Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 3338~3346
e-ISSN: 2548-6861

Facial Deepfake Detection System Using YOLOv11 and Xception Architecture

Fachril Akbar [1]*, Nurdin [2], Kurniawati [3]***
* Department of Informatics, Universitas Malikussaleh, Lhokseumawe, Indonesia
** Department of Information Technology, Universitas Malikussaleh, Lhokseumawe, Indonesia
fachril.220170072@mhs.unimal.ac.id [1], nurdin@unimal.ac.id[2], kuriawati@unimal.ac.id [3]

Article Info

Article history:

Received 2026-06-25
Revised 2026-07-24
Accepted 2026-08-08

Keyword:

Computer Vision,
Deepfake,
TensorFlow,
Xception,
YOLOv11

ABSTRACT

The rapid development of artificial intelligence has led to the emergence of deepfakes, which pose serious threats to information security and public trust in digital media. This study develops a facial deepfake detection system that integrates YOLOv11 for face detection and the Xception architecture for classifying real and manipulated faces. YOLOv11 successfully localized all facial regions in the tested dataset with high confidence scores. The Xception model achieved a testing accuracy of 90.10%, with a Recall of 97.11% for the Fake class and an AUC of 0.98. Visual explanation using Grad-CAM showed that the model focused on critical areas such as the forehead, temples, and face boundaries to detect manipulation artifacts. The system was implemented as a desktop application named "Snap Detector" and passed black-box testing. However, the average processing speed of 6.26 FPS on an NVIDIA T4 GPU indicates that further optimization is needed for real-time performance.

I. INTRODUCTION

The rapid advancement of artificial intelligence (AI) has enabled the creation of highly realistic synthetic media known as deepfakes. Deepfake technology can manipulate facial appearances, expressions, and speech with remarkable realism, making it increasingly difficult for humans to distinguish between authentic and manipulated content. Such developments pose serious threats to information integrity, public trust, privacy, and cybersecurity because manipulated content can be used for misinformation campaigns, identity fraud, and digital crimes [1], [2]. Furthermore, the increasing sophistication of generative AI technologies has accelerated the dissemination of deepfake content across social media platforms and digital communication channels, creating new challenges for digital forensic systems and content verification mechanisms [3].

In Indonesia, the threat of deepfake technology has become increasingly significant alongside the rapid growth of internet users and digital media consumption. Recent reports indicate that deepfake-based fraud cases increased dramatically, with Indonesia experiencing a 1,550% increase in deepfake fraud incidents between 2022 and 2023 [4]. In addition, Indonesian researchers have highlighted that deepfake technology poses serious risks to biometric security systems, privacy protection, and digital identity verification processes. The widespread use of social media platforms further amplifies the potential impact of manipulated content, making automated deepfake detection an urgent necessity for maintaining trust in digital information ecosystems [5].

Recent studies conducted in Indonesia have also emphasized the importance of developing automated deepfake detection systems. Raharjo et al. reported that Convolutional Neural Networks (CNNs) are effective in identifying texture inconsistencies and facial manipulation artifacts that are difficult to detect through visual inspection alone [6], [7]. Similarly, Gulo et al. demonstrated that CNN-based approaches can effectively identify manipulated facial images by learning distinctive visual patterns associated with deepfake generation processes [8]. These findings indicate that deep learning remains one of the most promising approaches for combating increasingly sophisticated deepfake attacks.

Although many deepfake detection approaches have achieved promising classification performance, accurate localization of facial regions remains a critical prerequisite.

Deepfake artifacts are primarily concentrated in facial areas; therefore, robust face detection must be performed before classification. Recent Indonesian research utilizing YOLO-based architectures demonstrated that object detection models can effectively identify facial regions in real-time environments while maintaining high detection accuracy [9] [10]. Consequently, integrating YOLOv11 for face localization and Xception for deepfake classification offers a promising solution for developing an efficient and reliable deepfake detection framework.

II. METHODOLOGY

This study proposes a deepfake face detection framework that combines YOLOv11 for face localization and Xception for deepfake classification. The proposed methodology consists of four main stages: dataset preparation, image preprocessing, face detection, and deepfake classification. To improve classification performance and model generalization, a two-phase transfer learning strategy was adopted during the training process. The overall workflow of the proposed system is illustrated in Figure 1.

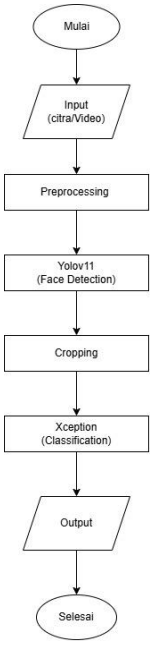

Figure 1 Research Stage Flowchart

A. Research Dataset

This study employed a facial image dataset consisting of 12,344 images collected from multiple sources. The primary dataset was obtained from the publicly available Deepfake Clean Final dataset available on Kaggle. To improve data diversity and simulate practical scenarios, additional authentic facial images were collected from self-captured photographs, while additional manipulated facial images were collected from publicly available TikTok content containing deepfake-generated faces.

The final dataset comprised 6,523 Real images and 5,821 Fake images. Since the manipulated images originated from publicly available online content, the specific deepfake generation methods used to create these images were not available. Nevertheless, the collected images represent diverse facial appearances, illumination conditions, facial poses, and image qualities, enabling the proposed model to learn robust visual representations for binary deepfake classification.

Before training, all images were manually organized into the Real and Fake classes, resized according to the Xception input requirements, and normalized to improve training stability. The dataset was divided using a hold-out strategy into 70% training data, 15% validation data, and 15% testing data to ensure an objective evaluation of model performance while reducing the risk of overfitting.[11].

B. Research Procedure

The research workflow consisted of several sequential stages, starting from dataset preparation and ending with model evaluation. The overall procedure is illustrated in Figure 2.

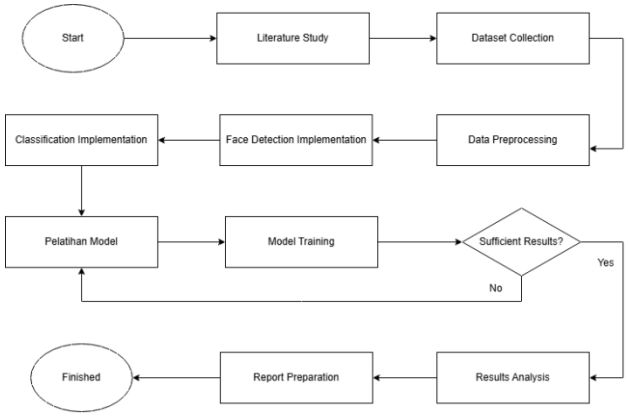

Figure 2 System Schematic

1. Literature Review
   Relevant studies concerning deepfake detection, facial recognition, YOLO-based object detection, and Xception-based image classification were reviewed to establish the theoretical foundation of the study.
2. Dataset Collection and Preparation
   Real and fake facial images were collected and organized into training, validation, and testing subsets.
3. Data Preprocessing
   Image preprocessing included resizing, pixel normalization, and data augmentation. These operations were performed to improve model generalization and reduce overfitting.
4. Face Detection
   The YOLOv11 model was employed to detect facial regions from input images and generate bounding boxes around detected faces.

5. Face Cropping and Extraction
   Detected facial regions were cropped and used as input for the classification stage.
6. Deepfake Classification
   The cropped facial images were classified as either real or fake using the Xception architecture.
7. Model Evaluation
   The proposed system was evaluated using Accuracy, Precision, Recall, F1-Score, Receiver Operating Characteristic (ROC) Curve, and Area Under Curve (AUC).

C. System Architecture

The proposed deepfake detection framework integrates YOLOv11 and Xception into a unified pipeline. The system operates in two sequential stages. First, YOLOv11 is employed to localize facial regions from the input image. Subsequently, the detected facial regions are cropped and forwarded to the Xception network for deepfake classification. This design enables the model to focus exclusively on facial information while reducing the influence of irrelevant background features.

1) YOLOv11-Based Face Detection

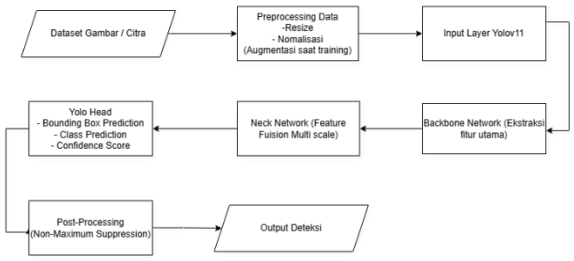

Figure 3 YOLOv11 system schematic

The first stage of the proposed framework utilizes YOLOv11 as a face detector. The primary objective of this module is to identify and localize facial regions accurately before the classification process is performed. Figure 3 illustrates the architecture of the YOLOv11 face detection module.

The detection process begins with image preprocessing, including resizing and normalization, to ensure compatibility with the input dimensions required by the network. The preprocessed image is then passed through the YOLOv11 architecture [12], which consists of three main components: Backbone, Neck, and Detection Head [13].

The Backbone Network is responsible for extracting hierarchical visual features from the input image. Through multiple convolutional operations, the network learns low-level features such as edges, contours, and textures, as well as higher-level semantic representations associated with facial structures.

The extracted features are subsequently processed by the Neck Network, which performs multi-scale feature fusion [14]. This mechanism combines information obtained from different feature levels, allowing the detector to recognize faces with varying sizes, poses, and illumination conditions. The integration of multi-scale information improves localization performance, particularly in challenging scenarios involving small or partially occluded faces [15], [16].

The final component, known as the Detection Head, generates bounding box coordinates, confidence scores, and class predictions simultaneously. Non-Maximum Suppression (NMS) is then applied to eliminate redundant detections and retain only the most reliable facial bounding boxes. The resulting facial regions are cropped and forwarded to the deepfake classification stage.

The adoption of YOLOv11 offers several advantages, including fast inference speed, efficient computation, and robust face localization performance, making it suitable for real-time deepfake detection applications [17], [18].

2) Xception-Based Deepfake Classification

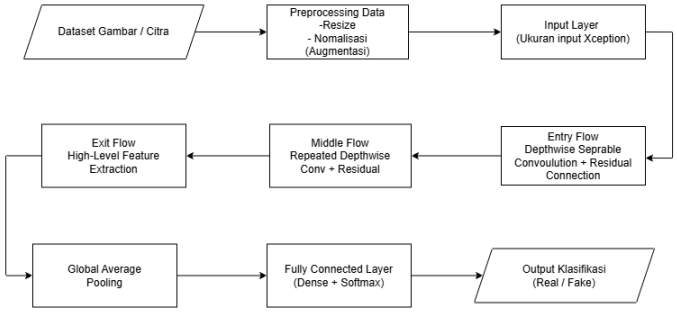

Figure 4 Xception system schematic

Following the face detection stage, the cropped facial images are processed by the Xception network to determine whether the face is authentic or manipulated. Figure 4 presents the architecture of the Xception-based classification model employed in this study.

Xception is a convolutional neural network architecture that utilizes depthwise separable convolution to improve computational efficiency while maintaining strong feature extraction capability [11]. Compared with conventional convolutional operations, depthwise separable convolution decomposes feature extraction into spatial and channel-wise operations, significantly reducing the number of trainable parameters [11].

The architecture consists of three primary stages: Entry Flow, Middle Flow, and Exit Flow.

The Entry Flow serves as the initial feature extraction stage. In this phase, the input facial image is transformed into a set of feature maps that capture fundamental visual characteristics, including facial contours, texture patterns, and illumination information.

The extracted features are then forwarded to the Middle Flow, which contains multiple residual blocks based on depthwise separable convolutions. This stage is responsible for learning more discriminative and abstract representations

associated with deepfake artifacts. During this process, the network captures subtle inconsistencies introduced by face manipulation techniques, such as abnormal skin textures, blending artifacts, unnatural facial boundaries, and inconsistencies in illumination and color distribution [19].

Finally, the Exit Flow further refines the extracted representations and produces high-level feature maps suitable for classification. Global Average Pooling is subsequently applied to reduce feature dimensionality while preserving relevant information. The resulting feature vector is passed through fully connected layers and a sigmoid activation function to generate the final classification probability.

The final output is a binary prediction indicating whether the analyzed facial image belongs to the Real or Fake class. By leveraging transfer learning and fine-tuning strategies, the Xception model is capable of learning robust visual representations that effectively distinguish authentic faces from deepfake-generated content [19], [20].

D. Transfer Learning Configuration

The Xception classifier was trained using a two-phase transfer learning strategy to improve convergence stability and enhance feature representation learning for deepfake detection [18]. This approach consists of two sequential training stages: Head-Only Training and Fine-Tuning. The training configuration adopted in both phases is presented in Table I.

TABLE I
TRAINING CONFIGURATION FOR THE TWO-PHASE LEARNING STRATEGY

Parameter	**Fase 1 (*Head* Only)**	**Fase 2 (Fine-Tuning)**
Model Architecture	Xception (pretrained ImageNet)	Xception (pretrained ImageNet)
Input Size	$299 \times 299 \times 3$ pixels	$299 \times 299 \times 3$ pixels
Batch size	5	5
Epochs	1 – 10	11 – 20
Optimizer	Adam	Adam
Learning Rate	1×10^{-4}	1×10^{-5}
Trained Layers	Classification head only (base frozen)	Base layers + head (unfrozen)
Loss Function	Binary Crossentropy	Binary Crossentropy
Monitored Metrics	AUC, Accuracy	AUC, Accuracy
Augmentation	RandomFlip, RandomRotation, RandomZoom, RandomBrightness, RandomContrast	RandomFlip, RandomRotation, RandomZoom, RandomBrightness, RandomContrast
Environment	Google Colab GPU NVIDIA T4	Google Colab GPU NVIDIA T4

As shown in Table I, both training phases utilize the same pretrained Xception architecture and input image dimensions. The distinction between the two phases lies primarily in the trainable layers and optimization objectives.

1) Phase I: Head-Only Training

During the first phase, all convolutional layers of the pretrained Xception backbone were frozen, while only the newly added classification head was trained. This strategy preserves the generic visual features learned from the ImageNet dataset and enables the classifier to adapt to the binary deepfake detection task without disrupting previously learned representations [18].

The classification head consisted of a Global Average Pooling layer, a fully connected layer, a Dropout layer, and a sigmoid output layer Binary Cross-Entropy was employed as the loss function, while the Adam optimizer was used for parameter optimization.

As shown in Figure 4, freezing the backbone during the initial stage reduces the number of trainable parameters and accelerates convergence. This phase primarily focuses on learning task-specific decision boundaries between authentic and manipulated facial images.

2) Phase II: Fine-Tuning

After the classification head reached satisfactory convergence, the training process proceeded to the fine-tuning stage. During this phase, selected upper layers of the Xception backbone were unfrozen and allowed to update their weights through backpropagation.

As presented in Table I, the fine-tuning process was performed during epochs 11–20 while maintaining the same input size and batch configuration. A lower learning rate was employed to prevent excessive modifications to the pretrained weights and preserve previously learned visual features [19].

The objective of this stage is to enable the network to learn deepfake-specific characteristics directly from facial images. By fine-tuning the higher-level convolutional layers, The model becomes more sensitive to subtle manipulation artifacts such as texture inconsistencies, boundary irregularities, illumination discrepancies, and synthetic generation traces that commonly appear in deepfake content [17].

The combination of Head-Only Training and Fine-Tuning allows the model to achieve a balance between training efficiency and classification performance. This transfer learning strategy also reduces the risk of overfitting while improving the model's ability to generalize to previously unseen facial images.

3) Training Monitoring

During the training process, model performance was continuously monitored using the validation dataset. Validation metrics were evaluated at the end of each epoch to assess the learning progress and generalization capability of the model. Monitoring validation performance enables the identification of potential overfitting or underfitting during both training phases.

To improve training stability, the model parameters were updated using the Adam optimizer and Binary Cross-Entropy loss function. In addition, an early stopping mechanism was employed to terminate training when no significant improvement was observed in the validation performance for a predefined number of epochs. This strategy helps prevent overfitting and ensures that the selected model represents the best balance between learning capacity and generalization performance.

The model achieving the highest validation performance during the fine-tuning stage was selected as the final model for subsequent testing and evaluation.

Unlike previous studies that generally investigate face detection and deepfake classification as separate tasks, this study proposes an integrated framework that combines YOLOv11-based face localization with Xception-based deepfake classification within a unified detection pipeline. The proposed framework employs automatic face extraction prior to classification, enabling the classifier to focus exclusively on facial regions that are more likely to contain manipulation artifacts. Furthermore, the framework incorporates a two-phase transfer learning strategy, Grad-CAM-based visual interpretation, and desktop-based implementation to provide an interpretable and practical end-to-end deepfake detection system. Therefore, the contribution of this study lies not merely in combining two existing models, but in integrating accurate face localization, robust deepfake classification, model interpretability, and practical deployment into a single framework.

III. Result And Discussion

A. YOLOv11 Face Localization Performance

The face detection stage utilizing YOLOv11 demonstrated highly stable performance as the initial component of the proposed framework. Evaluation across the entire dataset of 12,344 images showed that YOLOv11 successfully localized all facial regions with high confidence scores.The model efficiently generated bounding boxes and confidence scores above the predefined threshold for all samples.

As shown in Figure 5, YOLOv11 successfully detected both real and fake faces under various illumination conditions, including normal lighting, low-light, and overexposed environments. Despite significant variations in image brightness, the detector consistently localized facial regions with high accuracy. This robustness indicates that YOLOv11 effectively utilizes facial structural and geometric features, enabling reliable face localization even when visual quality is degraded by challenging lighting conditions.

Figure 5. Representative Face Localization Samples Detected by YOLOv11 Under Different Illumination Conditions. Real and fake facial images captured under normal lighting, low-light, and overexposed conditions were successfully localized by YOLOv11.

Figure 5 Representative YOLOv11 Face Localization Results Under Various Lighting Conditions

B. Xception Model Training Analysis

The classification model was trained using a Two-Phase Training approach over 20 epochs. Phase 1 (epochs 1–10) focused on basic feature extraction by freezing the base layers, while Phase 2 (epochs 11–20) performed fine-tuning on all network parameters.

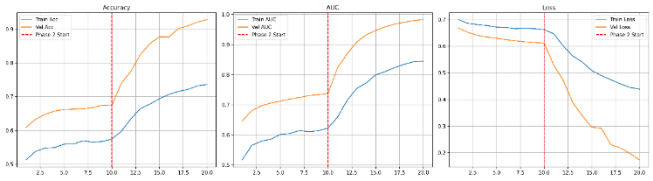

Figure 6 Xception Model Training Curves

The training results exhibited an excellent convergence pattern. Validation accuracy rose sharply from approximately 67% at the end of the first phase to 93%–95% by the final epoch. Simultaneously, the validation loss significantly decreased from 0.61 to 0.18. The AUC reached 0.98, which, according to the interpretation scale, falls into the "Excellent" category. This confirms the model's high discriminatory ability in distinguishing between authentic and manipulated faces.

C. Classification Metrics Evaluation

A final evaluation was conducted using 709 test images. The system achieved an overall testing accuracy of 90.10%. In the context of digital security, the most critical metric is the model's ability to detect manipulated content, represented by the Recall for the Fake class.

As shown in Figure 7, the model achieved a Recall of 97.11% for the Fake class with an F1-score of 91.61%, indicating that the proposed framework successfully detected the vast majority of manipulated facial images in the testing dataset.

Although the proposed model achieved an overall testing accuracy of 90.10%, it obtained a substantially higher Recall of 97.11% for the Fake class. This indicates that the classifier successfully identified almost all manipulated facial images while allowing a relatively higher number of authentic images to be incorrectly classified as fake. Consequently, the improvement in sensitivity toward manipulated content is accompanied by a decrease in specificity, resulting in an overall accuracy that remains around 90%. This behavior is also reflected in the confusion matrix (Figure 7), which shows that the model produces relatively few False Negatives while accepting a higher number of False Positives. This indicates that the classifier prioritizes detecting manipulated images, even at the expense of occasionally misclassifying authentic faces.

This trade-off is desirable in deepfake detection applications because the consequences of False Negatives are considerably more critical than False Positives. A False Negative allows manipulated media to remain undetected and potentially be used for misinformation, identity fraud, or other malicious activities. Conversely, a False Positive only causes an authentic image to be flagged for further verification. Therefore, prioritizing a high Recall for the Fake class represents a more appropriate decision for security-oriented deepfake detection systems.

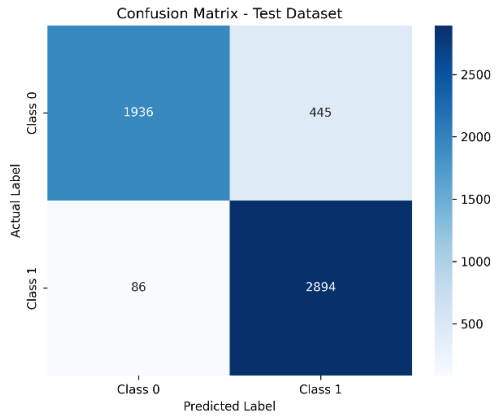

Figure 7 Confusion Matrix

TABLE II
SUMMARY OF THE FINAL EVALUATION METRICS OF THE PROPOSED YOLOv11–XCEPTION FRAMEWORK

Metrik	Fake	Real	Macro Avg	Weighted Avg
Precision	86,70%	95,75%	91,21%	90,70%
Recall	97,11%	81,34%	89,21%	90,10%
F1-score	91,61%	87,96%	89,77%	89,97%
Accuracy				90,10%

Based on the confusion matrix, the proposed framework achieved a Specificity of 81.34% and a Balanced Accuracy of 89.23%. The specificity indicates that the model correctly identified 81.34% of authentic facial images, while the balanced accuracy demonstrates that the proposed framework maintains balanced performance across both the Real and Fake classes despite prioritizing a higher Recall for manipulated images. These complementary metrics further confirm the effectiveness of the proposed framework for security-oriented deepfake detection applications.

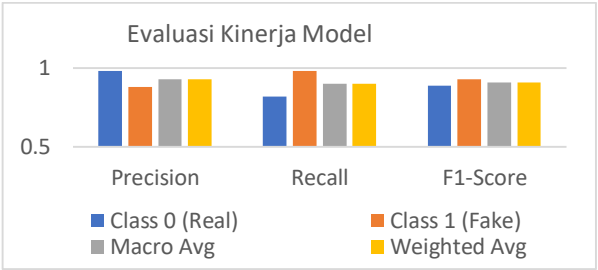

Figure 8 Summary of Final Model Evaluation Metrics

Visual explanation using Grad-CAM demonstrates that the proposed Xception model consistently focuses on the forehead, temples, and facial boundary regions when identifying manipulated images. These regions are commonly affected during deepfake generation because most face-swapping algorithms reconstruct facial components and blend the synthesized face with the original background around the outer facial contour.

The forehead frequently contains subtle inconsistencies in skin texture, illumination, and color distribution caused by imperfect synthesis. Likewise, the temple regions often exhibit slight blending artifacts resulting from the transition between synthesized and authentic facial textures. Meanwhile, the facial boundary represents one of the most informative regions because deepfake generation algorithms commonly introduce boundary discontinuities, misalignment, and unnatural edge transitions during face compositing.

The visualization results indicate that the proposed model learns meaningful manipulation-related features rather than relying on irrelevant background information. This behavior confirms that the integration of YOLOv11 effectively removes distracting background regions, allowing the Xception classifier to concentrate on facial artifacts that are semantically associated with deepfake generation.

D. Grad-CAM Interpretation

To better understand the decision-making process of the proposed model, Grad-CAM was employed to visualize the

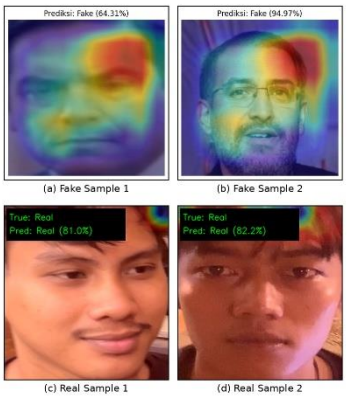

Figure 9 Grad-CAM Visualization of Representative Fake and Real Face Samples

image regions that contributed most to the classification results. Figure 9 presents representative Grad-CAM visualizations for both fake and real facial images correctly classified by the proposed YOLOv11–Xception framework. Warmer colors indicate regions receiving greater attention from the classifier, whereas cooler colors represent areas with lower contributions.

As shown in Figure 9(a) and Figure 9(b), the Grad-CAM visualizations of fake facial images consistently highlight the forehead, temple, and facial boundary regions. These regions are commonly affected during deepfake generation because most face-swapping algorithms reconstruct facial components and blend the synthesized face with the original image around the outer facial contour.

The forehead frequently contains subtle inconsistencies in skin texture, illumination, and color distribution caused by imperfect synthesis. Likewise, the temple regions often exhibit blending artifacts resulting from the transition between synthesized and authentic facial textures. Meanwhile, the facial boundary represents one of the most informative regions because deepfake generation algorithms commonly introduce boundary discontinuities, misalignment, and unnatural edge transitions during face compositing.

In contrast, the Grad-CAM visualizations of real facial images shown in Figure 9(c) and Figure 9(d) exhibit attention patterns that are more evenly distributed across natural facial structures. Although high-activation regions are still observed around the forehead and other facial features, the attention is not concentrated on facial blending boundaries as in manipulated images. This suggests that the classifier relies on intrinsic facial characteristics for authentic images while focusing on manipulation artifacts when identifying deepfake content.

Overall, the visualization results indicate that the proposed model learns meaningful manipulation-related features rather than relying on irrelevant background information. This behavior confirms that the integration of YOLOv11 effectively removes distracting background regions, allowing the Xception classifier to concentrate on facial artifacts that are semantically associated with deepfake generation.

Although the proposed framework achieved a testing accuracy of 90.10% with a Recall of 97.11% for the Fake class, recent studies have reported competitive performance by employing modern architectures such as EfficientNet, ConvNeXt, and Vision Transformer, particularly when evaluated on large benchmark datasets. These architectures generally provide stronger feature representation and improved classification capability, especially for complex deepfake manipulation scenarios.

Nevertheless, most existing studies primarily emphasize classification performance and commonly assume that facial regions have already been extracted prior to classification. In contrast, the proposed framework integrates YOLOv11 for automatic face localization with Xception for deepfake classification, complemented by Grad-CAM for model interpretability and implementation as a desktop application. Therefore, the contribution of this study lies not only in achieving competitive classification performance but also in providing an integrated and interpretable end-to-end deepfake detection framework suitable for practical applications.

E. System Implementation and Application Testing

The proposed deepfake detection framework was implemented as a desktop-based application named Snap Detector. The application integrates the YOLOv11 face detector and Xception classifier into a unified user interface that supports image analysis, real-time screen monitoring, webcam-based detection, and detection history management.

Figure 10 Main Interface Of The Snap Detector

The main interface of the Snap Detector application is presented in Figure 10. The interface provides four primary functionalities, namely image or video upload analysis, real-time screen monitoring, webcam-based detection, and detection history management. The interface was designed to simplify user interaction while maintaining accessibility to all detection features.

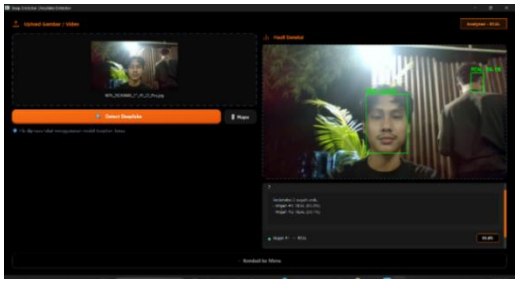

Figure 11 Deepfake Detection Result Using Uploaded Image

Figure 11 shows the detection result obtained from an uploaded image. After the image is processed, YOLOv11 automatically identifies facial regions and generates bounding boxes around detected faces. The cropped facial regions are subsequently analyzed by the Xception classifier to determine whether the detected face belongs to the Real or Fake category. In this example, two faces were successfully localized and classified as authentic with confidence scores of 93.0% and 59.1%, respectively.

FIgure 12 Real-Time Deepfake Detection Using Screen Monitoring

Figure 12 demonstrates the real-time screen monitoring functionality implemented in the proposed system. During operation, the application continuously captures screen content, detects facial regions using YOLOv11, and classifies each detected face using the trained Xception model. The system successfully identified multiple facial instances and generated confidence scores in real time, demonstrating the capability of the framework to perform continuous deepfake analysis.

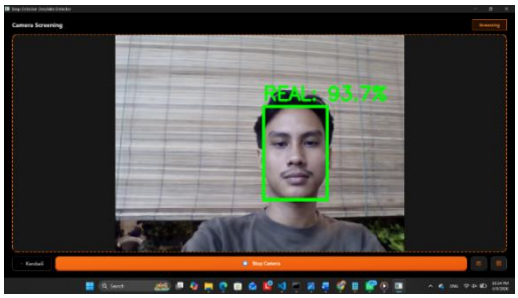

Figure 13 Webcam-Based Deepfake Detection

Figure 13 presents the webcam-based detection mode of the application. In this mode, video frames captured from the webcam are processed continuously through the YOLOv11-Xception pipeline. The detected face was successfully classified as Real with a confidence score of 93.7%, indicating that the system can effectively perform live facial authenticity verification.

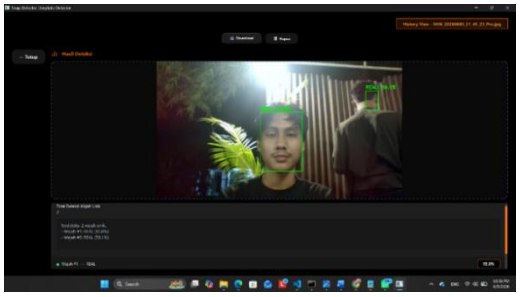

Figure 14 Detection History and Result Visualization

Figure 14 shows the history management feature integrated into the application. The system automatically stores previous detection results, including the analyzed image, classification outcomes, and confidence scores. This functionality enables users to review and manage previously processed data, thereby improving usability and facilitating result documentation.

IV. CONCLUSION

This study successfully developed a facial deepfake detection system by integrating YOLOv11 for face localization and Xception for deepfake classification. Experimental results showed that YOLOv11 achieved highly accurate face localization across the tested dataset, while the Xception classifier obtained a testing accuracy of 90.10%, a Recall of 97.11% for the Fake class, and an AUC value of 0.98. These results indicate that the proposed framework is effective in distinguishing authentic facial images from deepfake-generated content.

Furthermore, the proposed model was successfully implemented in a desktop application called Snap Detector, which supports image analysis, real-time screen monitoring, webcam-based detection, and detection history management. Although the system demonstrated promising performance, the average processing speed of 6.26 FPS indicates that further optimization is required for real-time applications.

The relatively low inference speed can be attributed to the multi-stage processing pipeline adopted in this study. As illustrated in the proposed system architecture, each input image must first undergo face localization using YOLOv11 before the detected facial region is cropped, resized, and subsequently forwarded to the Xception classifier for deepfake prediction. Consequently, the overall inference time represents the cumulative execution time of face detection, image preprocessing, and deepfake classification rather than a single forward pass.

Although the proposed YOLOv11–Xception framework demonstrated promising performance in facial deepfake detection, several limitations should be acknowledged. The proposed framework was trained and evaluated using a combined dataset consisting of 12,344 facial images collected from multiple sources, including the Deepfake Clean Final dataset, self-captured photographs, and publicly available TikTok content. However, the evaluation was conducted on a single combined dataset without cross-dataset validation, which may limit the generalization capability of the proposed framework when applied to other benchmark datasets or unseen deepfake generation methods. In addition, the current inference speed of 6.26 FPS still requires further optimization for practical real-time deployment.

Future work will focus on improving inference efficiency and evaluating the model using larger and more diverse benchmark datasets such as FaceForensics++, Celeb-DF, and DFDC to further enhance the robustness and generalization performance of the proposed system.

REFERENCES

[1] T. T. Nguyen *et al.*, "Deep learning for deepfakes creation and detection: A survey," *Computer Vision and Image Understanding*, vol. 223, p. 103525, Oct. 2022, doi: 10.1016/J.CVIU.2022.103525.

[2] Z. Shen *et al.*, "A Survey of the Past, Present, and Future of Erasure Coding for Storage Systems," *ACM Transactions on Storage*, vol. 21, no. 1, p. 39, Jan. 2025, doi: 10.1145/3708994.

[3] A. A. Khan, A. A. Laghari, S. A. Inam, S. Ullah, M. Shahzad, and D. Syed, "A survey on multimedia-enabled deepfake detection: state-of-the-art tools and techniques, emerging trends, current challenges & limitations, and future directions," *Discover Computing 2025 28:1*, vol. 28, no. 1, pp. 48-, Apr. 2025, doi: 10.1007/S10791-025-09550-0.

[4] N. B. P. Athallah, A. Candra, and H. A. Sanusi, "Urgensi dan Kecukupan Pengaturan Deepfake dalam Undang-Undang Informasi dan Transaksi Elektronik di Indonesia," *Judge : Jurnal Hukum* , vol. 6, no. 09, pp. 1665–1676, Apr. 2026, doi: 10.54209/JUDGE.V6I09.2279.

[5] S. W. Nurdin and I. F. Nugraha, "Ancaman Deepfake Dan Disinformasi Berbasis Ai: Implikasi Terhadap Keamanan Siber Dan Stabilitas Nasional Indonesia," *JIMR : Journal Of International Multidisciplinary Research*, vol. 4, no. 01, pp. 73–92, Jun. 2025, doi: 10.62668/JIMR.V4I01.1551.

[6] T. Raharjo *et al.*, "Analisis Forensik Deepfake Berbasis Convolutional Neural Network (CNN) Untuk Deteksi Inkonsistensi Tekstur dan Pola Pada Citra Wajah," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 2, pp. 2731–2738, Mar. 2025, doi: 10.36040/JATI.V9I2.13058.

[7] C. R. Gunawan, N. Nurdin, and F. Fajriana, "Deteksi Ikan Segar Secara Realtime dengan YOLOv4 menggunakan Metode Convolutional Neural Network," *J. Komtika (Komputasi dan Inform.*, vol. 7, no. 1, pp. 1–11, May 2023, doi: 10.31603/komtika.v7i1.8986.

[8] S. Adventino Gulo, A. Amelia Pertiwi, S. Putri Syaifullah Nasution, and H. Syahputra, "Deteksi Deepfake Dalam Citra Menggunakan Convolutional Neural Network (Cnn)," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 5, pp. 8655–8660, 2025, doi: 10.36040/jati.v9i5.14896.

[9] R. P. Pahlevi and M. Ma'sumah, "Deteksi Deepfake Pada Citra Biometrik Wajah Menggunakan Yolov9 Dan Convolutional Neural Network (Cnn)," *Conference on Innovation and Application of Science and Technology (CIASTECH)*, vol. 8, no. 1, pp. 559–567, Jan. 2025, doi: 10.31328/CIASTECH.V8I1.7518.

[10] L. M. Al Ikhlas, D. Abdullah, and N. Nurdin, "Environmental Image Analysis for Smoke-Free Area Compliance Mapping Using YOLOv11 and Geographic Information Systems", *JAIC*, vol. 10, no. 3, pp. 3034–3045, Jun. 2026.

[11] Y. J. Fan *et al.*, "Machine Learning: Using Xception, a Deep Convolutional Neural Network Architecture, to Implement Pectus Excavatum Diagnostic Tool from Frontal-View Chest X-rays," *Biomedicines 2023, Vol. 11, Page 760*, vol. 11, no. 3, p. 760, Mar. 2023, doi: 10.3390/BIOMEDICINES11030760.

[12] J. Ulfah and N. Nurdin, "Implementasi Metode Deteksi Tepi Canny Untuk Menghitung Jumlah Uang Koin Dalam Gambar Menggunakan Opencv," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3, Aug. 2023, doi: 10.23960/jitet.v11i3.3147.

[13] A. Ilyana, N. Nurdin, and M. Maryana, "Real-Time Detection of Coffee Cherry Ripeness Using YOLOv11," *J. Appl. Informatics Comput.*, vol. 9, no. 4, pp. 1170–1178, 2025

[14] J. Ulfah, M. Ula, F. Fajriana, and N. Nurdin, "Analisis Perbandingan Kinerja Algoritma You Only Look Once (YOLOv8) Dan Single Shot Detector (SSD) dalam Pengenalan Nominal Uang Kertas," Journal of Artificial Intelligence and Software Engineering, vol. 4, no. 2, 2024, doi: 10.30811/jaise.v5i4.7471.

[15] J. Huang, K. Wang, Y. Hou, and J. Wang, "LW-YOLO11: A Lightweight Arbitrary-Oriented Ship Detection Method Based on Improved YOLO11," *Sensors 2025, Vol. 25, Page 65*, vol. 25, no. 1, p. 65, Dec. 2024, doi: 10.3390/S25010065.

[16] C. R. Gunawan, N. Nurdin, and F. Fajriana, "Design of a Real-Time Object Detection Prototype System with YOLOv3 (You Only Look Once)," *Int. J. Eng. Sci. Inf. Technol.*, vol. 2, no. 3, pp. 96–99, 2022.

[17] L. He, Y. Zhou, L. Liu, and J. Ma, "Research and Application of YOLOv11-Based Object Segmentation in Intelligent Recognition at Construction Sites," *Buildings 2024, Vol. 14, Page 3777*, vol. 14, no. 12, p. 3777, Nov. 2024, doi: 10.3390/BUILDINGS14123777.

[18] C. Kwan, B. Li, L. Yu Gong, and X. Jun Li, "A Contemporary Survey on Deepfake Detection: Datasets, Algorithms, and Challenges," *Electronics 2024, Vol. 13, Page 585*, vol. 13, no. 3, p. 585, Jan. 2024, doi: 10.3390/ELECTRONICS13030585.

[19] M. Iman, H. R. Arabnia, and K. Rasheed, "A Review of Deep Transfer Learning and Recent Advancements," *Technologies 2023, Vol. 11, Page 40*, vol. 11, no. 2, p. 40, Mar. 2023, doi: 10.3390/TECHNOLOGIES11020040.

[20] Z. Wang, Q. Liu, and J. Zhang, "Small Object Detection Enhancement in YOLO Networks for Surveillance Applications," *Pattern Recognit. Lett.*, vol. 172, pp. 45–53, 2024.