ForgeryLens: Detecting Digital Image Forgery using Error Level Analysis and Transfer Learning with Lightweight CNN Architectures for Future Mobile Deployment

Sarthak Dey, Ritish Prabhat Bhatt
IIT Kharagpur
Emails: sarthak.dey.xyz@gmail.com, bhattritish05@gmail.com

***Abstract*—The widespread availability of powerful image editing software has made digital image forgery a significant problem, casting doubt on the authenticity of digital media. This project presents a deep learning approach for detecting digital image forgery, focusing specifically on lightweight architectures suited for low-power devices. The method combines Error Level Analysis (ELA) to highlight compression artifacts with a suite of convolutional neural networks (CNNs)—ResNet18, ResNet34, ResNet50, MobileNetV2, and EfficientNet-Lite0—for classification.**

Although the long-term vision of the project is to deploy ForgeryLens as a fully on-device mobile or embedded application, all training and experiments were performed on Kaggle GPU accelerators due to their reliability and availability. EfficientNet-Lite0, while originally designed for TensorFlow Lite environments, was used here through its TensorFlow Hub implementation rather than as an actual TFLite model.

ELA preprocessing was applied to all images from the CASIA2 dataset, after which a balanced training set was formed by sampling 2,100 authentic images and combining them with all 2,064 tampered images. MobileNetV2 achieved the best trade-off between accuracy, model size, FLOPs, and inference latency, demonstrating clear suitability for future deployment in mobile or edge-based systems.

***Index Terms*—Image Forgery Detection, Error Level Analysis (ELA), Deep Learning, Lightweight CNN, Transfer Learning, Mobile Deployment.**

I. Introduction

In the modern digital age, the manipulation of images has become increasingly simple and accessible [2]. This ease of forgery poses a significant threat in fields ranging from journalism and law enforcement to social media, where fake images can be used to spread misinformation. This phenomenon erodes public trust, impacts the integrity of legal evidence, and can be used for malicious purposes such as defamation or political propaganda. Therefore, the development of automated and reliable detection tools is a paramount concern for digital forensics.

One of the most common forms of forgery is image splicing, where a portion of one image is pasted into another. When a tampered image is re-saved (typically in a lossy format like JPEG), the spliced region will have a different compression history than the rest of the image. This difference in compression artifacts is often invisible to the naked eye.

Error Level Analysis (ELA) is a forensic technique designed to visualize these compression discrepancies [3]. It operates by re-compressing an image at a known quality level and calculating the difference between the original and the re-compressed version. Authentic sections of an image tend to have a uniform ELA, while tampered sections exhibit a significantly different error level [3].

While ELA provides a visual map of potential forgery, human interpretation can be subjective. This project leverages the power of Convolutional Neural Networks (CNNs), which are state-of-the-art in image classification, to automatically learn the patterns that differentiate authentic and tampered ELA maps [1]. Specifically, we evaluate multiple lightweight CNN architectures with the long-term goal of deploying ForgeryLens on mobile or embedded devices such as smartphones or Raspberry Pi platforms. All experiments, however, were performed on Kaggle-provided GPUs.

This paper details the methodology of combining ELA with a suite of CNN models, trained on the CASIA2 dataset, and presents the experimental results comparing their performance in terms of accuracy, efficiency, and suitability for low-resource deployment.

II. Related Work

Recent research in image forgery detection has increasingly shifted toward deep learning–based architectures that target both detection and localization of manipulated regions. A major direction in recent literature is the emergence of transformer-based and contrastive-learning-based systems.

A notable 2024 study introduced a forged-aware adaptive transformer built on a CLIP-ViT backbone combined with a manipulation-aware adapter module [7]. This transformer model significantly improved synthetic forgery detection but remains computationally expensive.

Another influential paper explored pixel-level contrastive learning paired with unsupervised clustering for representation learning [8]. While effective in capturing manipulation-

consistent features, the custom CNN architecture and contrastive pipeline require heavy compute resources.

Work in [9] proposed a GAN-based fusion network integrating multiple CNNs for joint forgery detection and localization. Such architectures offer fine-grained localization at the cost of high memory consumption and slow inference.

Other studies combined VGG16 encoders with U-Net decoders for simultaneous manipulation detection and localization [10]. Although effective, VGG16-based pipelines are large and unsuitable for low-power hardware.

A 2024 study also introduced an EfficientFormer-based classifier alongside a BCU-Net architecture with spatial-attention mechanisms [11]. Even though EfficientFormer reduces computation compared to transformers, it still exceeds the resource envelope of mobile-targeted architectures like MobileNetV2 or EfficientNet-Lite0.

In contrast, ForgeryLens focuses on models that balance performance and computational efficiency, enabling potential deployment on embedded systems and consumer mobile devices.

III. Methodology

The core methodology of this project consists of three stages: dataset preparation, ELA-based feature extraction, and transfer learning with a CNN.

A. Dataset

We used the CASIA 2.0 Image Tampering Detection Dataset [4], [5]. This dataset contains two main categories of images:

- **Au (Authentic):** Original, unaltered images.
- **Tp (Tampered):** Images that have been manipulated, primarily through splicing or copy-move forgery.

Because the genuine class is significantly larger than the tampered class, a balanced dataset was created by:

- Randomly sampling **2,100** authentic (Au) images.
- Using all **2,064** tampered (Tp) images.

This produced a balanced dataset of 4,164 images.

B. Error Level Analysis (ELA)

To expose forgery artifacts, every image in the dataset was processed using an ELA function. The function works as follows:

1) The original image is opened and converted to RGB.
2) It is re-saved as a temporary JPEG file at 90% quality.
3) The pixel-wise difference between the original image and the recompressed version is computed.
4) The resulting difference image is amplified based on the maximum intensity.

Below are placeholder figures representing original vs. ELA-processed images:

Fig. 1. Original vs. ELA images (placeholders).

C. Model Architecture and Training

The project evaluates five CNN architectures:

- **ResNet18** – Custom 18-layer residual network.
- **ResNet34** – Custom 34-layer residual network.
- **ResNet50** – Deep residual network pre-trained on ImageNet.
  Note: In the ResNet family, the number (e.g., 18, 34, 50) represents the total number of layers in the network. All ResNet variants share a similar residual block structure, with deeper models introducing bottleneck blocks.

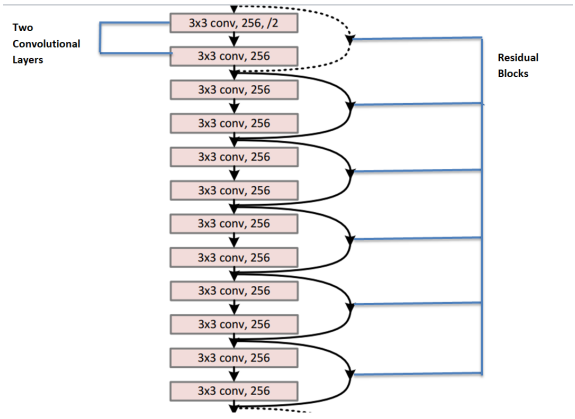

Fig. 2. General ResNet architecture

- **MobileNetV2** – Lightweight, mobile-friendly architecture.

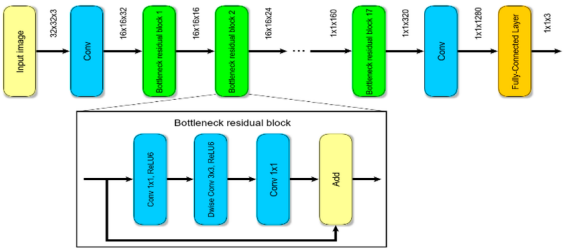

Fig. 3. MobileNetV2 architecture

- **EfficientNet-Lite0** – An architecture optimized for TensorFlow Lite; here used through TensorFlow Hub rather than as a TFLite model.

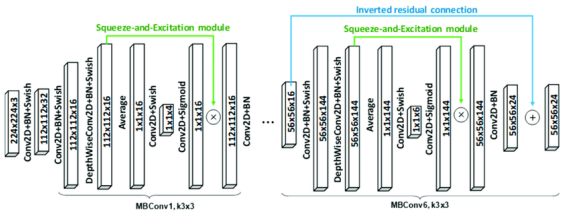

Fig. 4. EfficientNet-Lite0 architecture

All backbone networks were frozen to accelerate training. Only the custom classification head was updated during training. Although the long-term plan is edge deployment, all experiments were executed on Kaggle GPU accelerators.

IV. Experimental Results

A. Training and Validation

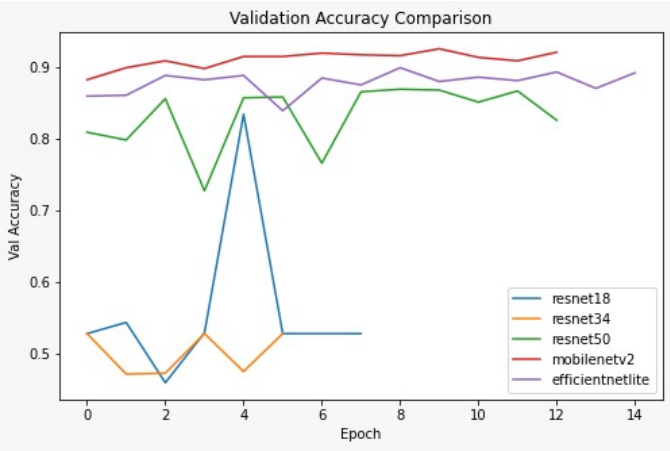

Fig. 5. Training and validation curves (placeholder).

B. Confusion Matrices

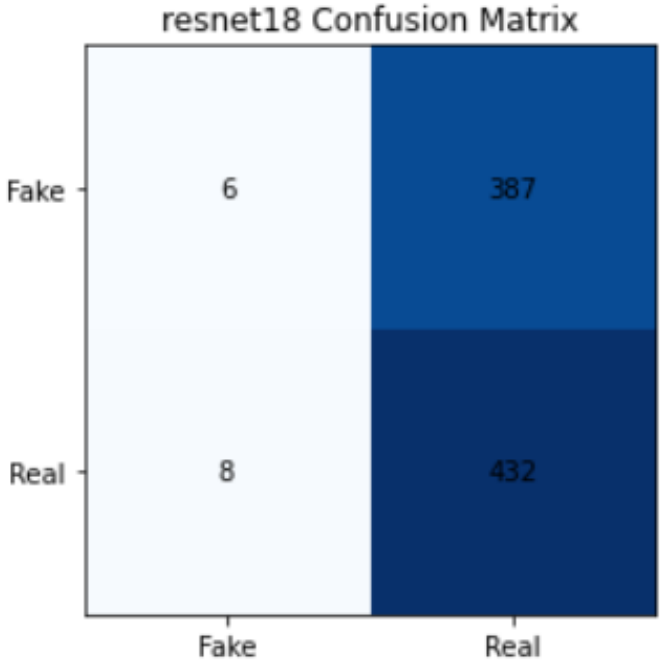

Fig. 6. Confusion Matrix — ResNet18

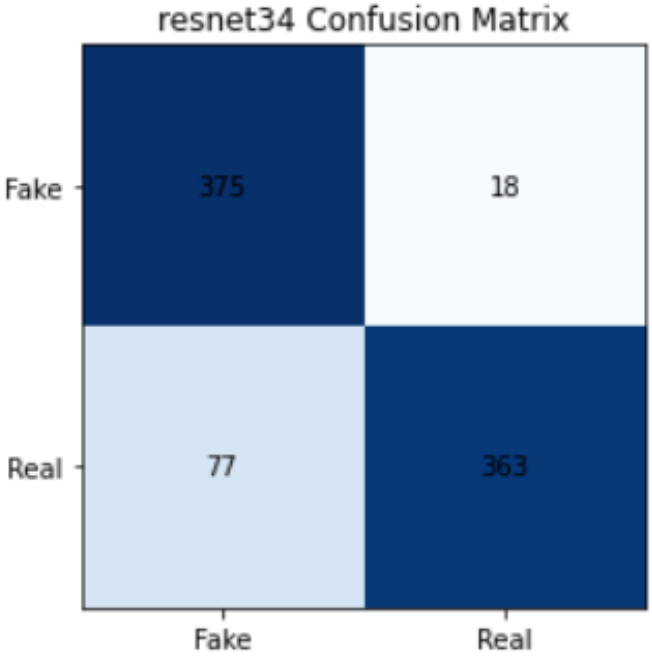

Fig. 7. Confusion Matrix — ResNet34

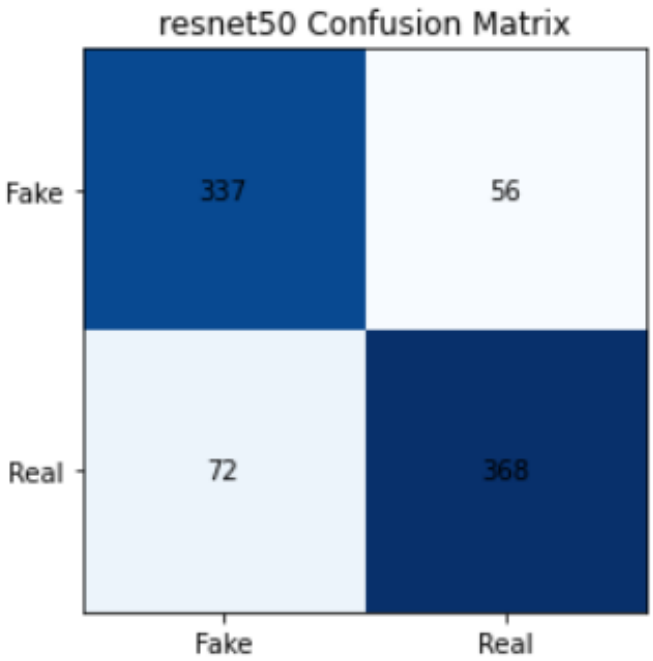

Fig. 8. Confusion Matrix — ResNet50

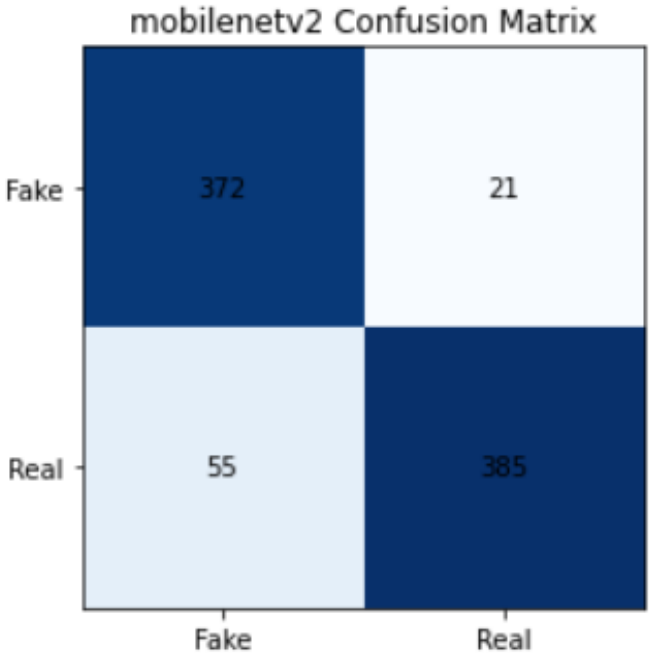

Fig. 9. Confusion Matrix — MobileNetV2

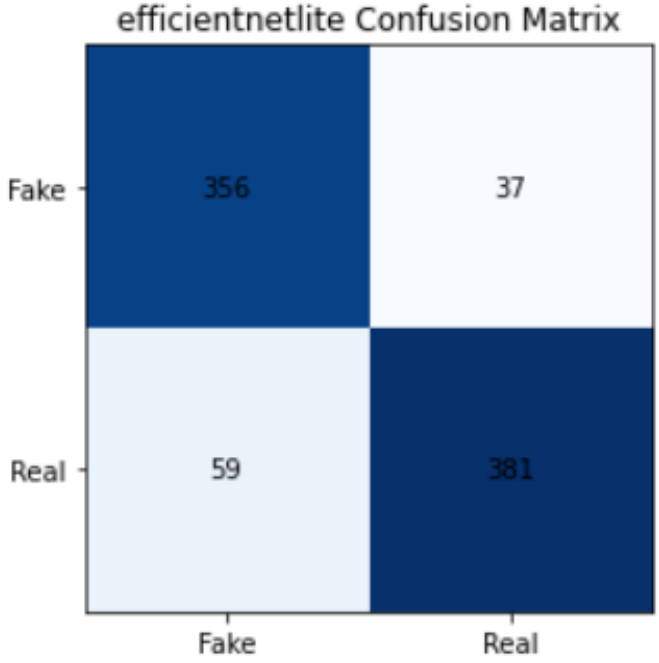

Fig. 10. Confusion Matrix — EfficientNet-Lite0

TABLE I
FINAL MODEL COMPARISON

Model	Accuracy	Infer (ms)	Params	Size (MB)	FLOPs
ResNet18	0.526	35.68	11,322,754	129.88	36,374,501,24
ResNet34	0.886	40.58	21,442,050	245.91	73,412,389,24
ResNet50	0.846	41.60	24,112,770	96.49	77,522,471,80
MobileNetV2	0.909	40.42	2,422,210	10.95	6,130,545,40
EfficientNet-Lite0	0.885	34.88	3,577,250	15.12	7,824,692,92

V. CONCLUSION

This project successfully demonstrates that a combination of Error Level Analysis and lightweight CNN architectures can reliably detect digital image forgeries. Of the five evaluated architectures, MobileNetV2 offered the best balance of accuracy, computational cost, and model compactness, aligning well with the long-term vision of deploying ForgeryLens on mobile and embedded devices.

VI. FUTURE WORK

- Convert models to TensorFlow Lite and test on mobile hardware.
- Deploy on Raspberry Pi for real-time inference.
- Expand the dataset with more manipulation types.

VII. APPENDIX

Sarthak Dey

- Identified and selected the most suitable deep learning architectures for the project, focusing on lightweight CNN models for mobile deployment.

- Trained and fine-tuned all models using transfer learning, including ResNet18, ResNet34, ResNet50, MobileNetV2, and EfficientNet-Lite0.
- Implemented the complete training pipeline, optimization loops, and evaluation metrics.

Ritish Prabhat Bhatt

- Conducted the literature review, summarizing existing approaches, transformer-based models, and recent advancements in image forgery detection.
- Led the Error Level Analysis (ELA) component, including understanding, implementing, and verifying the preprocessing pipeline.
- Handled dataset organization and ELA-based data preprocessing for the CASIA2 dataset.

REFERENCES

[1] A. R. Nandy et al., "Image Forgery Detection Comparison between MobileNetV2 and VGG16 Convolutional Neural Networks," BRAC University, 2020.

[2] "ERROR LEVEL ANALYSIS IN IMAGE FORGERY DETECTION," IRJET, vol. 10, no. 7, 2023.

[3] S. K. J. et al., "Image forgery detection using error level analysis and deep learning," ResearchGate, 2019.

[4] R. Z. et al., "Detecting image manipulation with ELA-CNN integration," NIH, 2023.

[5] A. M. et al., "DeepForgery Images Detection Using Deep Learning Approaches and Error Level Analysis," JOIV, 2024.

[6] A. K. et al., "Digital Image Forgery Detection Using CNN and Error Level Analysis (ELA)," Nanotechnology Perceptions, 2025.

[7] Z. Wang et al., "Forged-aware Adaptive Transformer for Generalizable Synthetic Image Detection," arXiv:2401.xxxxx, 2024.

[8] H. Li and J. Chen, "Rethinking Image Forgery Detection via Contrastive Learning and Unsupervised Clustering," arXiv:2305.xxxxx, 2023.

[9] M. Zhao et al., "Harmonizing Image Forgery Detection and Localization," *MDPI Electronics*, 2023.

[10] S. Gupta and A. Verma, "Deep Learning-Based Image Forgery Detection System Using VGG16-UNet Model," IEEE, 2024.

[11] R. Kumar et al., "Ensuring Visual Integrity: Authentic Image Forgery Detection using EfficientFormer and BCU-Net," 2024.