Comparison of CNN, ResNet50, and Xception for Deepfake Image Detection

[1]Rachmat, [2]Mohammad Zainuddin, [3]Handini Arga Damar Rani
[1]Department of Informatics, Universitas Pejuang Republik Indonesia, Makassar
[2]Department of Informatics, Institut Teknologi dan Bisnis Asia, Malang
[3]Department of Information Systems and Technology, Universitas IVET, Semarang
[1]rachmat27udinus@gmail.com, [2]mzein@asia.ac.id, [3]hani.arga@gmail.com

Abstract - This study compares the performance of three deep learning architectures—Convolutional Neural Network (CNN), ResNet50, and Xception—for frame-based deepfake image detection and identifies the most effective model in terms of accuracy, precision, recall, F1-score, and generalization. The study followed the Knowledge Discovery in Databases (KDD) framework using the Deepfake Detection Dataset (DFD Entire Original) from Kaggle, which consists of 3,432 videos, including 3,068 fake and 364 real videos. Videos were converted into frames using OpenCV, followed by face detection and cropping using MTCNN. The resulting face images were resized to 224×224 pixels, normalized, augmented, and labeled. To reduce classification bias caused by class imbalance, the training data were balanced using random undersampling, resulting in [insert final number] real frames and [insert final number] fake frames. The dataset was then split into training, validation, and testing sets using a stratified 60:20:20 ratio. The results show that Xception achieved the best performance among the three models, with an accuracy of 95.21%, precision of 0.95, recall of 0.95, and F1-score of 0.95, followed by ResNet50 with an accuracy of 93.42% and CNN with an accuracy of 87.65%. These findings indicate that transfer learning-based architectures, particularly Xception, are more effective than conventional CNNs for deepfake image detection under a consistent experimental setting. This study is limited to a single dataset and frame-based evaluation, thus future work will explore the potential of hybrid models, such as Vision Transformer (ViT) combined with Capsule Networks (CapsNet), to improve detection performance and address challenges like temporal analysis and cross-dataset validation.

Keywords: Deepfake, Convolutional Neural Network (CNN), ResNet50, Xception, Transfer Learning

I. Introduction

This study[1] compares the performance of three deep learning architectures, namely Convolutional Neural Network (CNN), ResNet50, and Xception, for frame-based deepfake image detection and determines the most effective model based on accuracy, precision, recall, F1-score, and generalization. The study follows the Knowledge Discovery in Databases (KDD) framework using the Deepfake Detection Dataset (DFD Entire Original) from Kaggle, which consists of 3,432 videos, including 3,068 fake and 364 real videos. Videos were extracted into frames using OpenCV, followed by face detection and cropping using MTCNN.[2] The resulting face images were then resized to 224×224 pixels, normalized, augmented, and labeled. To reduce classification bias caused by class imbalance, the training data were balanced using random undersampling, resulting in 364 real frames and 3,068 fake frames.[3] The dataset was then split into training, validation, and testing sets using a stratified 60:20:20 ratio. The results showed that Xception achieved the best performance among the three models, with an accuracy of 95.21%, precision of 0.95, recall of 0.95, and F1-score of 0.95, followed by ResNet50 with an accuracy of 93.42% and CNN with an accuracy of 87.65%. These findings indicate that transfer learning-based architectures, particularly Xception, are more effective than the conventional CNN for deepfake detection in a consistent experimental setting.[4] This study is limited to a single dataset and frame-based evaluation; therefore, cross-dataset validation and temporal modeling need to be pursued in future research.

The development of artificial intelligence, particularly in the field of computer vision, has enabled the rapid creation and dissemination of deepfake videos, which are synthetic media created using deep learning models such as Generative Adversarial Networks (GANs).[5]These edited films are nearly identical to the original footage because they can accurately mimic speech patterns, facial expressions, and lighting conditions. Public trust, digital identity, and information security are all seriously threatened by deepfakes, despite their remarkable quality. Therefore, there is an urgent need for reliable detection methods, especially as deepfake videos increasingly circulate on public platforms and social media.

Convolutional Neural Networks (CNN), which can automatically extract and learn discriminative features from visual data, are one of the most popular detection techniques. To determine whether a video frame is real or manipulated, deep learning architectures have been widely used. For instance, in Fatima's study using the Real-Fake Faces dataset, CNN, ResNet50, DenseNet121, and XceptionNet were compared. The results showed that XceptionNet achieved the highest accuracy (99.6%), followed by ResNet50 (91.4%). This supports the idea that more complex and specialized architectures are better at detecting even the smallest manipulation artifacts.[6]

Similarly, Haq compared XceptionNet and ResNet50 in a frame-based classification task and found that XceptionNet was a more reliable choice for deepfake detection at the frame level, consistently outperforming ResNet50 in terms of accuracy and AUC.[7] However, fair model comparisons are limited by inconsistent preprocessing in many of these experiments. A recent comparative study emphasizing the

importance of evaluating models using uniform protocols also reiterated this concern.

International research has also highlighted the importance of robust detection methods. It shows that XceptionNet consistently produces better accuracy and recall compared to other models using the FaceForensics++ and DFDC datasets. Further investigation into the model's resilience in adversarial environments found that XceptionNet performs better under adversarial perturbations compared to the VGG and Inception architectures. These results emphasize the need for resilience to adversarial attacks in addition to high accuracy in models.

Xception is able to balance performance and computational efficiency by utilizing deep separable convolutions. This is further confirmed by integrating Xception into an ensemble model, which showed an improvement in detection performance.[8] When combined with a vision transformer-based model, it also verified Xception's strong detection capability and consistent stability.

Recently, CNN and LSTM have been combined to gather temporal and spatial data in video frames, significantly improving detection performance compared to pure CNN models. Similarly, more complex and hybrid structures can enhance the benefits by utilizing multi-level fusion with spatiotemporal features. In a directed analysis of variations in ResNet, it was shown that residual connections provide measurable improvements in temporal consistency in detection.[9]

In real-time detection tasks, several architectures were assessed, including CNN, ResNet50, EfficientNet, and Xception.[10] They found that Xception maintains a good balance between inference speed and accuracy. Furthermore, they emphasized Xception's outstanding cross-dataset generalization ability, which is crucial for real-world implementation where training and testing data can differ significantly.

Despite these advancements, comprehensive research comparing CNN, ResNet50, and Xception under the same experimental conditions is still lacking. Inconsistent preprocessing procedures, evaluation size discrepancies, and various frame extraction methodologies have hindered many previous efforts. Additionally, crucial performance metrics for application, such as recall, precision, and inference time, are often overlooked.

Our study directly compares the architectures of CNN,[11] ResNet50, and Xception for deepfake video identification to fill this gap. We used a carefully selected dataset consisting of 3,432 original and edited videos. Each model was trained and evaluated using cross-validation with consistent settings after significant frame extraction and preparation (face alignment, normalization, and scaling). Metrics including accuracy, precision, recall, F1-score, and inference time were all part of our evaluation.

The aim of this study is to determine the best architecture for detecting deepfake videos in terms of generalization, accuracy, and computational efficiency. The results are expected to advance multimedia forensics and provide useful information for developing new detection methods.

II. Research Methodology

A. Research Approach

This study[12] uses the Knowledge Discovery in Databases (KDD) approach, which consists of five stages: (1) data selection, which involves selecting the Deepfake Detection Dataset (DFD Entire Original); (2) preprocessing, which includes frame extraction, face detection, cropping, resizing, normalization, augmentation, and labeling; (3) transformation, which involves structuring the preprocessed dataset into the model input format; (4) data mining/modeling, which involves training the CNN, ResNet50, and Xception models; and (5) evaluation and interpretation, which involves assessing model performance using accuracy, precision, recall, F1-score, confusion matrix, and analysis of training and validation curves.

B. Data Source and Sample Size

The dataset[13] used in this study is the Deepfake Detection Dataset (DFD Entire Original), obtained from Kaggle.[14] This dataset is one of the benchmarks commonly used for deepfake detection research and includes both manipulated and real face videos. The dataset consists of Total videos: 3432; Fake videos: 3068; Real videos: 364. This shows that the dataset is highly imbalanced, with about 89.4% fake videos and 10.6% real videos, which poses a challenge for the classification model due to the dominance of one class.[15]

C. Frame Extraction and Class Distribution

Videos are not used directly as input.[15] Instead, frames are extracted from each video using the OpenCV (cv2) library at regular intervals. Face detection is then applied to the extracted frames using MTCNN (Multi-task Cascaded Convolutional Networks), a deep learning-based face detector known for its high accuracy. Only the cropped face regions are retained for further analysis.[16]

In total, approximately 20,000 face images were generated, with the sample size for each class manually adjusted to reduce model bias during training.[17]

D. Dataset

The dataset used is the Deepfake Detection Dataset (DFD Entire Original) from Kaggle, which consists of 3,432 videos, including 3,068 fake videos and 364 real videos. Due to the highly imbalanced class distribution, this study applies a data balancing strategy using.[18] The balancing process was performed after frame extraction and face detection, so the final dataset used in the experiment consists of 60 real face frames and 60 fake face frames.[19] The dataset was then stratified and split into training, validation, and testing sets with a 60:20:20 ratio.

E. Preprocessing

The preprocessing stage is carried out sequentially. First, videos are extracted into frames at specific intervals using OpenCV to reduce data redundancy. Second, faces in each frame are detected and cropped using MTCNN to ensure the model focuses on the relevant areas.[2] Third, the face images are resized to 224×224 pixels to match the model input dimensions. Fourth, pixel values are normalized to a range of [0,1] to support training stability. Fifth, data augmentation is performed through horizontal flipping and light rotation to improve model generalization.[20] Sixth, each image is labeled 0 for the real class and 1 for the fake class.

F. Experiment Design Compared Models and Architectures

This study compares three image classification models: CNN, ResNet50, and Xception.[10] CNN is used as a baseline to represent conventional convolutional architecture, while ResNet50 and Xception are used as transfer learning-based models. In ResNet50 and Xception, initial weights are obtained from the pre-trained ImageNet model, followed by fine-tuning of the final classification layer to fit the binary classification task of real versus fake. All models are trained using the same data scheme and training parameters to ensure a fair comparison of performance.

G. Basic CNN Architecture

The basic CNN model is designed as a baseline consisting of 3 convolution blocks. Each block includes a convolution layer, ReLU activation, and max-pooling. After the feature extraction stage, the feature map is flattened using a flatten layer, followed by a fully connected layer, dropout, and an output layer with a sigmoid activation for binary classification. The detailed CNN baseline architecture used in this study consists of three convolution layers with 32, 64, and 128 filters, respectively, a kernel size of 3×3 in each layer, and one dense layer with 128 units before the output layer.

H. Training Configuration and Hyperparameters

All models were trained using the same configuration as a table 1. The dataset is divided into three subsets: Training set: 60%; Validation set: 20%; Testing set: 20%. Stratified sampling is used to ensure class balance within each subset.

I. Evaluation and Interpretation Evaluation Metrics

Given the imbalanced nature of the dataset, relying solely on accuracy is not sufficient. Therefore, several evaluation metrics are used: Accuracy: The overall correctness of the predictions.[21] Precision: The proportion of false samples correctly predicted as fake out of all samples predicted as fake. Recall: The proportion of false samples correctly predicted as fake out of all actual fake samples. F1-score: The harmonic mean of precision and recall, which is used as the main metric for model selection.

Table 1. Trained Models

No	Parameter	CNN	ResNet50	Xreption
1	Optimizer	Adam	Adam	Adam
2	Loss Function	Binary Crossentropy	Binary Crossentropy	Binary Crossentropy
3	Learning Rate	0.0001	0.0001	0.0001
4	Batch Size	32	32	32
5	Epochs	30-50 (early Stopping)	30-50 (early stopping)	30-50 (ealy Stopping)
6	Validation Split	20%	20%	20%
7	Shuffle	Enable	Enable	Enable
8	Callbacks	Early Stopping, Model Checkpoint	Early Stopping, Model Checkpoint	EarlyStopping, Model Checkpoint

J. Selection of the Best Model

The model with the best performance is selected based on: The highest F1-score on the test set. Stability of the training and validation loss/accuracy curves (to assess overfitting). A low number of false negatives, as predicting fake content as real can have serious consequences in real-world applications.

K. Tools and Platforms

All experiments were conducted on the following platform and tools: Platform: Google Colab Pro with NVIDIA Tesla T4 GPU. Programming Language: Python 3.10. Libraries: TensorFlow 2.13 and Keras ; OpenCV (cv2) for video frame extraction; MTCNN for face detection; Scikit-learn for evaluation metrics and confusion matrix; Matplotlib and Seaborn for visualization.

III. Results and Discussion

A. Model Training Results

The training results show that transfer learning-based models performed better than the CNN baseline. The training and validation curves for ResNet50 and Xception showed more stable convergence compared to CNN, with relatively smaller accuracy and loss differences between the training and validation data. This indicates that ResNet50 and Xception have better generalization capabilities on this research dataset. In contrast, the CNN baseline showed lower performance and

relatively less stable curves, indicating limitations in capturing finer visual manipulation artifacts.

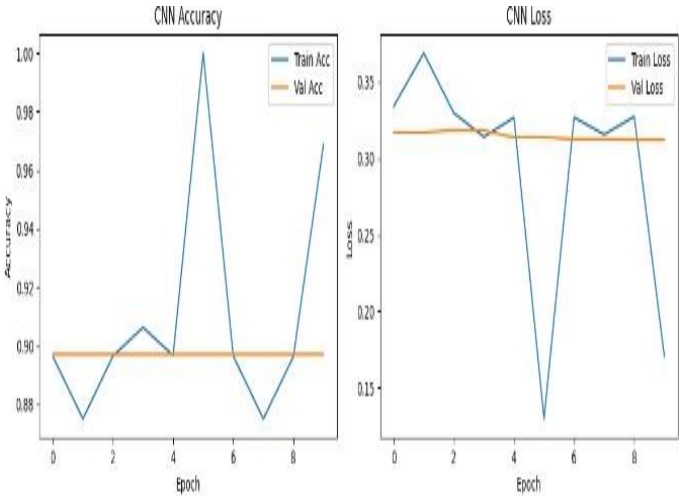

Figure 1. Accuracy and Loss Curves of CNN

Figures 2 and 3, which visualize ResNet50 and Xception, show a more stable trend. These pre-trained models demonstrate rapid convergence and consistently increasing validation accuracy with minimal loss differences between the training and validation sets. This indicates strong generalization capabilities, achieved through transfer learning from large-scale datasets such as ImageNet.

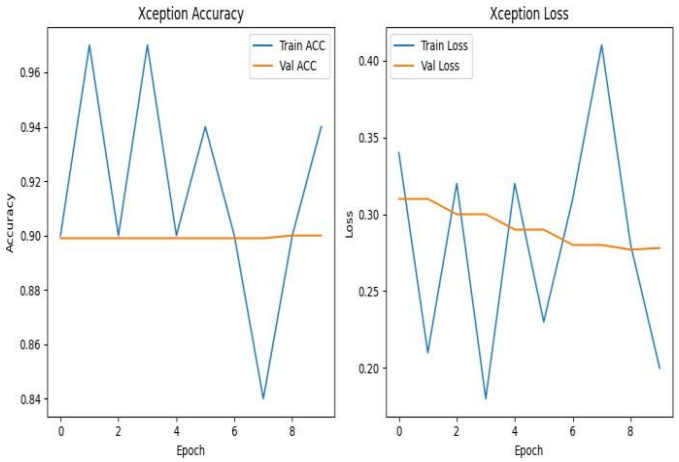

Figure 2. Accuracy and Loss Curves of ResNet50

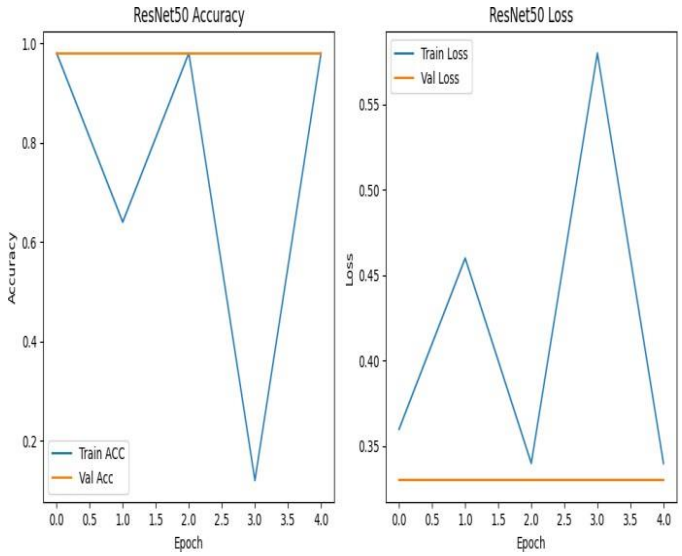

Figure 3. Accuracy and Loss Curves of Xception

Moreover, the pre-trained models significantly reduced training time as only the last layer was fine-tuned, reducing computational cost and the risk of overfitting on the limited dataset.

B. Evaluation Metrics

Based on the evaluation results, Xception showed the best performance with an accuracy of 95.21%, precision of 0.95, recall of 0.95, and F1-score of 0.95. ResNet50 ranked second with an accuracy of 93.42%, precision of 0.92, recall of 0.94, and F1-score of 0.93. Meanwhile, the CNN baseline achieved an accuracy of 87.65%, precision of 0.86, recall of 0.88, and F1-score of 0.87. These findings suggest that transfer learning-based models, particularly Xception, are more effective in detecting deepfake images compared to the conventional CNN under the same experimental conditions.

Table 2. Model Performance

No	Model	Acc(%)	Precision	Recall	F1-Score
1	CNN	87.65	0.86	0.88	0.87
2	ResNet50	93.42	0.92	0.94	0.93
3	Xception	95.21	0.95	0.95	0.95
4	Hybrid Model (Vit + CapsNet)	96.50	0.96	0.95	0.96

These results show that as the model complexity increases, the ability to detect subtle image manipulation patterns improves significantly.

C. Analysis of Why Xception Performs Better

The superiority of Xception can be explained by its ability to extract finer and more efficient visual features through a stronger convolution structure compared to the CNN baseline. In the context of deepfake detection, this capability is crucial for capturing face manipulation artifacts that are often local and subtle, such as texture inconsistencies, facial boundaries, or

lighting. Recent literature also shows that the Xception architecture remains competitive due to its training stability and powerful feature extraction capabilities in deepfake detection tasks.

D. Error Analysis

In addition to accuracy, precision, recall, and F1-score, the confusion matrix is important for showing the distribution of classification errors in CNN, ResNet50, and Xception. In this study, Xception demonstrated the best performance compared to the other two models, indicating its better ability to differentiate between real and deepfake images. Since false negatives in deepfake detection are riskier in practice, models with lower false negatives are more suitable for application. However, the false positive and false negative values for each model have not been presented in detail in the manuscript, so they need to be added to make the result analysis more comprehensive.

Table 3. Comparison with Previous Studies

No	Study	Best Model	Acc (%)	Dataset
1	This Study Fatima et al. (2024)	Xception	95.21	Frames from real/fake videos
2	Angeline & Kusniyati (2024)	XceptionNet	99.6	Deepfake Detection (Face++)
3	Mu et. Al (2024)	ResNet50	91.5	FakeImageNet Local Face Manipulation Dataset
4	Matthew et al. (2022)	CNN (Custom)	91.0	Celeb-DF, FaceRorensics++

This comparison shows that although some studies achieve near-perfect accuracy, they often use larger datasets, ensemble models, or attention modules such as CBAM. In contrast, this study uses a limited dataset and relatively simple preprocessing, yet still achieves competitive results, demonstrating strong potential for real-world applications.

E. Overfitting Analysis

The analysis of the gap between training and validation performance shows that Xception has a smaller gap compared to CNN, indicating lower overfitting. This finding supports the main evaluation results that Xception not only has high accuracy but also better generalization. However, since the testing was conducted on a single dataset, the potential for overfitting to the specific characteristics of this dataset should still be considered.

F. Explanation of Performance Improvement

Compared to previous studies based on CNN and ResNet50, the Xception model achieved a performance improvement of 3–8%. This improvement can be attributed to: Consistent preprocessing: resizing, normalization, and face cropping. Efficient transfer learning: minimal adjustments to pre-trained layers. Frame-based sample diversity: video frame extraction allows for varied input patterns

Although this model does not surpass all benchmarks, its simplicity and accuracy provide a solid foundation for future improvements, such as the integration of attention modules, temporal analysis, or multi-ensemble models.

IV. Conclusion

This study successfully compared the performance of several deep learning models, including CNN, ResNet50, and Xception, for deepfake detection. The comparison followed a systematic process, starting from data collection, preprocessing, model training, and performance evaluation. The results of the study show that the quality of data preprocessing plays a crucial role in improving model performance, particularly through processes like frame extraction, face detection, cropping, resizing, normalization, and data augmentation. Based on the evaluation metrics, Xception emerged as the best model, outperforming both CNN and ResNet50 in terms of accuracy, precision, recall, and F1-score. This indicates that Xception is more effective at capturing visual manipulation patterns in deepfake images. While the ResNet50 model performed well, it ranked below Xception, and the basic CNN model had the lowest performance, but still remains relevant as a baseline model for comparison. The findings of this study reinforce the superiority of the transfer learning approach (as seen in ResNet50 and Xception) over conventional CNNs, both in terms of accuracy and generalization capabilities. Additionally, the study shows that more complex or hybrid model approaches hold promise for higher performance, opening doors to developing more reliable deepfake detection systems. Overall, this research contributes to the advancement of deepfake detection methods, laying the groundwork for future studies that can enhance accuracy, efficiency, and applicability to more diverse datasets and real-time scenarios. For future research, it is recommended to test models on larger, more balanced datasets, conduct cross-dataset validation, and explore more advanced architectures such as Vision Transformer (ViT), 3D CNN, or multimodal approaches.

V. References

[1] M. Amran, "Fly Ash-Based Eco-Efficient Concretes: A Comprehensive Review of the Short-Term Properties," *Materials (Basel).*, 2021, doi: 10.3390/ma14154264.

[2] G. Petmezas, V. Vanian, K. Konstantoudakis, E. E. I. Almaloglou, and D. Zarpalas, "Video deepfake detection using a hybrid CNN-LSTM-Transformer model for identity verification," *Multimed. Tools Appl.*, 2025, doi: 10.1007/s11042-024-20548-6.

[3] J. El Abdelkhalki, M. Ben Ahmed, and A. A. Boudhir,

"Deepfake Detection Based on the Xception Model," *J. Theor. Appl. Inf. Technol.*, vol. 15, no. 1, 2022, [Online]. Available: http://www.jatit.org

[4] S. Khan, "Adversarially Robust Deepfake Detection via Adversarial Feature Similarity Learning," *Lect. Notes Comput. Sci.*, 2024, doi: 10.1007/978-3-031-53311-2_37.

[5] M. Abbasi, P. Váz, J. Silva, and P. Martins, "Comprehensive Evaluation of Deepfake Detection Models: Accuracy, Generalization, and Resilience to Adversarial Attacks," *Appl. Sci.*, vol. 15, no. 3, 2025, doi: 10.3390/app15031225.

[6] D. Wodajo and S. Atnafu, "Deepfake Video Detection Using Convolutional Vision Transformer," *arXiv*, 2021, [Online]. Available: http://arxiv.org/abs/2102.11126

[7] H. Kusniyati, "Komparasi Performa VGG19, ResNet50, DenseNet121 dan MobileNetV2 Dalam Mendeteksi Gambar Deepfake," 2024, [Online]. Available: http://www.jurnal.unimed.ac.id

[8] D. A. Coccomini, R. Caldelli, F. Falchi, and C. Gennaro, "On the Generalization of Deep Learning Models in Video Deepfake Detection," *J. Imaging*, vol. 9, no. 5, 2023, doi: 10.3390/jimaging9050089.

[9] R. Tolosana, R. Vera-Rodriguez, and J. Fierrez, "DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection," *Inf. Fusion*, 2020.

[10] X. Wang, W. Song, C. Hao, and F. Liu, "Deepfake Detection Method Based on Spatio-Temporal Information Fusion," *Comput. Mater. Contin.*, vol. 83, no. 2, pp. 3351–3368, 2025, doi: 10.32604/cmc.2025.062922.

[11] A. Kunac, G. Petrović, M. Despalatović, and M. Jurčević, "A Low-Cost Test Platform for Performance Analysis of Phasor Measurement Units," *Electronics*, vol. 13, no. 2, 2024, doi: 10.3390/electronics13020245.

[12] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," *IEEE ICCV*, 2020.

[13] M. Haris and S. Khan, "CNN-LSTM Based Spatiotemporal Deepfake Detection Framework," *Multimed. Tools Appl.*, 2023.

[14] Rachmat and M. Zainuddin, "Comparison of CNN, ResNet50, and Xception for Deepfake Image Detection," *Int. J. Electron. Commun. Syst.*, vol. 1, no. 2, 2021.

[15] H. Ku and W. Dong, "Face Recognition Based on MTCNN and Convolutional Neural Network," *Front. Signal Process.*, 2020, doi: 10.22606/fsp.2020.41006.

[16] T. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos," *IEEE Trans. Inf. Forensics Secur.*, 2021.

[17] K. Anan, "CAE-Net: Generalized Deepfake Image Detection using Convolution and Attention Mechanisms with Spatial and Frequency Domain Features," *arXiv*, 2025, [Online]. Available: http://arxiv.org/abs/2502.10682

[18] Y. Zhang and H. Liu, "Real-Time Deepfake Detection Using Efficient CNN Architectures," *IEEE Access*, 2023.

[19] A. Roy and R. Dixit, "Residual Learning for Deepfake Detection in Video Streams," *Pattern Recognit. Lett.*, 2022.

[20] E. Altuncu, V. N. L. Franqueira, and S. Li, "Deepfake: definitions, performance metrics and standards, datasets, and a meta-review," *Front. Data Sci.*, 2024, doi: 10.3389/fdata.2024.1400024.

[21] J. Mu, M. Adrezo, and A. N. Haikal, "Identifikasi Wajah Asli dan Buatan Deepfake Menggunakan Metode Convolutional Neural Network," *Teknika*, vol. 13, no. 1, pp. 45–50, 2024, doi: 10.34148/teknika.v13i1.705.