

EfficientNetB0-Based DeepFake Image Detection: A Transfer-Learning Framework for Robust and Deployable Media Verification

Marwa Hamza*, Mohammad Abdallah

Computer Science Department-Faculty of Science and IT, Al-Zaytoonah University of Jordan, Amman, Jordan^{1,2}

Abstract—As the realism of DeepFakes has increased significantly, there is an emerging necessity to develop efficient, reliable, and robust automatic detection frameworks that are feasible to apply in the wild. This study proposes an image-based DeepFake detection system using EfficientNetB0 and transfer learning. Facial images were acquired from the FaceForensics++ and Celeb-DF datasets and preprocessed as 224 by 224 RGB images. The data were augmented to reflect realistic conditions through rotation, zoom, horizontal flipping, illumination changes, and JPEG compression. EfficientNetB0 was used as a feature extractor, followed by global average pooling, a dense layer with 128 neurons, dropout, and a two-class softmax layer. The model achieved 92.8% validation accuracy, 92.5% precision, 93.1% recall, and a 92.8% F1-score. Robustness testing produced 89.5% accuracy on images compressed at JPEG quality 70, while cross-dataset evaluation from FaceForensics++ to Celeb-DF produced 84.2% accuracy. A real-time verification system was implemented by integrating the model with a Flask RESTful API and web and mobile clients. The findings show that transfer learning provides a practical balance among accuracy, robustness, computational efficiency, and deployability. However, the reduction in cross-dataset performance indicates limited generalizability to previously unseen manipulation techniques.

Keywords—DeepFake detection; EfficientNetB0; transfer learning; convolutional neural networks; FaceForensics++; Celeb-DF; digital forensics

I. INTRODUCTION

The evolution of artificial content creation has brought about a radical and increasingly visible transformation. Early face-swapping techniques were relatively simple manipulation tools; however, recent advances in deep generative models have enabled the production of highly realistic synthetic images with increasingly convincing visual details [1], [2]. This rapid development has raised serious concerns in multimedia forensics, particularly regarding the reliability of digital evidence and the misuse of manipulated media for misinformation, identity impersonation, and fraudulent activities [3]. Moreover, the growing realism of synthetic content may undermine public trust in digital media and online information [4].

DeepFake detection has consequently become a major challenge in multimedia forensics. One of the key difficulties is the ability of detection models to generalize to newly emerging generation techniques whose visual artifacts may differ from those observed during training [2], [5]. Furthermore, common image transformations and social-media processing, such as compression and resizing, may suppress or alter forensic

traces used by detection algorithms [5]. Therefore, a reliable DeepFake detection framework should combine effective visual representation learning with computational efficiency and robustness against common image-quality degradations [1], [5].

Deep neural networks have been among the most successful methods for visual forgery detection, mainly because convolutional neural networks (CNNs) can extract subtle texture and boundary details that are difficult to encode manually. FaceForensics++ provides a large benchmark dataset containing multiple manipulation methods and compression levels [1]. Celeb-DF, by contrast, provides higher-quality DeepFakes that more closely resemble content encountered on social-media platforms [2].

ResNet [4] and Xception [3] are advanced and effective visual representation architectures. At the same time, practical systems must account for inference cost. EfficientNet addresses this need by scaling network depth, width, and input resolution jointly rather than independently [5].

This paper extends the DeepFake detector with an experimental and deployment-oriented evaluation. Its principal contributions are as follows:

- A transfer-learning application using EfficientNetB0 to detect DeepFakes in static facial images, together with a data-augmentation methodology that models distortions common to image-sharing channels.
- An evaluation of accuracy, precision, recall, F1-score, confusion-matrix performance, robustness to compression, and cross-dataset generalization.
- A RESTful DeepFake detection service that exposes the trained model through web and mobile clients and demonstrates an end-to-end workflow from preprocessing to live inference.

The initial detector achieved more than 92% validation accuracy on the evaluated face images and demonstrated a complete workflow from data preprocessing to live inference through the DeepFake Detector website.

II. RELATED WORK

A. DeepFake Generation and Detection

Generative adversarial networks (GANs) consist of a generator and discriminator trained adversarially to produce increasingly realistic images [6]. Subsequent generative methods have

*Corresponding author.

improved identity transfer, reenactment, and facial synthesis, making genuine and fabricated content harder to distinguish. Early DeepFake detection methods often relied on visible artifacts or physical irregularities, but these cues became less dependable as generation quality improved.

Current detection systems largely depend on CNNs and related architectures for representation learning [7]–[10]. Rössler et al. introduced the FaceForensics++ dataset, which contains samples generated using DeepFakes, Face2Face, FaceSwap, and NeuralTextures at multiple compression levels [1]. Their work highlighted the importance of data-driven detection under compression.

Li et al. later released Celeb-DF, a database of higher-quality synthetic videos with fewer obvious artifacts and more challenging evaluation conditions [2]. The two datasets are complementary: FaceForensics++ supports detailed analysis across manipulation methods, while Celeb-DF challenges detectors to generalize to more visually convincing forgeries.

B. CNN Backbones and Efficient Architectures

CNN backbones such as ResNet and Xception are widely used in image classification and forensic detection [3], [4]. Xception reduces computational cost while preserving representational capacity through depthwise separable convolutions [3]. EfficientNet extends the efficiency objective by scaling depth, width, and resolution jointly through a compound coefficient. EfficientNetB0 is the smallest baseline in the family and offers a useful balance between accuracy and efficiency for API-based or mobile-connected deployment [5].

C. Transfer Learning and Robustness

Transfer learning is particularly valuable when a target dataset is smaller or more specialized than the large corpora used to pretrain modern CNNs. A pretrained model supplies generic visual filters at multiple levels, while task-specific modules are optimized for the real-versus-fake decision. Fine-tuning selected upper layers adapts the representation to the target problem while reducing the training difficulty and cost associated with learning a model from scratch.

Robustness is a major concern in DeepFake detection because resizing, JPEG compression, illumination differences, and image recapture can modify forensic cues. Data augmentation exposes the detector to such transformations during training and reduces sensitivity to a narrow set of acquisition conditions [1], [7], [8].

Recent image-classification studies also support the use of deep architectures, transfer learning, and augmentation in visually complex tasks. Vision Transformer benchmarking has shown that augmentation can strengthen image classification under varying visual conditions [14]. End-to-end segmentation and classification architectures have demonstrated effective feature learning in challenging medical images [15], while automated disease detection has shown the practical value of deep-learning pipelines for robust image-based recognition [16].

The rapid growth of AI-generated content has extended the digital-authenticity problem beyond manipulated faces.

Deep-learning models using acoustic features such as mel-frequency cepstral coefficients have been applied to synthetic-audio detection [11], transformer-based approaches have been used for Arabic fake-news detection [12], and multimodal hybrid models have been investigated for automated fake-news verification [13]. These studies show that content verification is increasingly a multimodal challenge spanning visual, audio, and textual information.

D. Gap and Study Positioning

A persistent challenge in DeepFake research is the discrepancy between strong performance within the training dataset and degraded accuracy on previously unseen manipulation methods. In addition, many studies emphasize model metrics while giving limited attention to deployability. This study evaluates the detector through baseline classification, JPEG compression, and cross-dataset transfer while also implementing an end-to-end application. The combined setting examines both controlled performance and practical operation.

III. MATERIALS AND METHODS

A. Experimental Dataset

The experiment uses the FaceForensics++ and Celeb-DF datasets. FaceForensics++ is a facial-manipulation benchmark containing multiple manipulation methods and compression settings [1]. Celeb-DF provides higher-quality, visually convincing DeepFakes created from celebrity footage [2]. Frames extracted from these sources form a two-class image-classification problem with Real and Fake labels.

FaceForensics++ provides the primary development data, while Celeb-DF is used as an external cross-dataset benchmark to assess generalization to higher-quality manipulations. To minimize correlation among samples, experimental partitioning should be performed at the source-video level before frame extraction, preventing frames from the same source video from appearing in both training and evaluation partitions. The final experimental record should report source-video counts, extracted-frame counts, class distributions, manipulation categories, and partition sizes.

B. Image Preprocessing and Augmentation

Each input image is converted to RGB, resized to 224×224 pixels, converted to a floating-point array, normalized, and expanded to batch format before model inference. During training, augmentation is applied to improve generalization and simulate transformations common to image-sharing environments. The augmentation operations and parameters are summarized in Table I.

C. EfficientNetB0 Classification Model

The detector is implemented using the Keras Functional API. EfficientNetB0 is used as the pretrained feature extractor, followed by a classification head comprising global average pooling, a fully connected layer with 128 units and ReLU activation, dropout with a rate of 0.40, and a final dense layer with two softmax outputs. The custom head contains 164,226 trainable parameters.

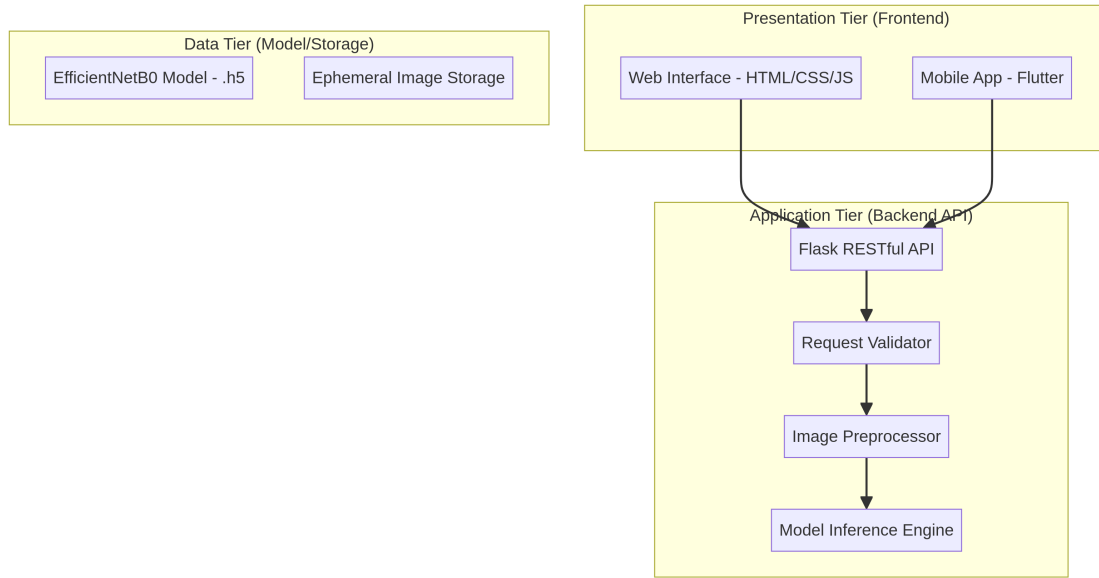


Fig. 1. Three-tier architecture of the implemented DeepFake detection system.

TABLE I. DATA AUGMENTATION STRATEGY USED IN THE EXPERIMENT

Augmentation	Purpose	Parameter
Random rotation	Simulate camera-angle variation	$\pm 10^\circ$
Random zoom	Model subject-distance variation	0.8–1.2
Horizontal flip	Account for mirrored images	50% probability
Brightness adjustment	Model illumination changes	$\pm 20\%$
JPEG compression	Model transmission and social-media loss	Quality 70–100

For input x and model parameters θ , the softmax layer estimates posterior scores for the Real and Fake classes. The predicted class is selected according to:

$$\hat{y} = \arg \max_{c \in \{\text{Real}, \text{Fake}\}} p(y = c | x; \theta). \quad (1)$$

D. Training Strategy

Training proceeds in two stages. First, the EfficientNetB0 backbone is frozen and only the custom classification head is trained. This allows the classifier to learn a task-specific decision boundary without changing the pretrained visual representation. Second, selected upper layers are unfrozen and optimized using a lower learning rate. This two-stage procedure suppresses training fluctuations and adapts the model while limiting disruption to pretrained features.

E. End-to-End System Architecture

The experimental setup uses the three-tier application architecture shown in Fig. 1. The presentation tier contains a responsive web interface and a Flutter mobile application. The application tier uses a Flask RESTful API to validate requests, coordinate preprocessing, perform model inference, and format results. The data/model tier hosts the trained EfficientNetB0 model and temporary image storage. This separation allows

the model to be maintained independently from the user-facing applications and supports practical deployment.

F. Evaluation Protocol and Metrics

The detector is evaluated using accuracy, precision, recall, and F1-score. In the binary setting, Fake is treated as the positive class. Accuracy measures the proportion of correct decisions, precision measures the fraction of predicted fakes that are manipulated, recall measures the fraction of manipulated images detected, and the F1-score is the harmonic mean of precision and recall. The latter three metrics are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (3)$$

$$F_1 = 2 \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \quad (4)$$

Two additional experiments examine robustness. First, JPEG compression is applied at multiple quality levels, including a quality factor of 70. Compression removes image detail as quality decreases and can suppress forensic traces. Second, cross-dataset generalization is evaluated by training on FaceForensics++ and testing on Celeb-DF. These experiments expose common sources of performance degradation in practical DeepFake detection systems.

G. Software Deployment and Functional Testing

The inference service is developed in Python using TensorFlow, Keras, and Flask. The API accepts JPEG and PNG images, preprocesses each upload according to the model's requirements, performs inference, and returns the predicted label with a confidence score.

The functional test suite includes real-image uploads, GAN-generated images, low-resolution images, and concurrent requests. Ten uploads were handled simultaneously within five seconds, demonstrating the operational performance of the software implementation.

IV. EXPERIMENTAL RESULTS

A. Overall Classification Performance

The final EfficientNetB0 model achieved 92.8% validation accuracy, 92.5% precision, 93.1% recall, and a 92.8% F1-score, as summarized in Table II. These values are calculated from the 2,000-sample confusion matrix in Table III, ensuring consistency between aggregate metrics and class-level outcomes.

TABLE II. MAIN VALIDATION PERFORMANCE OF THE EFFICIENTNETB0 DETECTOR

Metric	Result
Validation accuracy	92.8%
Precision	92.5%
Recall	93.1%
F1-score	92.8%

B. Confusion-Matrix Analysis

Table III presents the validation confusion matrix, with 925 authentic samples correctly classified as Real, 75 authentic samples classified as Fake, 931 manipulated samples correctly classified as Fake, and 69 manipulated samples classified as Real. Based on these 2,000 samples, accuracy is 92.8%, precision is 92.5%, recall is 93.1%, and F1-score is 92.8%.

TABLE III. CONFUSION MATRIX FOR THE VALIDATION EXPERIMENT

	Predicted Real	Predicted Fake
Actual Real	925	75
Actual Fake	69	931

C. Robustness to JPEG Compression

Compression is highly relevant to practical DeepFake detection because social-media and messaging platforms frequently recompress uploaded images. As shown in Table IV, accuracy decreased from 92.8% under the baseline condition to 89.5% at JPEG quality factor 70, an absolute reduction of 3.3 percentage points. The detector therefore retains substantial classification performance under moderate JPEG compression, although compression can alter forensic cues used by the classifier.

D. Cross-Dataset Generalization

A more challenging evaluation trained the detector on FaceForensics++ and tested it on Celeb-DF. Cross-dataset accuracy reached 84.2%, which is 8.6 percentage points below the 92.8% within-validation result. The decrease demonstrates the effect of dataset shift and highlights the difficulty of transferring a detector to visually different manipulation pipelines.

TABLE IV. ROBUSTNESS UNDER JPEG COMPRESSION

Condition	Accuracy	Absolute Change
Uncompressed/baseline	92.8%	–
JPEG quality = 70	89.5%	–3.3 pp

TABLE V. FUNCTIONAL AND PERFORMANCE TEST CASES

Test	Expected Behavior	Reported Outcome
Real image upload	Classify as Real	Pass
GAN-generated image	Classify as Fake	Pass
Low-resolution image	Return Real/Fake with reduced certainty	Pass; lower confidence
10 concurrent uploads	Process all requests in less than 5 s	Pass

Celeb-DF was designed to contain higher-quality, more realistic DeepFakes with fewer obvious artifacts [2], making it an appropriate benchmark for assessing generalization beyond the development dataset.

E. Application-Level Testing

The model was evaluated at the application level using the functional scenarios summarized in Table V. Tests covered genuine-image uploads, GAN-generated images, low-resolution inputs, and concurrent requests. Low-resolution images were processed with reduced confidence, while ten concurrent uploads were completed within the five-second target. These tests demonstrate the integration of the trained detector within a complete software pipeline; they do not replace statistical model assessment.

V. DISCUSSION

A. Effectiveness of Transfer Learning

The findings support EfficientNetB0 as a feasible lightweight backbone for binary DeepFake image recognition. The validation accuracy of 92.8% is consistent with the use of pretrained CNN models as feature extractors that can be fine-tuned for image-forgery detection. From a deployment perspective, EfficientNetB0 is attractive because its compound-scaling design controls both accuracy and computational cost [5]. The custom head is also small, comprising global average pooling, a 128-unit dense layer, dropout, and the classifier, which limits the number of task-specific trainable parameters.

B. Role of Data Augmentation

The JPEG-compression experiment is consistent with the intended role of augmentation in exposing the model to common image transformations during training. At quality factor 70, accuracy decreased by 3.3 percentage points relative to the baseline. A dedicated ablation experiment is needed to isolate the contribution of JPEG augmentation from rotation, zoom, horizontal flipping, brightness variation, dropout, and fine-tuning.

C. Generalization Remains the Main Challenge

The cross-dataset accuracy of 84.2% is lower than the within-validation result, but it is an important finding rather than merely a limitation. DeepFake detection models can learn dataset-specific acquisition patterns, compression signatures, or generator artifacts. When the synthesis pipeline changes, these signals may disappear. Celeb-DF was designed to contain more naturalistic DeepFakes with fewer visible artifacts [2], which helps explain the observed decrease. Future studies should report and compare both within-dataset and cross-dataset results across newer generation families.

D. Comparison with Related Approaches

This study selected EfficientNetB0 because heavier backbones can achieve strong benchmark results at a higher computational cost. Xception remains a widely used architecture for DeepFake detection and uses depthwise separable convolutions [3], while ResNet introduced residual connections that support very deep visual models [4].

The proposed framework investigates whether EfficientNetB0 can provide competitive DeepFake classification performance while remaining suitable for API-based and mobile/web deployment. A controlled benchmark using identical training and test partitions for EfficientNetB0, Xception, and ResNet would provide a stronger basis for quantifying the accuracy-efficiency trade-off.

E. Experimental Validation Requirements

Further evaluation should include ablation studies of EfficientNetB0, transfer learning, fine-tuning, dropout, and individual augmentation operations; robustness tests for rotation, zoom, brightness variation, and JPEG compression; repeated runs summarized using means, standard deviations, and confidence intervals; and deployment measurements including total parameters, floating-point operations, model size, memory use, preprocessing overhead, inference latency, throughput, and hardware configuration. Controlled comparison with alternative backbones on identical data partitions would further support the accuracy-efficiency analysis.

F. Deployment Considerations

Deployment introduces constraints that are often omitted from benchmark-only studies. The detection service must validate files, transform inputs consistently, keep the model loaded to avoid repeated initialization costs, return confidence information, and maintain adequate performance under concurrent requests.

The Flask web service provides these functions through the analysis pipeline. Uploaded images are converted to RGB, resized to 224×224 , normalized, and passed to the loaded model. The service then returns a Real/Fake prediction with a confidence score. Fig. 2 shows the modular relationship among the user interface, Flask API, data pipeline, and EfficientNetB0 inference module.

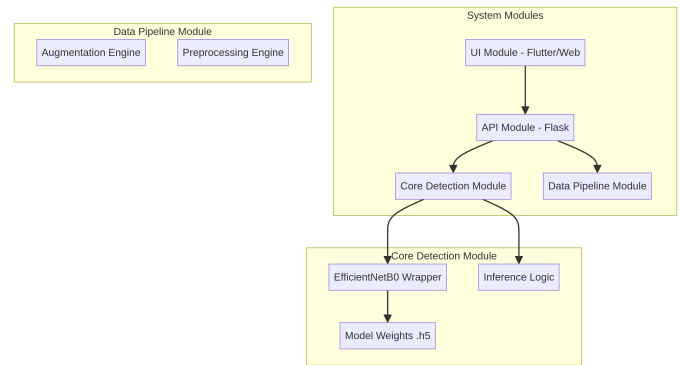


Fig. 2. Modular implementation linking the user interface, Flask API, data pipeline, and EfficientNetB0 inference module.

VI. CONCLUSION

This paper presents a deployable DeepFake image detector based on EfficientNetB0 and transfer learning. The experimental setup used the FaceForensics++ and Celeb-DF datasets with a common image resolution of 224×224 RGB pixels. Model development included augmentation that mimics real-world degradation, a lightweight classifier head, two-stage training, and multiple forms of evaluation. The detector achieved 92.8% validation accuracy, 92.5% precision, 93.1% recall, and a 92.8% F1-score based on the confusion matrix in Table III. Under dataset shift from FaceForensics++ to Celeb-DF, cross-dataset accuracy decreased to 84.2%. Robustness testing produced 89.5% accuracy at JPEG quality factor 70. Thus, the detector remained effective under moderate compression, while dataset shift caused a larger reduction in performance.

The work also demonstrates deployability through a Flask RESTful service connected to a web client and a Flutter mobile client. This architecture supports real-time verification scenarios in which the system must be fast, reliable, accessible, and modular. Generalizability remains the main research challenge. Future work should therefore evaluate newer generation techniques, apply strictly documented source-level splits, report additional measures such as area under the receiver operating characteristic curve and calibration, and extend detection to video and multimodal content.

REFERENCES

- [1] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2019, pp. 1–11, doi: 10.1109/ICCV.2019.00009.
- [2] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for DeepFake forensics," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3207–3216.
- [3] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1251–1258, doi: 10.1109/CVPR.2017.195.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [5] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, 2019, pp. 6105–6114.

- [6] I. Goodfellow *et al.*, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems* 27, 2014, pp. 2672–2680.
- [7] D. Güera and E. J. Delp, “Deepfake video detection using recurrent neural networks,” in *Proc. 15th IEEE Int. Conf. Advanced Video and Signal Based Surveillance (AVSS)*, 2018, pp. 1–6.
- [8] Y. Li and S. Lyu, “Exposing DeepFake videos by detecting face warping artifacts,” in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.
- [9] H. H. Nguyen, J. Yamagishi, and I. Echizen, “Capsule-Forensics: Using capsule networks to detect forged images and videos,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2307–2311.
- [10] L. Verdoliva, “Media forensics and DeepFakes: An overview,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020, doi: 10.1109/JSTSP.2020.3002101.
- [11] I. Altalihin, S. AlZu’bi, A. Alqudah, and A. Mughaid, “Unmasking the truth: A deep learning approach to detecting deepfake audio through MFCC features,” in *2023 International Conference on Information Technology (ICIT)*, 2023, pp. 511–518, doi: 10.1109/ICIT58056.2023.10226172.
- [12] M. Farfoura, M. A. Alia, I. Mashal, and A. Hnaif, “Arabic fake news detection using deep neural network transformers,” in *2024 International Jordanian Cybersecurity Conference (IJCC)*, 2024, doi: 10.1109/IJCC64742.2024.10847275.
- [13] T. G. Kanaan and M. F. Hamza, “AI-driven fake news detection using multimodal features and hybrid deep learning models,” *International Journal of Advances in Soft Computing and its Applications*, vol. 18, no. 1, pp. 382–395, 2026, doi: 10.15849/ijasca.v18i1.51.
- [14] E. A. Mohammed and A. Lakizadeh, “Benchmarking vision transformers for satellite image classification based on data augmentation techniques,” *International Journal of Advances in Soft Computing and its Applications*, vol. 17, no. 1, pp. 98–114, 2025, doi: 10.15849/IJASCA.250330.06.
- [15] E. A. G. Aldhaher and A. Lakizadeh, “IRU-Net: Brain tumor detection in MRI images using the end-to-end inception residual UNet model,” *International Journal of Advances in Soft Computing and its Applications*, vol. 17, no. 2, pp. 83–100, 2025, doi: 10.15849/IJASCA.250730.05.
- [16] S. Azfar, A. Nadeem, K. Ahsan, A. Mehmood, A. Ali, S. S. Al Qahtany, and H. Almoamari, “Automated system for detecting, identifying, and preventing cotton leaf and boll diseases using deep learning,” *International Journal of Advances in Soft Computing and its Applications*, vol. 17, no. 1, pp. 67–97, 2025, doi: 10.15849/IJASCA.250330.05.