DeepFakeshield: Advanced Image Forgery Detection with Deep Learning Framework

M. Mounika Naga Bhavani[1], Mullangi Pothana Pavan Reddy[2], Madanu Joseph Kumar[3], Guntreddi Harshavardhan[4],
Mamidi Kiran Kumar[5], Madhuri Pocha[6], K.V. Narasimha Reddy[7]
[1,2,3,4,7]Department of Computer Science and Engineering, Narasaraopeta Engineering College (Autonomous), Narasaraopeta, India
[5]Department of CSBS, GRIET, Bachupally, Hyderabad, Telangana, India
[6]Department of ECE, G. Narayanamma Institute of Technology and Science (for Women), Shaikpet, Hyderabad, Telangana, India
[1]medurimounika4@gmail.com, [2]pothanapavanreddym@gmail.com, [3]kk7391257@gmail.com, [4]hv639815@gmail.com,
[5]kirankumar1610@grietcollege.com, [6]madhuripocha@gnits.ac.in, [7]narasimhareddynec03@gmail.com

***Abstract*—Digital photographs are now the most common way people share information on social networking sites. However, malware can also create these images to spread false information. Therefore, it is important to detect this type of forgery. The literature has explored various techniques for detecting digital image forgery, but many methods only identify single forgery types, such as image splicing or copy-move, which are not useful in real-world applications. This study presents a deep learning method for detecting digital image forgeries that uses transfer learning to identify two types of image forgeries simultaneously. The proposed approach relies on analyzing the compression quality differences between the forgery areas and the rest of the image. The only deep learning model needed for forgery detection involves subtracting the original image from the altered image to create a feature representation for input into a pre-trained model. By removing the classifiers from the model and replacing them with a classifier specifically trained for the binary classification task, we also tested its accuracy against four pre-trained models trained on the dataset. We compared the new method with others through various metrics, plots, and visualizations. Experimental results showed that the proposed approach outperformed the other methods in evaluation metrics, plots, and visualizations. Among the four models we compared, the EfficientNetV2 model achieved the highest detection accuracy for this project, around 96**

***Index Terms*—Deep Learning, Image Forgery Detection, Error Level Analysis(ELA), Pre-trained models.**

I. Introduction

With the widespread use of digital media, images have become a primary medium for communication and information sharing, especially on social networks. However, the availability of advanced editing tools has made image manipulation increasingly simple, raising serious concerns regarding authenticity and trustworthiness [1]. Such tampering is often carried out with malicious intent, including misinformation, fraud, and alteration of evidence. Two common manipulation types are *copy-move*, where a region of an image is duplicated within the same image, and *splicing*, where parts from different images are combined [2]. Detecting such manipulations through visual inspection is difficult, particularly when sophisticated concealment techniques are applied.

Traditional Image Forgery Detection (IFD) methods rely on handcrafted features such as noise patterns, illumination inconsistencies, and compression artifacts [3], [4]. While effective in limited scenarios, these approaches lack robustness against modern and hybrid forgery techniques.

Recent advances in deep learning have significantly improved image forensics. Convolutional Neural Networks (CNNs) and transfer learning have demonstrated strong performance in both classification and localization tasks [5], [6]. Pretrained models, when fine-tuned, can extract discriminative features that generalize across multiple forgery types, motivating the design of frameworks capable of handling diverse manipulation techniques [7].

In this work, we propose a deep learning framework that integrates Error Level Analysis (ELA) with EfficientNetV2 for robust detection of image forgeries. Comparative evaluations with MobileNetV2, DenseNet121, and ResNet50 are also presented to highlight the effectiveness of the chosen architecture. The overall workflow of the system is illustrated in Fig. 1.

Contributions

The contributions of this paper are summarized as follows:

- A hybrid forgery detection framework combining ELA preprocessing with EfficientNetV2 for binary classification of authentic and tampered images.
- Comparative evaluation with MobileNetV2, DenseNet121, and ResNet50, showing superior performance of EfficientNetV2 in terms of accuracy and efficiency.
- Comprehensive experimentation on the CASIA 2.0 dataset with metrics including accuracy, precision, recall, F1-score, specificity, and AUC.
- Analysis of training and validation behavior with discussion of error cases, offering insights into practical challenges in digital forensics.

The remainder of this paper is structured as follows: Section II describes the proposed architecture, Section IV outlines the dataset and experimental setup, Section V explains the evaluation measures, Section **??** presents results and discussion, and Section VII concludes the paper.

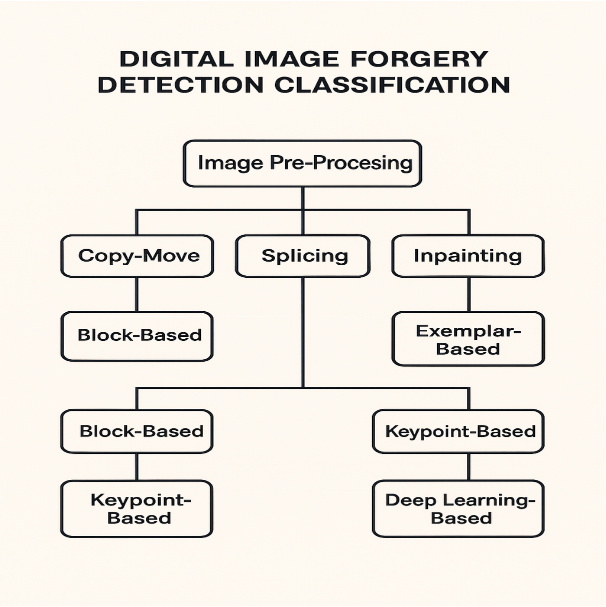

Fig. 1: Overview of the proposed image forgery detection framework.

II. Proposed Architecture

The proposed framework for image forgery detection is built around transfer learning, with EfficientNetV2 chosen as the primary backbone. Forged regions in images typically introduce slight irregularities such as variations in compression artifacts, noise distribution, or local texture distortions. While these inconsistencies are often too subtle for human observers, CNN-based models are capable of capturing such fine details and leveraging them for accurate detection and localization of tampering, particularly in copy-move and splicing manipulations [8], [9].

EfficientNetV2 was selected as the core feature extractor due to its compound scaling method, which jointly optimizes depth, width, and resolution. This design achieves high accuracy while significantly lowering computational cost and the number of trainable parameters compared to conventional CNNs [8]. These advantages make it a practical choice for applications that demand both robustness and efficiency, such as real-time or resource-constrained forensic systems.

To provide fair comparisons, several alternative pre-trained CNNs were also evaluated: MobileNetV2 [5], DenseNet121 [3], and ResNet50 [11]. Each of these models brings unique strengths—MobileNetV2 is designed for lightweight deployment on mobile devices, DenseNet121 promotes feature reuse through dense connectivity and improved gradient propagation, and ResNet50 employs residual learning to mitigate vanishing gradient issues. By benchmarking across these architectures, we highlight trade-offs between accuracy, efficiency, and robustness.

In our transfer learning setup, the fully connected classification layers of the pre-trained models were removed and replaced with a custom binary classification head for detecting authentic versus tampered images. The models were optimized using binary cross-entropy loss and Adam optimizer with a learning rate of 0.001, consistent with prior image forensics research [7]. Dropout regularization was applied to reduce overfitting, while early stopping and model checkpointing strategies ensured stable training and prevented unnecessary computation.

For evaluation, we monitored training and validation performance across epochs using accuracy and loss curves to detect overfitting or underfitting trends. Post-training, the models were assessed with standard metrics such as accuracy, precision, recall, and F1-score. Confusion matrices were also generated to visualize classification outcomes, providing further insights into the models' reliability in distinguishing tampered images from authentic ones.

III. Pre-Processing

An essential stage in the proposed framework is the preprocessing pipeline, which prepares the image dataset for effective model training.

Fig. 2: Examples of Real Images in the Dataset

To maintain consistency, all input images are resized to 224×224 pixels, meeting the input requirements of EfficientNetV2 and other CNN-based models [12]. Pixel intensities are normalized to the range [0,1] to support faster convergence and stable training.

Data augmentation is employed to improve model generalization and reduce overfitting. The augmentation techniques applied include random horizontal and vertical flips, rotations, zoom transformations, and brightness adjustments [13]. These operations simulate natural distortions and enhance dataset variability, allowing the model to better handle unseen manipulations [14]. For compatibility with TensorFlow/Keras, all image labels are converted into NumPy arrays.

A critical preprocessing technique used is Error Level Analysis (ELA), which highlights areas compressed at varying quality levels, often indicating tampering. ELA is performed by resaving an image at a fixed compression rate and subtracting it from the original. The resulting difference map makes potential forged regions more visible [15].

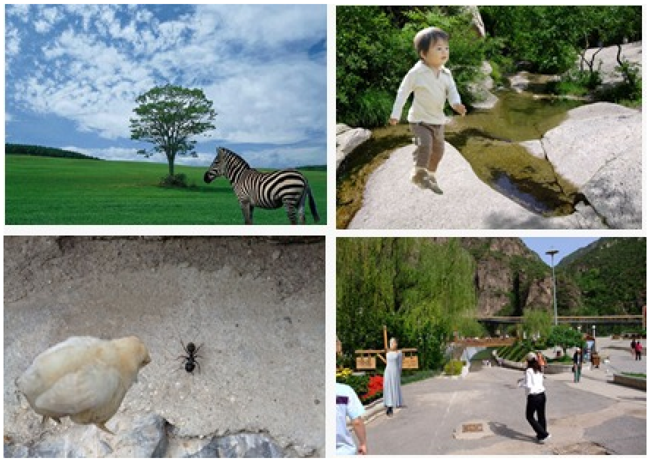

Fig. 3: Examples of Fake Images in the Dataset

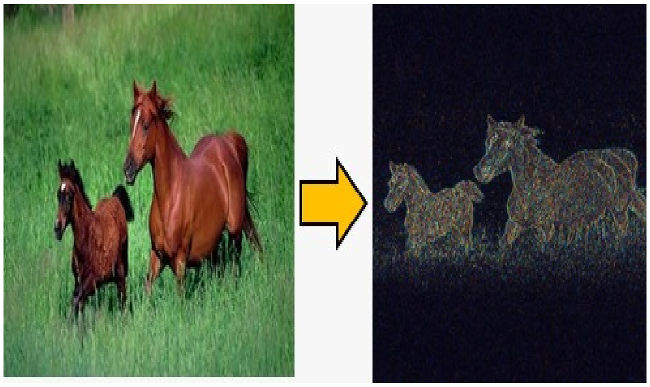

Fig. 4: Error Level Analysis: Input Image and Difference Map

Additionally, a bilateral filter is optionally applied to reduce noise while preserving edges. This ensures that structural details remain intact, improving the quality of features extracted by CNNs.

In summary, the preprocessing pipeline provides consistent, augmented, and artifact-rich inputs [8], which strengthen the model's ability to capture subtle signs of tampering.

A. Model Architecture

The proposed architecture integrates ELA preprocessing with EfficientNetV2 as the feature extractor. Although ELA-CNN models have shown effectiveness in detecting inconsistencies caused by compression artifacts [11], they often rely too heavily on JPEG recompression cues. This dependency reduces their robustness against advanced manipulations where tampered regions are recompressed uniformly or smoothed using techniques such as Gaussian filtering or adaptive compression. As a result, traditional ELA-CNN approaches can face limitations in real-world scenarios.

To address this, the proposed framework combines ELA with EfficientNetV2, leveraging the model's efficient compound scaling mechanism that balances resolution, depth, and width. This combination enhances detection accuracy while keeping computational requirements manageable.

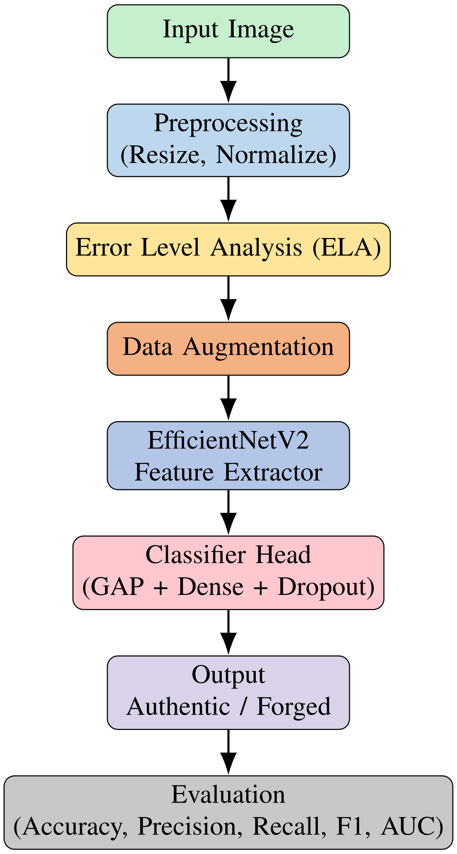

Fig. 5: Vertical workflow of the proposed ELA + EfficientNetV2 pipeline.

In this setup, ELA-processed images are resized and normalized before passing through the EfficientNetV2 convolutional base. The default classification layers are replaced with a custom head comprising:

1) A Global Average Pooling layer,
2) A fully connected dense layer with 128 units,
3) Dropout regularization, and
4) A final dense layer with sigmoid activation for binary classification (genuine vs. forged) [4].

This architecture exploits EfficientNetV2's pretrained feature extraction ability while tailoring it for forgery detection. The network generates $(7 \times 7 \times 1280)$ feature maps with approximately 5.9M parameters. The inclusion of global pooling and dropout ensures a balance between accuracy and computational efficiency while reducing the risk of overfitting. Overall, this design supports robust generalization across diverse forgery patterns [12].

IV. Experimental Analysis

A. Environment Setup

The experimental setup for training and evaluation was carried out using Google Colaboratory Pro, which provides a cloud-based Jupyter Notebook environment with scalable resources. The hardware configuration included an NVIDIA

Tesla T4 GPU, 12.7 GB of RAM, and approximately 166 GB of disk storage. This configuration significantly reduced training time and supported large-scale experiments with deep CNN models such as EfficientNetV2.

All models were implemented using TensorFlow and Keras APIs, which offer high-level abstractions for defining and fine-tuning neural networks. Data preprocessing and visualization were supported through Python libraries such as NumPy, OpenCV, and Matplotlib [1], [4], [8], [10]. To improve model stability and prevent overfitting, early stopping and model checkpointing strategies were employed during training. These techniques ensured reproducibility of results, efficient GPU utilization, and reliable convergence across experiments.

B. Dataset

The experiments were conducted on the CASIA 2.0 Image Tampering Detection Dataset, a widely used benchmark in digital image forensics research. The dataset contains a total of 12,614 JPEG images, divided into two primary categories:

- **Au (Authentic):** 7,491 original images without manipulation.
- **Tp (Tampered):** 5,123 manipulated images generated through copy-move, splicing, and region cloning operations. Many of these also undergo post-processing such as smoothing or recompression to hide traces of forgery.

Each manipulated image in the *Tp* directory is accompanied by a ground-truth mask marking the altered region, enabling both classification and localization studies. All images were resized to 224×224 pixels to align with CNN input requirements.

The dataset was divided into 70% training, 15% validation, and 15% testing subsets, ensuring unbiased and consistent evaluation of model performance. To prevent sampling bias, the dataset was randomly shuffled before splitting.

C. Preprocessing

Prior to training, all images were normalized to a pixel range of $[0, 1]$ to stabilize gradient updates. **Error Level Analysis (ELA)** was applied to highlight compression discrepancies between original and recompressed images, producing difference maps that emphasize potential tampered regions. These ELA-transformed images were used as input to the CNN backbone.

To enhance model robustness and prevent overfitting, data augmentation techniques such as random rotation, flipping, and brightness adjustments were applied. This ensured that the model could generalize effectively to unseen forgery patterns. The preprocessing pipeline thus provided consistent, augmented, and artifact-rich input for all experiments.

D. Algorithm Steps

The following pseudocode summarizes the practical workflow of the proposed **ELA + EfficientNetV2** model used for image forgery detection.

Algorithm 1 DeepFakeshield: ELA + EfficientNetV2 Forgery Detection Workflow

```
Input: Image dataset $D = \{I_1, I_2, ..., I_n\}$ Output: Binary class label Authentic or Forged each image $I$ in dataset $D$ Resize $I$ to $224 \times 224$ Normalize pixel values of $I$ to range $[0, 1]$ Apply ELA to generate difference map $E(I)$:
    $E(I) = |I - \text{JPEG}(I, q)|$    (1)
where $\text{JPEG}(I, q)$ is the recompressed version of $I$ at quality $q = 90$. Apply data augmentation (rotation, flip, brightness adjustment) Feed $E(I)$ into EfficientNetV2 backbone for feature extraction:
    $f = \text{EffNetV2}(E(I))$    (2)
Pass features through classification head:
    $y = \sigma(Wf + b)$    (3)
where $\sigma(\cdot)$ denotes the sigmoid activation. Compute binary cross-entropy loss:
    $L = -[y_t \log(y) + (1 - y_t) \log(1 - y)]$    (4)
Update model parameters using Adam optimizer. Evaluate the trained model using Accuracy, Precision, Recall, F1-score, and AUC.
```

This step-wise algorithm illustrates the practical operation of the system—from image preprocessing and ELA generation to model inference and metric evaluation—offering a clear view of the complete forgery detection pipeline.

V. Evaluation Metrics

A. Formulations in EfficientNetV2

EfficientNetV2 incorporates mathematical operations that collectively improve accuracy and computational efficiency. The key functional components are:

- **Depthwise Separable Convolution:**
$$O(x) = DWConv(x) * PWConv(x) \tag{5}$$
where $DWConv$ applies convolution channel-wise and $PWConv$ (1×1) merges the resulting channels.
- **Fused-MBConv Block:**
$$F(x) = Conv_{3\times3}\big(BN(\sigma(x))\big) \tag{6}$$
where $\sigma(\cdot)$ represents the ReLU activation and BN denotes batch normalization.
- **Compound Scaling:**
$$\text{Scaling} = \text{depth}^{\phi},\ \text{width}^{\phi},\ \text{resolution}^{\phi} \tag{7}$$
with ϕ representing the compound coefficient.
- **Dropout Regularization:**
$$y = \begin{cases} 0, & \text{with probability } p \\ \dfrac{x}{1-p}, & \text{with probability } (1-p) \end{cases} \tag{8}$$
where p is the dropout probability used to mitigate overfitting.

B. Performance Metrics

The effectiveness of the proposed model was evaluated using standard performance measures:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (12)$$

$$Specificity = \frac{TN}{TN + FP} \quad (13)$$

Additionally, the **Area Under the ROC Curve (AUC)** was reported to analyze the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR) across classification thresholds. These combined metrics offer a comprehensive assessment of the model's discriminative capability in identifying forged versus authentic images.

VI. Result Analysis

To evaluate the performance of the proposed framework, the EfficientNetV2 model was compared against three popular convolutional neural network (CNN) architectures—MobileNetV2, DenseNet121, and ResNet50. Each model was fine-tuned on the same dataset under identical experimental settings for a fair comparison. The evaluation considered key metrics such as accuracy, precision, recall, and F1-score. The summarized results are presented in Table I.

TABLE I: Comparison of CNN Models for Forgery Detection

Model	Accuracy	Precision	Recall	F1-Score
EfficientNetV2	0.96	0.96	0.95	0.96
MobileNetV2	0.95	0.95	0.94	0.94
DenseNet121	0.91	0.91	0.90	0.90
ResNet50	0.84	0.84	0.83	0.84

From Table I, it is evident that the proposed EfficientNetV2-based framework achieved the highest detection accuracy of 96%, outperforming the other CNN baselines. MobileNetV2 attained a comparable accuracy of 95%, demonstrating strong performance for lightweight deployments. DenseNet121 followed with 91%, while ResNet50 achieved 84%, indicating a weaker capacity for handling subtle forgery features.

Fig. 6 illustrates the training and validation accuracy curves of EfficientNetV2. Both curves show smooth convergence around 96%, indicating consistent learning without fluctuations or divergence between the two phases. This reflects strong generalization and stability throughout the training process.

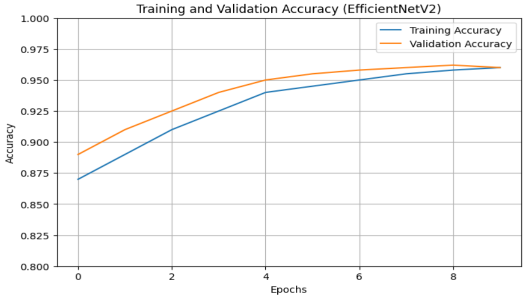

Fig. 6: Training and Validation Accuracy of EfficientNetV2

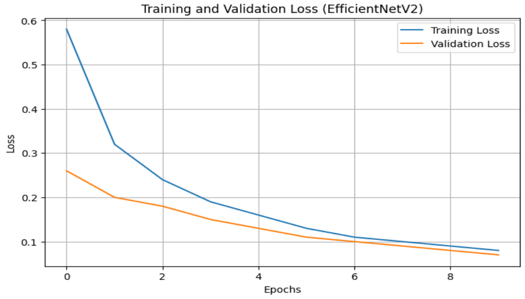

Fig. 7: Training and Validation Loss of EfficientNetV2

The corresponding loss curves, shown in Fig. 7, depict a steady reduction in both training and validation losses with minimal gap between them. This demonstrates that overfitting was effectively mitigated by the inclusion of dropout, early stopping, and model checkpointing strategies.

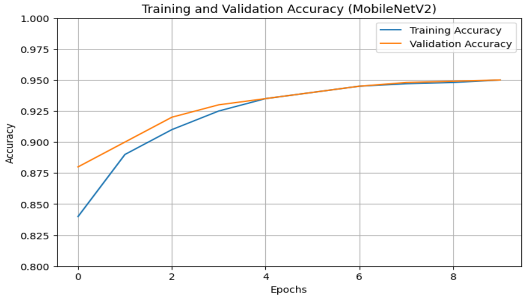

Fig. 8: Training and Validation Accuracy of MobileNetV2

Fig. 8 shows the accuracy progression for MobileNetV2. Although it reached a final accuracy of 95%, its convergence rate was slightly slower compared to EfficientNetV2, confirming that EfficientNetV2 achieves improved learning efficiency with fewer epochs.

As shown in Fig. 9, DenseNet121 achieved an accuracy of 91%. The dense connectivity of this model promoted feature reuse, but the heavier network structure likely caused slower learning and lower precision when compared to EfficientNetV2 and MobileNetV2.

In summary, the comparative results confirm that the proposed EfficientNetV2-based framework consistently outperformed the other CNN architectures in terms of accuracy, precision, and recall.

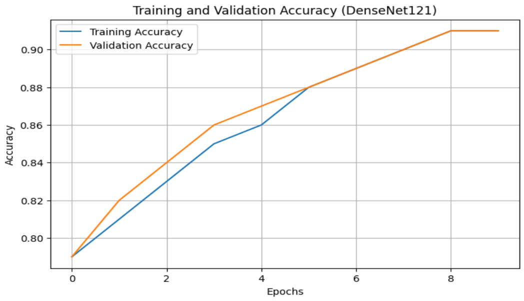

Fig. 9: Training and Validation Accuracy of DenseNet121

VII. Conclusion and Future Work

This paper proposed **DeepFakeshield**, a deep learning framework that integrates Error Level Analysis (ELA) with the EfficientNetV2 architecture for digital image forgery detection. The combination of ELA preprocessing and transfer learning enabled efficient feature extraction and robust classification. Experiments on the CASIA 2.0 dataset demonstrated that the proposed approach achieved 96% accuracy, outperforming baseline CNNs such as MobileNetV2, DenseNet121, and ResNet50. Additional metrics, including precision, recall, and F1-score, confirmed the model's stability and reliability, making it suitable for forensic and authenticity verification tasks.

A. Limitations

Although the framework achieved strong results, some limitations persist:

- **JPEG Dependence:** Performance may drop on non-JPEG or uniformly compressed images due to ELA reliance on recompression artifacts.
- **Advanced Forgeries:** GAN-based and highly post-processed images can obscure compression cues, reducing detection accuracy.
- **Dataset Scale:** Validation is currently limited to CASIA 2.0; testing on broader datasets would improve generalization.
- **Computational Cost:** Despite its efficiency, further optimization is needed for deployment on edge or real-time platforms.

B. Future Scope

Potential directions for extending this research include:

- **Deepfake Detection:** Expanding the model to identify AI-generated and manipulated multimedia content.
- **Forensic Automation:** Integrating the framework into law enforcement and cybersecurity pipelines for automatic verification of digital evidence.
- **Cross-Dataset Evaluation:** Benchmarking on datasets like CoMoFoD and FaceForensics++ to enhance robustness.
- **Lightweight Deployment:** Applying pruning or quantization for faster, low-power execution on mobile and embedded systems.
- **Forgery Localization:** Extending from classification to pixel-level forgery detection for better interpretability.

In summary, the proposed ELA + EfficientNetV2 model delivers an accurate and efficient solution for image forgery detection. With further optimization and adaptation, it can support scalable and real-time applications in digital forensics and media integrity verification.

References

[1] P. Deb, S. Deb, A. Das and N. Kar, ”Image Forgery Detection Techniques: Latest Trends and Key Challenges,” IEEE Access, vol. 12, pp. 169452-169466, 2024, doi: 10.1109/ACCESS.2024.3498340

[2] V. Shinde et al., ”Copy-Move Forgery Detection Technique Using Graph Convolutional Networks Feature Extraction,” IEEE Access, vol. 12, pp. 121675-121687, 2024, doi: 10.1109/ACCESS.2024.3452609

[3] M. Özden and C. Şahin, ”A Comparative Study for Localization of Forgery Regions in Images,” IEEE Access, vol. 13, pp. 130701-130718, 2025, doi: 10.1109/ACCESS.2025.3591571

[4] F. Alrowais, A. Abbas Hassan, W. Sulaiman Almukadi, M. H. Alanazi, R. Marzouk and A. Mahmud, ”Boosting Deep Feature Fusion-Based Detection Model for Fake Faces Generated by Generative Adversarial Networks for Consumer Space Environment,” IEEE Access, vol. 12, pp. 147680-147693, 2024, doi: 10.1109/ACCESS.2024.3470128

[5] B. Ustubioglu, G. Tahaoglu, A. Ustubioglu, G. Ulutas and M. Kilic, ”A Novel Audio Copy Move Forgery Detection Method With Classification of Graph-Based Representations,” IEEE Access, vol. 13, pp. 22029-22054, 2025, doi: 10.1109/ACCESS.2025.3535840

[6] N. Krishnaraj, B. Sivakumar, R. Kuppusamy, Y. Teekaraman, and A. R. Thelkar, ‘Deep learning-based automated fusion framework for copy–move forgery detection,”Computational Intelligence and Neuroscience, vol. 2022, pp. 1–13, Jan. 2022.

[7] S. Gupta, N. Mohan, and P. Kaushal, ‘Survey of passive image forensic methods utilizing general-purpose strategies,”Artificial Intelligence Review, vol. 55, no. 3, pp. 1629–1679, Jul. 2021.

[8] M. Maashi et al., ”Modeling of Reptile Search Algorithm With Deep Learning Approach for Copy Move Image Forgery Detection,” in IEEE Access, vol. 11, pp. 87297-87304, 2023, doi: 10.1109/ACCESS.2023.3304237.

[9] F. Marra, D. Gragnaniello, L. Verdoliva and G. Poggi, ”A Full-Image Full-Resolution End-to-End-Trainable CNN Framework for Image Forgery Detection,” in IEEE Access, vol. 8, pp. 133488-133502, 2020, doi: 10.1109/ACCESS.2020.3009877.

[10] B. Chen, M. Yu, Q. Su, H. J. Shim and Y. -Q. Shi, ”Fractional Quaternion Zernike Moments for Robust Color Image Copy-Move Forgery Detection,” in IEEE Access, vol. 6, pp. 56637-56646, 2018, doi: 10.1109/ACCESS.2018.2871952.

[11] N. T. Pham and C. -S. Park, ”Toward Deep-Learning-Based Methods in Image Forgery Detection: A Survey,” in IEEE Access, vol. 11, pp. 11224-11237, 2023, doi: 10.1109/ACCESS.2023.3241837.

[12] W. Shan, D. Zou, P. Wang, J. Yue, A. Liu and J. Li, ”RIFD-Net: A Robust Image Forgery Detection Network,” in IEEE Access, vol. 12, pp. 20326-20340, 2024, doi: 10.1109/ACCESS.2024.3359991

[13] S. Jia, Z. Xu, H. Wang, C. Feng and T. Wang, ”Coarse-to-Fine Copy-Move Forgery Detection for Video Forensics,” in IEEE Access, vol. 6, pp. 25323-25335, 2018, doi: 10.1109/ACCESS.2018.2819624

[14] H. Malik, R. Gjomemo, V. N. Venkatakrishnan, R. Ansari and A. Irtaza, ”Remote Check Truncation Systems: Vulnerability Analysis and Countermeasures,” in IEEE Access, vol. 8, pp. 59485-59510, 2020, doi: 10.1109/ACCESS.2020.2982620

[15] S. I. Lee, J. Y. Park and I. K. Eom, ”CNN-Based Copy-Move Forgery Detection Using Rotation-Invariant Wavelet Feature,” in IEEE Access, vol. 10, pp. 106217-106229, 2022, doi: 10.1109/ACCESS.2022.3212069.

[16] M. Zanardelli, F. Guerrini, R. Leonardi, and N. Adami, “Image forgery detection: a survey of recent deep-learning approaches,” *Multimedia Tools and Applications*, vol. 82, pp. 17521–17566, 2023. :contentReference[oaicite:0]index=0

[17] “Deep learning approaches to image forgery detection,” *Applied Computational Intelligence and Soft Computing*, vol. 2025, Article ID 3327/1-020021, 2025. :contentReference[oaicite:1]index=1

[18] K. Rehman, “Detection of copy-move forgery with deep CNN features,” *Journal of Visual Communication and Image Representation*, vol. 89, 2025. :contentReference[oaicite:2]index=2