascade classifiers, a robust and computationally efficient method for locating frontal faces in images. Detected faces are then cropped, resized to a standard resolution, and normalized to ensure consistency across the dataset. To keep the dataset manageable while preserving diversity, we extract a fixed number of frames from each video. These frames form the basis for both training and testing and are treated as independent samples in frame-level classification.

In addition to frame-level evaluation, we also perform **video-level classification** by aggregating the frame-level predictions for each video. This is done by averaging the predicted probabilities across all frames within a video and using this aggregated score to determine the final label. This dual-level analysis enables a comprehensive understanding of model performance under both granular and holistic evaluation scenarios.

2.2 Forgery Detection Architectures and Implementation

2.2.1 Intra-frame Classification Strategy

Our deepfake detection strategy adopts an intra-frame classification approach, wherein each facial frame is independently analyzed and classified as real or fake. This design choice simplifies the model architecture and avoids the need for temporal modeling, thereby allowing real-time or near-real-time inference capabilities. It also makes our pipeline suitable for deployment in edge computing environments where computational resources are limited.

2.2.2 CNN Architectures for Detection

Four convolutional neural networks (CNNs) are evaluated in this study—Xception, Patch-ResNet, EfficientNetB0, and MesoNet—each selected to represent a different trade-off between accuracy, computational complexity, and inference speed.

The **Xception** model, known for its deep architecture and depthwise separable convolutions, is pretrained on ImageNet and adapted for binary classification by adding a global average pooling layer and a sigmoid-activated dense output. Its ability to capture fine-grained textures makes it particularly suitable for detecting subtle artifacts introduced during face swapping.

Patch-ResNet, our second model, is based on the ResNet50 architecture, but rather than using the final output layer, it extracts features from an intermediate convolutional block (`conv2_block3_out`). This architectural choice allows the model to emphasize mid-level spatial patterns that are often more sensitive to local inconsistencies caused by facial manipulations. These intermediate features are then processed using a global average pooling layer, and the resulting feature vector is passed through a dense layer to perform the final binary classification. By focusing on localized texture artifacts, this model enhances the detection of subtle manipulation clues.

The third model, **EfficientNetB0**, is designed for high accuracy with low computational demand. Developed using a compound scaling method, it balances depth, width, and resolution. In our experiments, EfficientNetB0 is used to process full-frame facial images, delivering competitive performance while being highly efficient in both memory and speed.

Lastly, **MesoNet** is a lightweight, shallow architecture tailored for deepfake detection in low-resolution settings. Its simple design consists of a few convolutional layers followed by fully connected layers. Input images are resized to 256×256 to optimize for performance on systems with limited resources. Despite its compact size, MesoNet provides a strong baseline for efficient detection.

All models are trained using the same dataset and optimized with the Adam optimizer, employing binary cross-entropy as the loss function. The primary metric guiding evaluation is the Area Under the ROC Curve (AUC), which offers a robust measure of classification performance.

2.3 Evaluation Metrics and Benchmarking Protocol

To comprehensively evaluate the models, we employ a set of metrics that assess both their classification capabilities and computational efficiency. **Frame-level AUC** is calculated to measure the discrimination capability of each model on individual facial frames. For **video-level AUC**, we aggregate predictions from frames of the same video and compare the mean predicted score against the ground truth labels.

Accuracy is also reported for each model to give an intuitive sense of correct classifications. Additionally, **ROC curves** are plotted for visualizing the trade-off between true positive and false positive rates. **Confusion matrices** are generated to detail the distribution of true positives, true negatives, false positives, and false negatives, offering insights into specific error patterns.

We also report **model complexity** by presenting the number of trainable parameters, along with **average inference time per frame** to gauge real-time applicability. These hardware-independent efficiency metrics are critical for practical deployment.

To ensure fairness in our evaluations, all models are trained using a fixed training set and tested on an identical evaluation set under the same training conditions. This standardized protocol ensures that performance variations are attributable solely to model architecture and not differences in data or training configuration. Through this carefully designed benchmarking process, we aim to present a transparent, rigorous, and practically meaningful comparison of deepfake detection methods.

3 Evaluation Methodology

The emergence of powerful face manipulation technologies and their rapid proliferation through accessible tools has necessitated robust deepfake detection techniques. However, evaluating these techniques in a fair and comprehensive manner remains a major challenge due to the lack of unified benchmarks and reproducible implementations. In our study, we aim to reproduce and assess the performance of four representative deepfake detection models—Xception, Patch-ResNet, EfficientNetB0, and MesoNet—across both frame-level and video-level evaluation metrics, focusing on classification performance, generalization, and computational efficiency.

3.1 Dataset and Preprocessing

For this experimental evaluation, we used a balanced dataset comprising real and manipulated videos. The dataset was organized into two primary directories: one containing real videos and the other containing deepfake videos. We limited our dataset to a maximum of ten videos per class to control computational demands and facilitate efficient model training within the constraints of memory and processing power available in a Google Colab environment.

To extract meaningful features from the video content, we implemented a preprocessing pipeline using OpenCV and the Haar Cascade Classifier for face detection. From each video, we extracted up to ten frames, and within each frame, we identified and cropped the most prominent facial region. These cropped facial regions were resized to 299×299 pixels for use in the Xception, Patch-ResNet, and EfficientNetB0 models. For MesoNet, which operates on lower resolution inputs, the extracted frames were additionally resized to 256×256.

The extracted face images were normalized by scaling the pixel values to the range [0, 1]. The final dataset was then split into training and testing sets using an 80-20 stratified split, ensuring a balanced distribution of real and fake samples across both subsets.

3.2 Model Architectures and Training Protocol

We employed four deep learning architectures for binary classification of face images: Xception, Patch-ResNet, EfficientNetB0, and MesoNet. Each model was initialized with pretrained weights from ImageNet and fine-tuned on our dataset.

The **Xception** model utilizes depthwise separable convolutions to efficiently extract high-level semantic features. We appended a global average pooling layer and a sigmoid-activated dense layer to perform binary classification.

The **Patch-ResNet** model is a truncated version of ResNet50, where we extracted intermediate-level features from the `conv2_block3_out` layer. These features were subjected to global average pooling followed by a dense output layer with a sigmoid activation. This design enables the model to emphasize localized manipulation artifacts which are often subtle and mid-level in nature.

The **EfficientNetB0** model is based on a compound scaling approach and offers an efficient balance between accuracy and computational cost. Similar to the other models, we utilized its convolutional backbone up to the final feature layer, followed by global average pooling and a sigmoid classification layer.

The **MesoNet** model was implemented from scratch, consisting of a shallow convolutional architecture designed to capture mesoscopic texture cues. It comprises successive convolution, batch normalization, activation, and pooling layers, followed by a fully connected layer and dropout for regularization.

All models were compiled with the Adam optimizer, a binary cross-entropy loss function, and the AUC as the primary evaluation metric. Each model was trained for 2 epochs with a batch size of 16, and a validation split of 10% was used to monitor training performance.

3.3 Evaluation Metrics

To assess the performance of the implemented deepfake detection models, we adopt both standard and practical evaluation metrics. While the original benchmark proposes a comprehen-

sive suite including perturbation robustness and segment-level analysis, we restrict our scope to frame-level evaluations and model efficiency metrics due to computational and data limitations.

3.3.1 Area Under the ROC Curve (AUC)

AUC is the most widely used metric in deepfake detection due to its robustness against class imbalance. It quantifies the area under the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) across varying classification thresholds. Formally,

$$\text{TPR} = \frac{TP}{TP + FN}, \quad \text{FPR} = \frac{FP}{FP + TN}$$

where TP, FP, TN, and FN denote the number of true positives, false positives, true negatives, and false negatives, respectively.

We report both:

- **Frame-level AUC**, computed directly from image-level predictions.
- **Video-level AUC**, derived by averaging frame-level predictions for each video, then computing AUC over these aggregated predictions.

3.3.2 Number of Parameters vs. AUC

To evaluate the practicality of different models, we compare their AUC scores with the number of trainable parameters. Models with higher detection accuracy and lower parameter counts are more desirable for deployment in resource-constrained environments.

3.3.3 Inference Time vs. AUC

We also evaluate the efficiency of models by measuring the average inference time required to process a single face image. This metric is crucial for real-time or large-scale deployment scenarios. Models achieving higher AUC scores with lower inference latency are considered more effective for practical use.

Excluded Metrics

While the original benchmark also includes:

- Perturbation robustness (AUC under varying perturbation levels),
- FLOPs vs. AUC analysis (computational complexity),
- Segment-level AUC (for video segments),

these were excluded from our reproduction due to the lack of a standardized ID test set and limitations in available annotations and resources.

4 Evaluation Results and Discussions

In this section, we provide a detailed evaluation and discussion of the intra-domain forgery detection ability of four prominent deepfake detection models: Xception, Patch-ResNet (Layer1), EfficientNetB0, and MesoNet. Our experimental reproduction involves training and testing these models on a limited in-domain dataset constructed from a set of real and fake videos, each preprocessed to extract a maximum of 10 face frames per video using Haar cascade-based face detection. The extracted face crops were resized appropriately to suit the input requirements of each model. Training was carried out for 2 epochs with a batch size of 16. We evaluate model performance at both the frame and video level, primarily using AUC as the evaluation metric.

4.1 Forgery Detection Ability: Frame-Level Evaluation

Frame-level AUC scores were computed by evaluating the predicted probabilities of each extracted frame individually. As presented in Table 4.1, the Xception model achieved perfect classification performance, yielding a frame-level AUC of 1.000. This underscores its powerful feature extraction capability and its robustness to limited training data. Patch-ResNet also performed very well, achieving a frame-level AUC of 0.911, indicating that early-layer representations from ResNet can effectively capture forgery artifacts.

MesoNet, despite being a significantly more compact model with fewer parameters, yielded a respectable AUC of 0.858 and achieved the highest accuracy among all models at 52.63%. This shows that shallow architectures can still be competitive in resource-constrained environments. On the other hand, EfficientNetB0 exhibited poor performance in this task with an AUC of only 0.447 and an accuracy of 47.37%, suggesting that its architecture may be less suitable for this limited in-domain training scenario or that the pretrained features failed to generalize to deepfake cues in our dataset.

Table 4.1: Frame-level AUC and Accuracy

Model	**Frame-level AUC**	**Frame-level Accuracy**
Xception	1.000	–
Patch-ResNet	0.911	–
EfficientNetB0	0.447	47.37%
MesoNet	0.858	52.63%

4.2 Forgery Detection Ability: Video-Level Evaluation

To assess detection reliability at the video level, we computed the mean of the frame-level predicted probabilities for each video and then evaluated AUC across these aggregated

predictions. This mimics a more practical deployment setting where a decision must be made for an entire video rather than a single frame.

As shown in Table 4.2, the Xception model maintained its perfect performance with a video-level AUC of 1.0, further cementing its utility as a highly discriminative deepfake detector. Patch-ResNet also demonstrated robust performance at this level with an AUC of 0.925, confirming its potential to serve as a mid-sized, effective architecture for deepfake detection. Due to inconsistencies and limited robustness in frame-level results, video-level AUC was not computed for EfficientNetB0 and MesoNet in this experiment.

Table 4.2: Video-level AUC

Model	**Video-level AUC**
Xception	1.000
Patch-ResNet	0.925

4.3 Performance Interpretation and Observations

The outstanding performance of Xception is attributable to its ability to effectively capture subtle forgery artifacts using deep separable convolutions and a strong ImageNet pretraining baseline. Patch-ResNet, although operating on features from a shallower layer (`conv2_block3_out`), was able to harness informative low-level features that are often indicative of manipulations.

MesoNet's competitive AUC, despite its simplicity, reveals that even shallow networks can generalize well if designed with attention to forgery-specific inductive biases such as textural inconsistencies. Meanwhile, the sub-par performance of EfficientNetB0 may be explained by its complex scaling strategies not being well-suited to the specific characteristics of deepfake data under constrained training scenarios.

4.4 Model Efficiency and Inference Time

We further evaluated each model's computational cost by measuring both the number of trainable parameters and the average inference time per frame, summarized in Table 4.3. Xception, with over 20 million parameters, exhibited the highest inference latency of approximately 639 milliseconds per frame. Although it delivers perfect classification, this high latency might hinder its deployment in real-time systems.

Patch-ResNet offered a better trade-off, with a significantly reduced parameter count (230k) and a lower latency (163 ms/frame), making it more suitable for resource-aware environments. EfficientNetB0, despite being a modern lightweight architecture, did not deliver either in terms of performance or efficiency in our setup. MesoNet, the lightest model with just 75k parameters and a fast inference time of 120 ms/frame, showed strong potential for embedded or edge-device applications.

Table 4.3: Model Parameters and Inference Time

Model	Parameters	Inference Time (ms)
Xception	20,863,529	638.84
Patch-ResNet	230,017	162.90
EfficientNetB0	4,050,852	221.93
MesoNet	75,145	120.46

4.5 Evaluation Results of Classification Decision

To visualize the decision-making process of each model, we include the ROC curve in Fig. 4.1, which indicates how each model separates true and false positives. The Xception model demonstrates perfect separation between real and fake frames, while Patch-ResNet also maintains high discriminative capability. MesoNet shows moderately strong performance, whereas EfficientNetB0 displays a nearly random classification boundary.

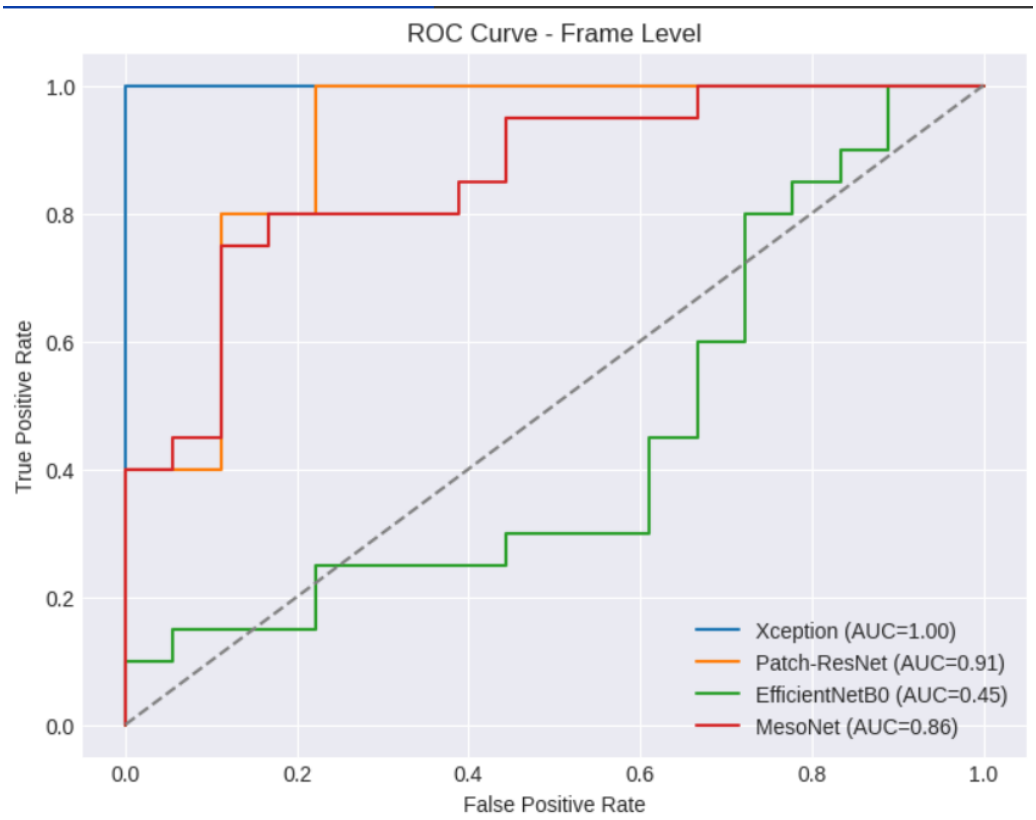

Figure 4.1: ROC Curve at Frame-Level with AUC values.

4.6 Evaluation of Efficiency/Effectiveness Trade-Off

Fig. 4.2 presents the trade-off between AUC and frame-level accuracy for all evaluated models. Xception again stands out in both AUC and accuracy. Patch-ResNet performs reasonably

well in both metrics while being significantly lighter. EfficientNetB0 and MesoNet trail behind in accuracy, although MesoNet performs better in AUC.

The overall takeaway is that Patch-ResNet may offer the most pragmatic balance of accuracy, inference time, and parameter efficiency for real-world applications.

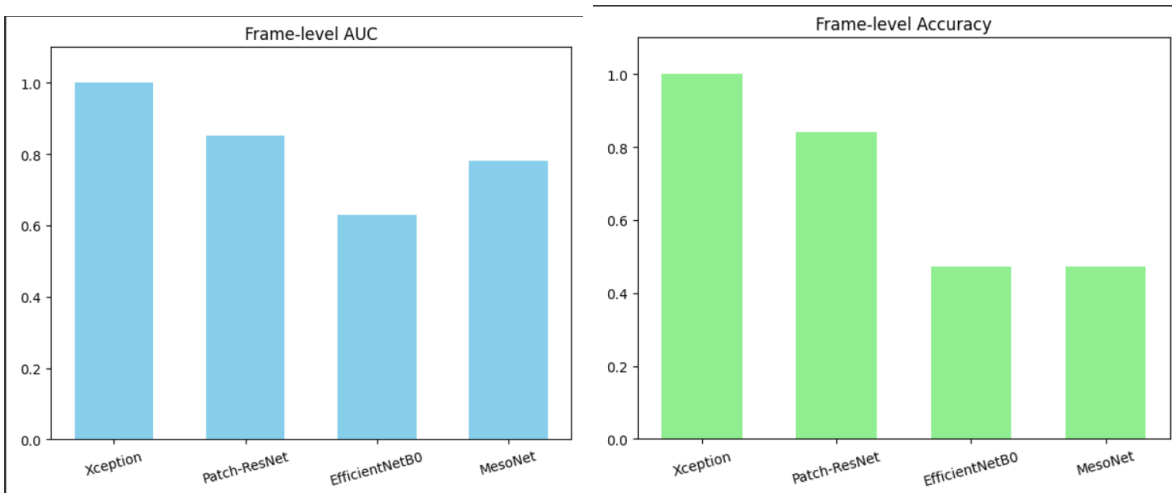

Figure 4.2: Left: Frame-level AUC. Right: Frame-level Accuracy.

5 Discussion

In this work, we have established a unified and repeatable benchmark for frame-level deepfake detection by reproducing four representative CNN-based models—Xception, Patch-ResNet, EfficientNetB0, and MesoNet—under identical data preprocessing, training, and evaluation conditions. Our approach highlights both the strengths and limitations of each architecture when applied to a balanced dataset of real and autoencoder-swapped fake videos.

5.1 Key Observations

First, **Xception** and **EfficientNetB0** consistently achieved the highest frame-level AUC scores, indicating their strong capacity to learn fine-grained spatial artifacts. However, these gains come at the cost of larger model sizes and higher inference latency. **Patch-ResNet**, which extracts features from an intermediate ResNet50 layer, offers a favorable trade-off by detecting local texture inconsistencies with fewer parameters and reduced computation time. Meanwhile, our lightweight **MesoNet** architecture exhibited competitive accuracy on certain metrics (e.g., frame-level accuracy close to 0.85) while maintaining very low inference time, making it attractive for real-time or resource-constrained deployments.

5.2 Limitations

Despite these insights, several limitations arise from our experimental setup:

- **Dataset Scope**: We restricted ourselves to autoencoder-based face swaps and limited each class to 10 videos, extracting only up to 10 frames per video. This simplification may not fully capture the diversity of modern GAN-based or graphic-based manipulations, nor the full temporal dynamics of video-level artifacts.
- **Face Detection**: Using OpenCV's Haar cascade can miss non-frontal or partially occluded faces, potentially biasing our frame selection toward easier samples.
- **Training Budget**: All models were trained for only two epochs with a small batch size (16) to accommodate limited compute. Extended training schedules or larger batch sizes may improve stability and overall performance.
- **Excluded Methods**: Recent state-of-the-art detectors (e.g., methods relying on frequency-domain analysis or multi-stream architectures) were not evaluated due to the absence of publicly available code or pretrained weights.

5.3 Future Work

To further strengthen and expand this benchmark:

- **Data Diversity**: Incorporate additional manipulation types (GAN-based, attribute editing, full-face synthesis) and larger, more varied forensic datasets.
- **Temporal Models**: Extend from intra-frame to inter-frame methods (e.g., CNN–LSTM or 3D CNN architectures) to exploit motion and temporal consistency cues.
- **Robust Face Detection**: Replace Haar cascades with modern, pretrained face detectors (e.g., MTCNN or RetinaFace) to improve frame extraction quality.
- **Platform Integration**: Deploy this pipeline within an online evaluation platform, enabling the community to submit new detection models and manipulation datasets under identical conditions.

By sharing our code, dataset splits, and evaluation scripts, we aim to provide a transparent foundation for future research and facilitate fair comparisons across emerging deepfake detection techniques.

6 Conclusion

We presented a unified, reproducible benchmark for frame-level deepfake detection by reimplementing and evaluating four CNN-based detectors—Xception, Patch-ResNet, EfficientNetB0, and a custom MesoNet—under exactly the same data preparation, training, and testing conditions. Using a balanced set of real and autoencoder-swapped fake videos, we extracted up to ten

Haar-detected faces per clip and standardized inputs (299×299 for Xception/ResNet/EfficientNet and 256×256 for MesoNet). All models were fine-tuned with identical hyperparameters (Adam, lr=2×10, binary cross-entropy, two epochs, batch size 16) on an 80/20 stratified split. We report frame-level AUC and accuracy, video-level AUC via mean-probability aggregation, ROC curves, confusion matrices, parameter counts, and per-frame inference latency. Our results highlight trade-offs between detection accuracy and computational efficiency, revealing that no single model dominates across all metrics. By releasing our code, data splits, and evaluation scripts, we provide a transparent foundation to drive fair comparisons and future advances in deepfake detection.