

Advancing Digital Forgery Detection: A Comparative Analysis of U-Net, Buster Net, Efficient B0+ U-Net, and VGG 16+ U-Net Models

Preeti Dhiman and Usha Chauhan

Department of Electrical, Electronics and Communication Engineering, Galgotias University, Greater Noida, India

Article history

Received: 22-05-2026

Revised: 21-07-2026

Accepted: 22-08-2026

Corresponding Author:

Preeti Dhiman

Department of Electrical, Electronics and
Communication Engineering, Galgotias
University, Greater Noida, India

Email:

preeti.20sece3020008@galgotiasuniversity.edu.in

Abstract: Forgery Detection in the area of digital images is indispensable for insuring authenticity, particularly in detecting Copy-Move, Splicing. In this research, a meticulous analysis of latest learning architectures i.e. U-Net, Buster Net, Efficient B0+U-Net and VGG16+U-Net are compared for the detecting these types of forgeries. U-Net is popular due to its potent segmentation capabilities to provide pixel-level forgery localization while Efficient B0+U-Net is a lightweight variation, which aims to enhance computational efficiency along with preserving accuracy, enabling faster computation time. VGG16+U-Net utilizes pre-trained VGG 16 model as encoder to extracts hierarchical features and decoder up-samples the features while merging spatial details from the encoder through skip connections to provide faster training, better generalization, better accuracy and faster computation. Buster Net, a leading-edge network particularly designed for forgery detection, employ dual-stream processing to predict manipulated regions in an image effectively. All these models utilize IEEE IFS-TC Image Forensic Challenge dataset and CASIA V2. The evaluation of these models is done based on accuracy, F1, IoU and AUC along with other parameters, computational speed, and robustness to post-processing attacks. Experimental results demonstrate the strengths and limitations of each model, with U-Net and Lightweight Buster Net excelling in complex manipulations in terms of Evaluation metrics, U-Net and Buster Net models have given higher Accuracy ($\approx 97\%$), F1 score ($\approx 97\%$) and less computation time for both datasets as compared to other two Models. These findings highlight the trade-offs in model selection for this particular application.

Keywords: Digital Image Forensics, Forgery Detection, Copy-Move Forgery, Image Splicing Forgery, U-Net, Efficient B0+ U-Net, Buster Net, Convolutional Neural Networks (CNNs), IEEE IFS-TC Image Forensic Challenge Dataset, Image Manipulation Detection, Deep Learning, Pixel Level Localization, Lightweight CNNs, Robustness to Forgeries, Computational Efficiency

Introduction

In this modern era of ubiquitous digital media, image manipulation has emerged as paramount issue due to

profusion of accessible image editing tools. Copy-Move (CM) and Image Splicing (IS) are two most prevailing forgery techniques, and they cause great difficulties to detect. CM forgeries particularly involve duplicating a

region within the same image once or multiple times, while IS forgery combines regions from different images to create an amalgamated image. These manipulations can be annihilating and used for dissemination of misinformation. The widespread dissemination of manipulated images poses significant risks in the areas of journalism, legal evidence, social media integrity, agriculture and remote sensing, emphasizing on robust and advanced forgery detection methods. These manipulations are challenging in detection and localization of area of manipulation, especially in the case of post-processing techniques with the application of blurring or compression.

For detecting image forgeries, the complexity arises from the seamless integration of copied regions in the image (blending smoothly with the original image) or due to fine manipulation of the lighting/color and/or textures in the spliced image or due to subtle changes at the pixel level, which are hard to detect by using any conventional methods.

Conventional methods based on metadata analysis or statistical methods are often not sufficient to detect complex manipulations, particularly those manipulated or created with Generative Adversarial Networks (GANs). Therefore, advanced models like U-Net, Buster Net, Efficient B0 + U-Net, and VGG16 + U-Net are utilized for detection and learning spatial and structural relationships in the digital images.

The objective, therefore, is to develop and assess the effectiveness of these Deep Learning (DL) architectures in detecting copy-paste and image splicing forgeries or both, along with addressing the issues related to computational speed, model scalability, and robustness. By attaining these objectives, the research aims to design the cutting-edge technology in image forgery detection, leveraging deep learning models for the above-mentioned objectives.

DL Models have evolved the field of digital image forensics by providing improved performance over conventional feature-based methods. Among all, U-Net is widely applied due to its encoder-decoder architecture, allowing pixel-level localization of manipulated segments. Further, to address the balance between accuracy and computational speed, Efficient B0+ U-Net is utilized. Additionally, VGG16 + U-Net is a faster model that provides better accuracy and better generalization with pre-trained VGG16 models (ImageNet) with fewer parameters for unseen data. It also utilizes compound scaling to improve detection of subtle artifacts. Buster Net, is a dual-stream architecture specifically designed to localize forged regions. This paper aims to provide detailed analysis of proposed architectures for CM and IS forgery detection for IEEE IFS-TC Image Forensic Challenge and CASIA V2 dataset.

Proposed models, U-Net, Buster Net, Efficient B0 + U-Net, and VGG16 + U-Net, are advanced hybrid DL models intended for forgery detection and localization. The proposed U-Net model has encoder-decoder structure that makes it potent for detecting fine-grained tampering regions. The proposed Buster Net is optimized compact model to reduce computational cost while possessing strong discriminative capability for detecting and localizing diverse type of forgery. The proposed Efficient B0 + U-Net integrate EfficientNet-B0 backbone as an encoder in U-Net framework to exploit better accuracy-to-parameter efficiency and faster inference. Finally, the proposed VGG16 + U-Net utilize U-Net encoder with the pre-trained VGG16 network to enhance feature extraction through deeper hierarchical representations. Collaboratively, these models balance accuracy, efficiency, and robustness and address the issues of detecting diverse type of manipulations and generalizing across two datasets.

Literature Review

Research in image forgery detection has been transformed from conventional approaches to DL-based approaches (Table 1). Conventional methods such as Scale-Invariant Feature Transform (SIFT) and Principal Component Analysis (PCA) were predominantly based on features for detecting copy-move, delved in the work by Fridrich et al. (2003). These methods adopted feature matching and block-based comparisons but were often attuned to rotation and scaling types of transformations and struggled with transformations and post-processing noise. Ensuing work on statistical and JPEG artifact-based analysis (Chu and Li, 2019) and sensor noise or PRNU-based detection (Korus and Huang, 2017) further enhanced detection capability but manual feature design and dataset sensitivity were main limitations.

The introduction of Convolutional Neural Networks (CNNs) marked a notable transformation in image forensics, where Bayar and Stamm (2018) proposed constrained convolutional layers to suppress semantic content and prioritized on manipulation residuals, pioneering data-driven feature extraction for forensics, while Bappy et al. (2019) introduced a hybrid LSTM-CNN architecture for spatial dependency modeling. Zhou et al. (2017) manifested deep CNNs that learn manipulation-aware features for improved splicing localization. Architectures such as U-Net (Ronneberger et al., 2015), Buster Net (Wu et al., 2018), and UCM-Net (Weng et al., 2024) acclimatized encoder-decoder designs for improving dense tamper localization, and segmentation precision.

Table 1: Literature review

Reference	Year	Method	PROS	CONS
Timothy and Santra	2024	CNN-based deep learning classifier for copy-move and splicing detection	Covers two major forgery types in one pipeline; benchmarked on standard datasets	Focused on detection/classification rather than pixel-level localization
Ferreira et al.	2020	Survey of digital image forensics methods	Broad, structured overview of the field useful for positioning new work	No new method proposed; findings now dated relative to 2024–2026 work
Xu et al. (FD-GAN)	2024	GAN-based generalizable and robust forgery detector	Strong cross-dataset generalization by design	GAN training is unstable and computationally heavier than encoder-decoder CNNs
Fridrich et al.	2003	Block-matching copy-move detection (classical, non-deep-learning)	Foundational method; low computational cost, interpretable	Fails under rotation/scaling and post-processing attacks; superseded by deep methods
Ronneberger et al. (U-Net)	2015	Encoder-decoder with skip connections for biomedical segmentation	Recovers fine spatial detail lost during downsampling; became the reference architecture for dense prediction	Not originally designed for forensic cues; needs adaptation for forgery-specific artifacts
Weng et al. (UCM-Net)	2024	U-Net-like tampered-region-related framework for copy-move detection	Tailored specifically to copy-move task with region-relation modeling	Narrower scope than general-purpose forgery localization
Ahmed & Naskar	2021	EfficientNet encoder + modified U-Net decoder for splicing localization	Lightweight encoder reduces parameter count while keeping decoder localization strength	Evaluated on a narrower set of splicing scenarios; limited robustness testing reported
Wu et al. (Buster Net)	2018	Dual-stream (similarity + manipulation) network for copy-move detection with source/target localization	Distinguishes source vs. target copy-move regions, unlike single-stream models	Specialized for copy-move; less effective on splicing or other forgery types
Zhou et al. (RGB-N)	2018	Two-stream CNN combining RGB and noise features for manipulation detection	Noise-stream captures tampering artifacts invisible in RGB alone	Two-stream design increases model complexity and inference cost
Kafali et al. (RoBuster Net)	2021	Volterra-based convolutions added to Buster Net for copy-move robustness	Improves robustness to post-processing compared to base Buster Net	Added complexity from Volterra kernels increases training cost
ISO/IEC IFS-TC dataset	2019	Benchmark dataset (not a method)	Standard, widely-used evaluation benchmark for forensic challenges	Known class imbalance between authentic and forged samples
Zhao et al.	2025	Universal/generalizable feature learning for forgery localization	Targets cross-dataset generalization directly as the main objective	arXiv preprint; not yet peer-reviewed at time of citation
Joshi et al.	2022	SOTA CNN classifiers combined with Error Level Analysis (ELA)	Combines handcrafted ELA cues with learned features	ELA is sensitive to JPEG recompression and can be unreliable on non-JPEG sources
Frank et al.	2020	Frequency-domain analysis for deepfake image recognition	Reveals GAN upsampling artifacts invisible in spatial domain	Tailored to GAN-generated fakes, not general splicing/copy-move forgery
Kumar et al. (VI-NET)	2022	Hybrid VGG + Inception V3 for copy-move classification	Combines complementary feature extractors from two backbones	Higher computational overhead than a single-backbone model
Shelke and Kasana	2024	VGG-16 + KPCA for multiple forgery detection in video	Extends forgery detection from images to video domain	KPCA adds a separate dimensionality-reduction stage, complicating the pipeline
Smitha et al.	2023	Ensemble of VGG-16 and CNN architectures	Ensembling typically improves robustness over a single model	Ensemble inference cost is higher; gains over single VGG-16 not always large
Jaiswal et al.	2022	Convolutional autoencoders for document forgery detection	Effective at capturing document-specific tampering patterns	Autoencoder reconstruction approach may miss subtle, localized edits

Zhou et al. (two-stream)	2017	Two-stream CNN for tampered face detection	Early and influential two-stream design for face-specific forgery	Domain-specific to faces; not general-purpose image forgery
Chu and Li	2019	Survey of blind JPEG tampering forensics techniques	Good grounding in classical JPEG-artifact-based forensics	Survey only; predates most deep-learning forgery localization work
Zanardelli et al.	2023	Survey of deep-learning forgery detection approaches	Comprehensive, recent synthesis of CNN/segmentation-based methods	As a survey, offers no new experimental results of its own
Alencar et al.	2024	Combination of passive detection methods with neural networks	Combines multiple passive cues for more robust detection	Method fusion adds pipeline complexity relative to end-to-end models
O'Flaherty et al. (PhotoHolmes)	2025	Open-source Python library bundling several forgery-detection methods	Practical, reproducible toolkit; useful for benchmarking across methods	A software library rather than a novel detection method
Liao et al. (MSMAE-Net)	2025	Multi-semantic, multi-attention network for forgery localization	Attention mechanism improves sensitivity to subtle manipulation cues	Added attention modules increase parameter count and training time
Niloy et al. (CFL-Net)	2023	Contrastive learning for forgery localization	Contrastive objective improves separation between authentic/forged features without heavy supervision	Requires careful contrastive-pair construction; sensitive to sampling strategy
Bayar & Stamm	2018	Constrained CNN with a custom manipulation-detection convolutional layer	Learns manipulation-suppressing features directly, generalizes across manipulation types	Constrained layer adds architectural complexity vs. standard convolutions
Bappy et al.	2019	Hybrid LSTM + encoder-decoder for forgery detection	Captures spatial and boundary-context dependencies jointly via LSTM	Recurrent component increases training time vs. purely convolutional models
Korus and Huang	2017	Multi-scale PRNU-based tampering localization	Sensor-noise (PRNU) approach works even without visible tampering artifacts	PRNU is sensitive to image resizing/compression and needs camera reference patterns
Hu et al. (SPAN)	2020	Spatial pyramid attention network for manipulation localization	Multi-scale attention improves localization at varying forgery region sizes	Attention pyramid adds computational overhead relative to plain decoders
Wu et al. (robust OSN)	2022	Robustness-focused detection for social-network-shared (recompressed) images	Directly targets real-world robustness to platform recompression	Robustness gains are tuned to specific platforms' compression behavior
Zhu et al.	2024	Multi-scale ConvNeXt feature fusion for tampering localization	Modern ConvNeXt backbone improves feature quality over older CNNs	ConvNeXt backbone is heavier than lightweight encoders like EfficientNet-B0
Kumari and Garg	2025	Review of image splicing forgery detection	Recent, focused survey specifically on splicing (vs. general forgery)	Survey only; no new experimental comparison
Hao et al. (TransForensics)	2021	Dense self-attention transformer for forgery localization	Early transformer-based approach shows self-attention can aid boundary localization	Requires more training data/compute than CNN counterparts, per the ViT literature
Yu et al. (MSU-Net)	2024	Multi-scale supervised U-Net for splicing localization	Multi-scale supervision improves boundary precision	Added supervision heads increase training complexity
Fu et al. (DANet)	2019	Dual attention network for scene segmentation	General-purpose dual (channel + spatial) attention improves segmentation quality	Originally designed for scene segmentation, not forensic-specific artifacts
Xu et al. (FakeShield)	2025	Multi-modal LLM-based explainable forgery detection/localization	Adds natural-language explanation alongside localization, aiding interpretability	Relies on large multi-modal LLM backbones, high compute cost
Yu et al. (KD)	2025	Reinforced multi-teacher knowledge distillation for efficient forgery detection	Distillation reduces inference cost while retaining multi-teacher accuracy	Requires training multiple teacher models first, raising upfront training cost
Sheng et al.	2026	Weakly-supervised vision-language reasoning for	Reduces need for costly pixel-level annotation via weak supervision	Weak supervision typically trades off some localization

		forgery localization		precision vs. fully supervised models
Peng et al. (ForensicsSAM)	2025	SAM-based unified detection/localization robust to adversarial attack	Explicitly evaluated against adversarial robustness, unlike most compared methods	Built on the large Segment Anything Model, adding significant compute overhead
Li et al.	2026	Forgery detection/localization for document desensitization scenarios	Addresses a specific, practically important document-privacy use case	Domain-specific to documents; generalization to natural images unclear
Cheng et al.	2025	Curiosity-driven deep RL with sparse reward compensation for localization	Reinforcement-learning framing can handle sparse/weak localization signals	RL training is typically less stable and more sample-inefficient than supervised training
Wong et al. (ADCD-Net)	2025	Adaptive DCT feature + hierarchical content disentanglement for document forgery	DCT features target compression-domain tampering cues effectively	Tailored to document images; less validated on natural-scene forgery
Song et al.	2024	Multi-modal forgery representation learning for diffusion-generated video	Targets the emerging diffusion-video forgery problem directly	Focused on video/diffusion content, not static-image splicing/copy-move
Li et al. (prompt learning)	2025	Noise-assisted prompt learning for forgery detection/localization	Prompt-learning reduces the need for full fine-tuning of large backbones	Prompt-based methods can be sensitive to prompt design and noise-injection tuning
Zhang et al. (SE-Network)	2022	Multi-task Squeeze-and-Excitation network for splicing localization	SE blocks recalibrate channel-wise features cheaply, improving accuracy with low overhead	Multi-task setup requires balancing multiple loss terms during training
Sharma et al.	2023	Survey of detection/mitigation methods for fake images on social media	Broad platform-level perspective beyond just detection accuracy	Survey only; no new detection method benchmarked
Kadam et al.	2022	Mask R-CNN + MobileNet V1 for copy-move/splicing detection [Retracted]	Combines instance segmentation with a lightweight backbone	Article has been formally retracted; findings should not be relied upon
Guo et al. (LDFnet)	2024	Lightweight dynamic fusion network integrating local artifacts and global texture	Balances lightweight design with both local and global cue fusion	Primarily validated on face forgery, not general image splicing/copy-move
Rehman & Jain	2025	Deep CNN features with a separate ML classifier for copy-move detection	Simple, modular pipeline; classifier stage is easy to swap/tune	Two-stage design (feature extractor + classifier) is less end-to-end than unified CNNs
Liu et al. (hierarchical UNet)	2024	Hierarchical progressive detection/localization based on U-Net	Progressive/hierarchical refinement improves fine-boundary accuracy	Multi-stage hierarchy increases inference-time complexity
Dong et al. (CASIA dataset)	2013	Benchmark dataset (not a method)	Long-standing, widely-adopted benchmark enabling cross-paper comparison	Aging dataset; forgeries less representative of modern generative-AI manipulation
Akkaya et al.	2024	Local InFormation Enhancer (LIFE) module adding local inductive bias to ViTs	Directly addresses ViT's small-dataset weakness with low added compute/memory cost	Still requires architectural modification of the base ViT, not a drop-in fix
Mormille et al.	2023	Gram-matrix similarity-based self-attention regularization for ViTs	Improves ViT convergence/generalization on mid-size datasets via a loss-function-only fix	Benefit shown mainly on mid-size, pre-trained settings, not from-scratch tiny datasets like ours
Shi et al. (MEAFormer)	2025	Multi-exit transformer architecture for robust forgery localization	Reduces reliance on large-scale pre-training, addressing a core ViT limitation for this task	Multi-exit architecture adds design/training complexity vs. a single-exit CNN decoder
Pawar et al.	2025	ViT classification + fine-tuned SAM for forgery localization	Directly tested on CASIA v1/v2 and related benchmarks, enabling direct comparison	Two-model pipeline (ViT + SAM) increases inference cost vs. a single encoder-decoder

Ahmed and Naskar (2021) presented an Efficient U-Net with downsized computational complexity while Wu et al. (2022) designed models for social-media compressed and resized images. Kafali et al. (2021) worked on improving robustness with post-processing distortions using Volterra-based convolution filters. Hybrid architectures like VI-Net (Shelke and Kasana, 2024) (VGG + InceptionV3) combined local and global features, whereas ensemble and auto-encoder methods (Smitha et al., 2023; Jaiswal et al., 2022) utilized feature fusion and unsupervised learning for improving generalization. These CNN-based models achieved higher accuracy while faced challenges in generalization and localization in manipulated images.

Joshi et al. (2022) utilized ELA along with state-of-the-art CNN classifiers to provide visual interpretability and strong accuracy on standard datasets but had sensitivity towards compression artifacts and less effectiveness against GAN-generated forgeries. Shelke and Kasana (2024); Smitha et al. (2023) proposed CNN-based models for video forgery detection using VGG-16 and Kernel Principal Component Analysis (KPCA) but consistency was the major challenge.

Xu et al. (2024) proposed FD-GAN, an adversarial trained model for improving generalization against unseen data, while Zhao et al. (2025); Niloy et al. (2023) delved into domain-agnostic forgery detection through contrastive and universal feature learning. Meanwhile, Wu et al. (2022); Zhu et al. (2024) captured finer manipulation cues, balancing accuracy and computational cost by utilizing multi-scale fusion and ConvNeXt architectures. Surveys done by Ferreira et al. (2020); Timothy and Santra (2024); Zanardelli et al. (2023); Kumari and Garg (2024) provided analysis of these CNN-based hybrid approaches, highlighting generalization, robustness, and interpretability. Frank et al. (2020) exploited GAN-generated images for exhibiting tell-tale high-frequency artifacts in the frequency domain, and explored robust detector by utilizing many spatial-domain methods.

Recent studies included attention mechanisms/transformer-based architectures to enhance representation learning where Hao et al. (2021) for forgery localization introduced TransForensics, a dense self-attention transformer, which produces interpretable attention maps along with improved localization accuracy. Fu et al. (2019); Zhang et al. (2022) for boundary-aware splicing localization exploited dual attention and squeeze-and-excitation mechanisms in proposed model. Yu et al. (2025) used multi-scale supervised U-Net (MSU-Net) for achieving high F1 and IoU scores for multiple datasets along with strong fine-grained detection capabilities but incurred higher computational demands.

Xu et al. (2025) introduced FakeShield, a large-language-model-based framework for image forgery

localization which includes visual and textual reasoning while Peng et al. (2025) for forensic use adapted the Segment Anything Model (SAM) for achieving strong cross-domain generalization and robustness to adversarial manipulations. Sheng et al. (2026), proposed vision language reasoning to reduce annotation requirements and Li et al. (2025) presented Noise-Assisted Prompt Learning with noise-aware prompts to improve robustness in multimodal detection systems. These large-scale pre-training models require high computational resources but provide good generalization.

Recent works have addressed document forgery (Li et al. 2026; Jaiswal et al. 2022) and privacy-preserving detection (Rehman and Jain, 2025; Liu et al., 2024), Reinforcement-learning-based exploration methods (Cheng et al., 2025) and dynamic fusion approaches (Guo et al., 2024) have also been proposed to adaptively detect complex or subtle manipulations. Despite these advances, current research continues to face limitations in real-time applicability, performance on low-quality or highly compressed images, and the ability to adapt to novel manipulation patterns. Zhang et al. (2022) proposes a dual-stream convolutional encoder-decoder network with an edge-guided and mask stream, alongside squeeze-and-excitation modules and low-/high-level feature fusion, to more accurately localize spliced regions in images.

In summary, this literature review gives deep insight into the evolution from conventional to deep CNNs, followed by attention-based and multimodal transformer architectures. Early CNNs such as constrained-convolution models provided strong baseline performance, but generalization was a huge problem. The proposed models provide high performance with better generalization. The proposed models are chosen over ViT based models (Shi et al., 2025; Pawar et al., 2025) because Image Forgery localization is fundamentally a per-pixel boundary problem where the model has to decide, pixel by pixel, region of tempering. ViT's lacks the inductive biases and requires either large-scale pre-training or an explicit architectural modification. Thus, ViT's global self-attention is more prone to producing coarse, poorly-localized boundaries on a small dataset (Akkaya et al., 2024; Mormille et al., 2023) whereas Encoder-decoder architectures with skip connections are the established mechanism for these specific problems. U-Net's encoder-decoder structure with skip connections was developed specifically to recover fine spatial detail lost during downsampling, and remains the reference architecture for dense, pixel-level prediction tasks (Ronneberger et al., 2015). EfficientNet-B0 and VGG16 based model with the same decoder structure is proposed specifically to test whether a different encoder (one optimized for parameter efficiency, the other for depth) changes localization quality while keeping the decoder mechanism constant.

Methods

Implementation of U-Net Model

U-Net is essentially a CNN architecture substantially used for image segmentation tasks. U-Net Architecture (Akkaya et al., 2024) comprises of Encoder (Contracting Path) to obtain the features by down-sampling through convolutional and pooling layer from the input image while Decoder (Expansive Path) retrieves the spatial image information through up-sampling, it also combines encoder extracted features through skip connections with precise predictions and for transferring high resolution features to the decoder which allows the model to generate accurate segmentation maps, along with fine details.

U-Net Model Architecture has been modified slightly as shown in Fig. 1 to make it suitable for forgery detection tasks with Input Layer accepting images of 512 X 512 X 3 pixels as input.

Further is the Encoder (Contracting Path) which consists of Convolutional layers with activation function Rectified Linear Unit (ReLU) to extract input image features followed by a batch normalization and a max-pooling layer with the spatial dimensions reduction. The image is down-sampled by each layer to capture higher-level features and Bottleneck layer is the deepest part of the network and captures the most abstract features. Decoder (Expansive Path) is used for up-sampling, and transposed convolutions or up-sampling layers increase the image resolution while retaining important features. Skip Connections allows up-sampled features at corresponding levels from the encoder, allowing the network to recover spatial details lost during down-sampling by the decoder. A binary mask from final convolutional layer produces the segmentation map indicating the manipulated regions of the image. At last, sigmoid activation function provides pixel-wise classification.

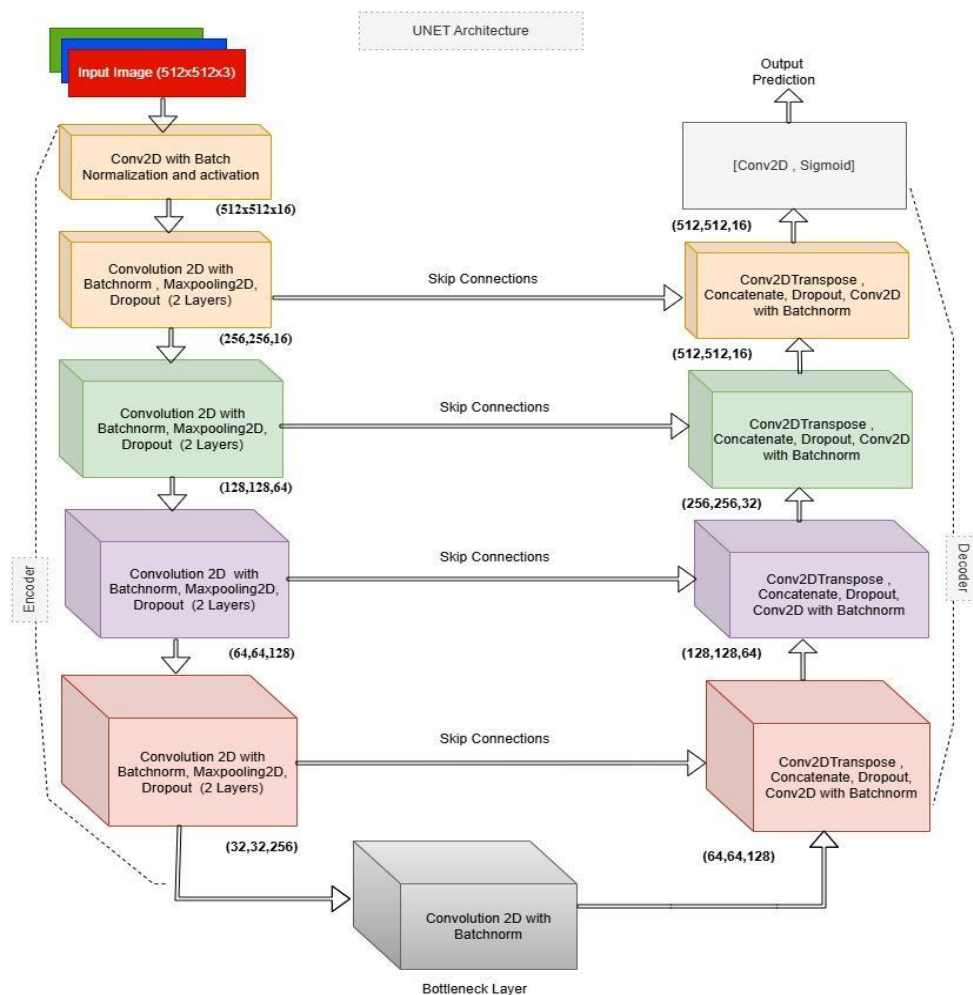


Fig. 1: U-Net architecture

Model Training: Data Splitting is done with the dataset divided as: 60%: Training, 20%: Validation and 20% testing with network's weights during train update using an ADAM optimizer to minimize the loss function. With labeled dataset, model is trained and loss is minimized over each epoch iteration along with early stopping which is used to prevent over-fitting by monitoring the validation loss.

For Model evaluation, Evaluation Metrics are used and after detecting manipulated regions, an additional post-processing technique as threshold is applied to enhance the results for clearer identification of forged areas.

Implementation of Buster Net Model

Buster Net is a basic three-branch DL Model (Fig. 2) designed for Image Forgery Detection and localization with input layer accepting images of $512 \times 512 \times 3$ size. Buster Net processes each input through two parallel branches, first is similarity-detection branch $S(x)$ which extracts copy-move patterns via self-correlation, and then manipulation-artifact branch $M(x)$ that picks up on tampering traces like resampling and JPEG-block inconsistencies. The two branch outputs are fused as:

$$F = S(x) \oplus M(x) \quad (1)$$

Where $\oplus$ is channel-wise concatenation, F went through shared decoder head to produce the final mask, structurally equivalent to how the U-Net variants take $D_i = fup(D\{i-1\}) \oplus E\{5-i\}$ through their own decoder. Buster Net concatenates two independently-computed branches at one point before decoding while the U-Net

variants concatenate encoder and decoder features across a single stream splitting and rejoining.

Following are the three key components with specific functions to provide better accuracy for copy paste and Image forgery detection:

1. S-Net (Similarity Network) are based on a Convolutional Neural Network (CNN) architecture, which is used for comparing patches in an image to measure their similarity. These are particularly used to detect copy-move forgeries. It is similar to Siamese network structure, where feature maps from two patches are compared to compute pixel-level similarity. The network generates patches on the feature map where copied and moved regions are highly correlated, which is known as the similarity mask representing duplicated regions. Thus, the output of S-Net is represented as a similarity mask that highlights duplicate regions, and the size of the S-Net output is $512 \times 512 \times 128$, which is further given to F-Net.
2. M-Net (Manipulation Network) is basically a ResNet-based encoder-decoder network that particularly focuses on splicing forgeries, i.e., it detects boundary inconsistencies, Color or contrast mismatches, blending artifacts, and noise-level inconsistencies. It basically extracts sharp edges, blending artifacts, or any difference in contrast in an image to detect manipulations, i.e., and further localize the altered regions. The squeeze-and-excitation (SE) blocks focus on extracting manipulation-related features. The output layer employs a ReLU activation followed by post-processing that converts the response into a forgery localization map to detect and splice the forgery region.

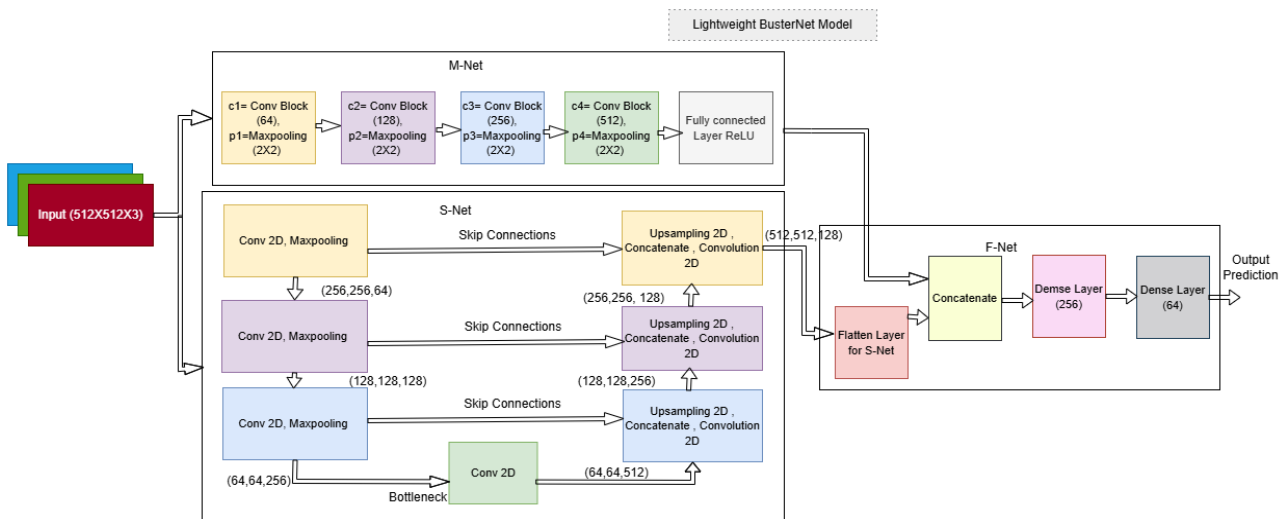


Fig. 2: Buster Net architecture

3. F-Net (Fusion Network) integrates S-Net and M-Net outputs for final prediction. It utilizes a similarity mask from S-Net and a manipulation mask from M-Net. As the output of S-Net is 512X512X128, F-Net uses a flatten layer to convert the output to match the size with the M-Net model. It further utilizes a concatenate layer (skip connections) to restore the spatial details that were lost during down-sampling by the decoder. It uses 2 dense layers, where the second dense layer has a sigmoid function as the final layer to predict the forgery. Thus, the resultant forgery localization mask combines spatial, structural, and frequency information from both networks effectively

Implementation of Efficient U-Net (EfficientNet B0+ U-Net) Model

The EfficientNet B0 model (EfficientNet-based Bottleneck Optimized Model), as shown in Fig. 3, is a lightweight, high-performance CNN that is designed for achieving balance between accuracy and computational speed with effective forgery detection and localization. The base model EfficientB0 inherits the foundational design of EfficientNet-B0, which further utilizes Mobile Inverted Bottleneck Convolution (MBConv) blocks. Each MBConv block basically consists of:

- a) Depth-wise separable convolutions, which reduces number of parameters and computation into depth-wise and point-wise operations
- b) Squeeze-and-Excitation (SE) modules, adapts to recalibrate the channel-wise feature responses to focus on informative regions
- c) Residual connection enables efficient gradient flow, further reduces vanishing gradient problems in deeper networks

All these modules in the proposed model are effective in capturing fine-grained spatial and texture-level details, which are critical for image forgery

localization. The Efficient B0 model helps to reduced parameter count, reduces parameter count and enhances generalization in performance on small or imbalanced datasets

EfficientNet B0 in the proposed model is a feature extractor for identifying tampered images. When coupled with feature fusion layers, it can localize forgery regions at the pixel level. Its architecture is particularly suitable for detecting manipulations such as copy-move and splicing, which requires precise recognition of texture-level discrepancies. EfficientNet-B0 outputs a single classification vector and has no decoder to produce a pixel-level mask, so it is paired with a U-Net decoder specifically to enable the spatial localization. A combination of EfficientNet-B0 and U-Net provides an organized architecture to offer global feature extraction by using EfficientNet-B0 (the basic architecture is shown in Fig. 3) and precise localization provided by U-Net for image forgery detection, specifically with pixel-wise tampering detection for detecting CM and IS forgeries. It follows two-stream approach with EfficientNet-B0 to extract high-end multi-scale hierarchical features for classification and U-Net as decoder for reconstructing a segmentation mask that localizes tampered regions at the pixel level. Hence, this architecture allows high accuracy in forgery classification and precise localization of manipulated regions. The model (Fig 4) consists of an input layer with an RGB image having an image size of 512x512x3, followed by preprocessing with normalization and data augmentation (i.e., flipping, rotation, Gaussian, grid distortion, elastic transform, optical distortion, random brightness, etc.) for improving robustness in the model. Next is the Feature Extraction Layer, which consists of EfficientNet-B0 as the encoder. It is lightweight because of MBConv and SE layers to improve extraction of essential features and captures multi-scale hierarchical features, which are beneficial for detecting forgeries. Few modifications are done in the base efficient U-Net model for its use in forgery detection.

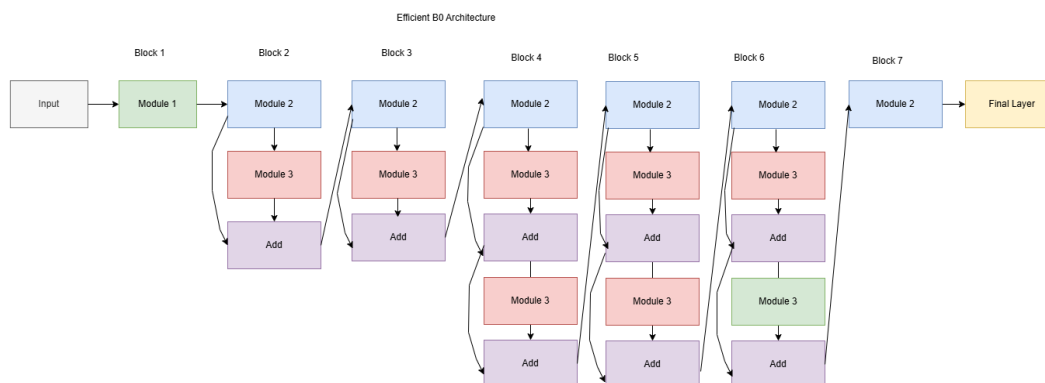


Fig. 3: Basic efficient Net-B0 architecture

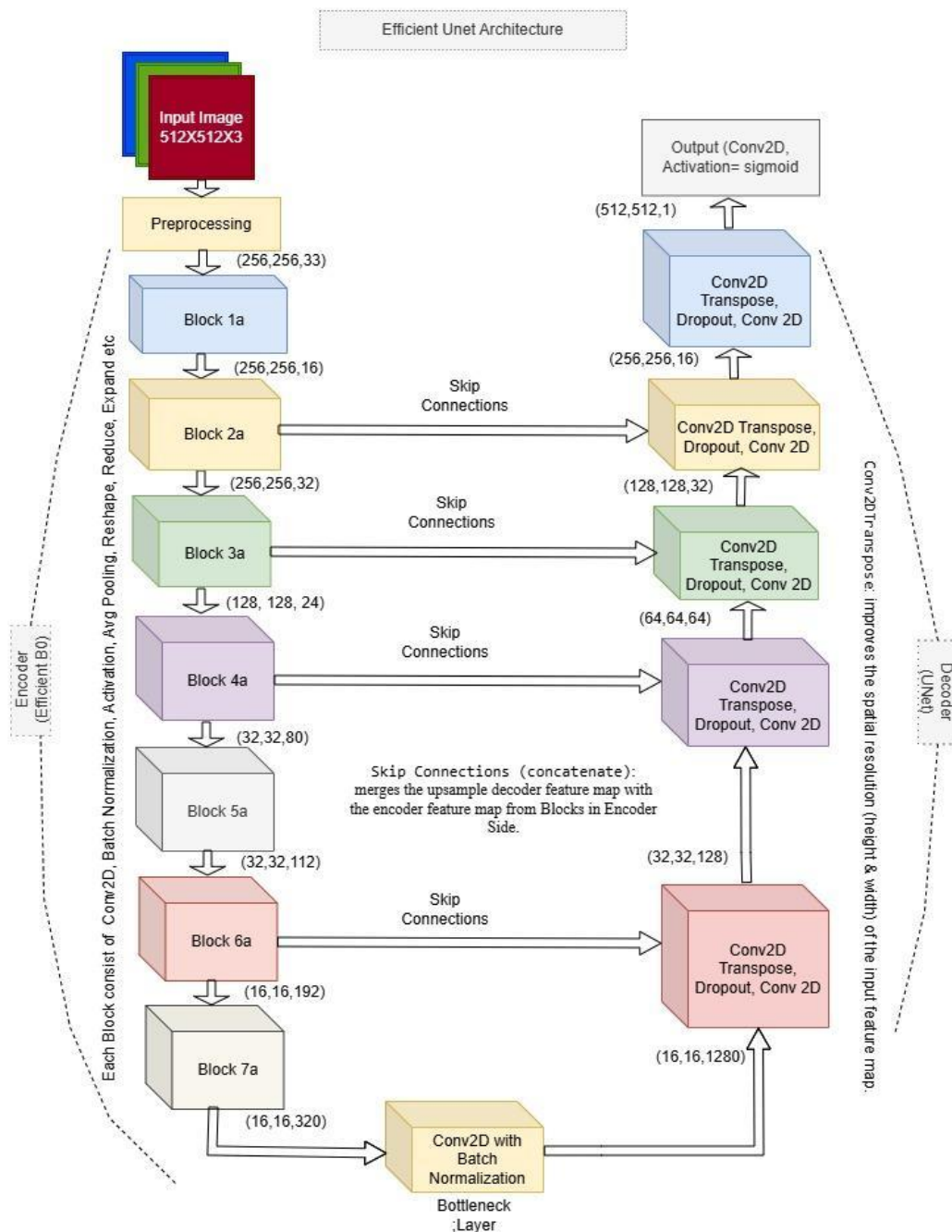


Fig. 4: Proposed efficient B0+ U-Net architecture

The model eliminates fully connected layers for classification and rather utilizes skip connection for U-Net. The first few layers of EfficientNet-B0 capture low-level features to extract texture and edge inconsistencies in forgeries. Middle layers extract deep or complex features to capture global manipulations such as unnatural patterns. It reconstructs a pixel-wise tampering map that highlights the forged areas by using the up-sampling blocks that

progressively refine the tampered region's boundaries. Also, skip connections are utilized to retain fine details.

The U-Net decoder structure consists of up-sampling blocks, i.e., de-convolutional layers, which gradually increase spatial resolution. Concatenation with corresponding encoder layers (skip connections) helps to retain the low-level features useful for texture inconsistencies.

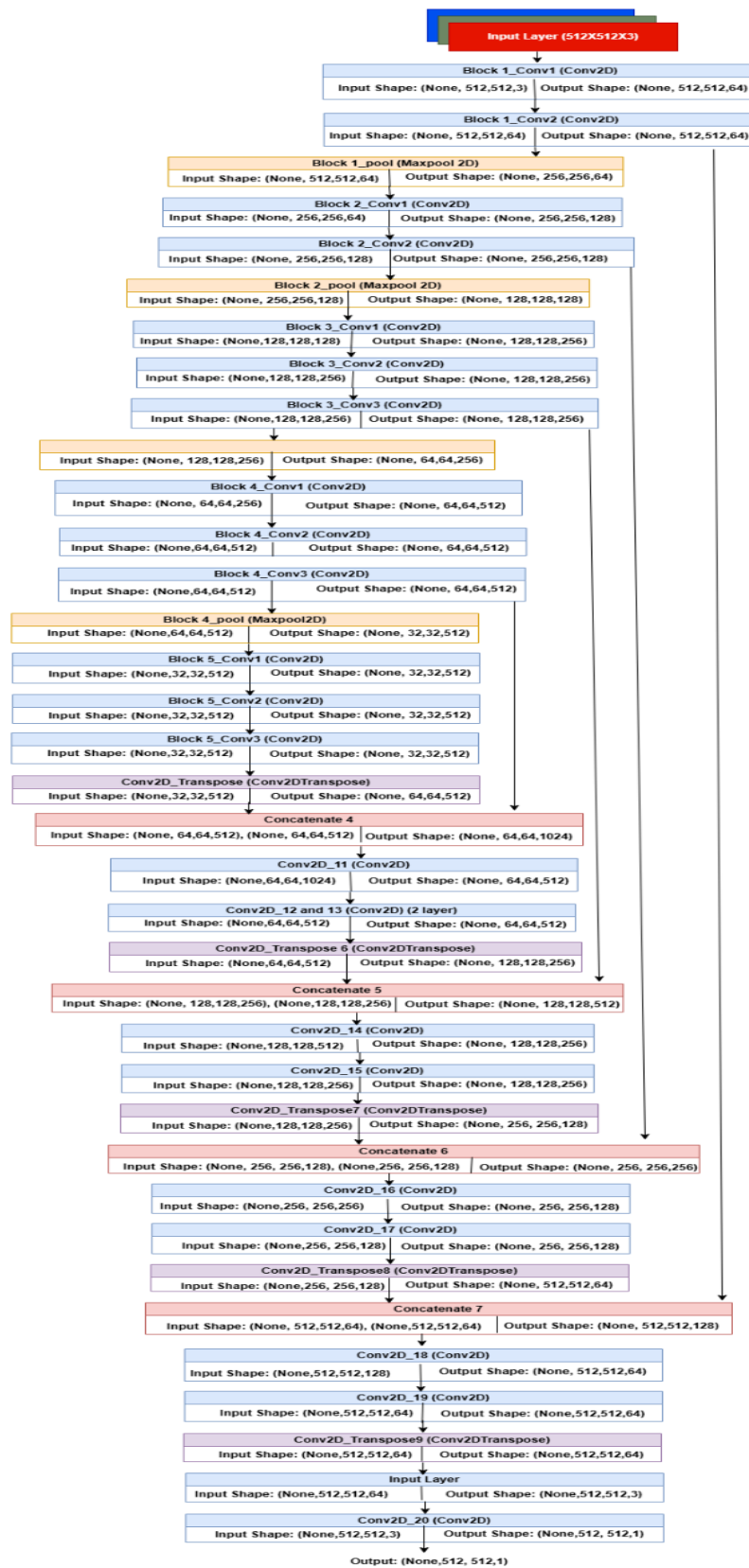


Fig. 5: VGG16 + U-Net architecture

The final output layer uses sigmoid activation for binary segmentation and produces output as forged vs. authentic pixels.

Implementation of VGG16 + U-Net Architecture

This model as shown in Fig. 5 combines VGG16 (pre-trained) as an encoder with image segmentation performed by U-Net decoder. The VGG16 + U-Net architecture is a hybrid DL model combining the feature extraction capability of VGG16 for spatial reconstruction of U-Net, which makes it highly effective for pixel-level image segmentation and forgery localization tasks.

In this architecture (Table 2), Input Shape is $512 \times 512 \times 3$ is given as input for further processing. The encoder, which is derived from the pre-trained VGG16 network's convolutional layers, extracts the hierarchical features using a sequence of convolutional layers and provides faster training and better feature extraction. Blocks 1 and 2 consist of 2 convolutional layers along with Max-pool layer to extract low-level textures, corners, and gradients in an image, respectively.

The term Conv2D (2D Convolutional Layer) refers to the building blocks of CNNs, especially to classify image, object detection, and forgery detection/localization. It is responsible for extracting spatial features from an input image or feature map. Block 3 extracts mid-level object components, while Block 4 captures deeper semantic structures, and then progressively integrates mid-level and fine-grained features from previous blocks (3, 2, and 1). The final step outputs a full-resolution prediction map ($512 \times 512 \times 32$) that captures both global and local information for accurate pixel-wise classification, and Block 5 serves as a bottleneck encoding the most abstract representation with an output shape of ($16 \times 16 \times 512$).

The decoder (Table 3) mirrors this process in reverse, using transposed convolution (Conv2DTranspose) layers for gradually upsampling feature maps back to original image resolution. Conv2DTranspose (Transposed Convolution / Deconvolution) layer in deep learning is used for increasing the spatial dimensions of feature map. It is basically the opposite of Conv2D, which is used in decoder parts of encoder-decoder models like U-Net, autoencoders, GANs, etc. Skip connections are used to restore spatial details lost during downsampling. Specifically, the decoder starts by upsampling the bottleneck.

Mathematically, the decoder step can be represented as:

$$D_i = \text{fup}(D_{i-1}) \oplus E(5-i) \quad (2)$$

- Here fup = transposed convolution (used to up-sample the previous decoder output $D(i-1)$)
- $\oplus$ denotes channel-wise concatenation, where up-sampled decoder features are combined with the corresponding encoder feature map $E(5-i)$ through skip connection
- D_0 is the bottleneck output and recursion start from there

The output segmentation mask mask comes from a 1×1 convolution on the last decoder output, followed by sigmoid or SoftMax:

$$\hat{y} = \sigma(W * D_5 + b) \quad (3)$$

- $\hat{y}$ — predicted output mask
- D_5 — output feature map of the final (fifth) decoder block, at full input resolution
- W — learned weight tensor of the 1×1 convolution, shape ($1 \times 1 \times C_s \times 1$) for binary output
- b — learned bias term of the 1×1 convolution
- $*$ — standard 2D convolution operation
- $\sigma(\cdot)$ — sigmoid activation for binary (forged vs. authentic) prediction, or SoftMax when multi-class output is used

This strategy allows to make a balance between semantic understanding and spatial precision, which helps in effectively identifying manipulated regions in forged images. Overall, the VGG16 + U-Net model utilizes pretrained VGG16 filters for accelerating convergence and enhancing generalization. U-Net's skip connections provide fine details for forgery boundary detection; thus, the architecture demonstrates high localization accuracy and strong feature reuse, but is sensitive to input resolution and computationally exhaustive.

Overall, it provides a powerful framework for image forgery analysis with effective integrating global context and detailed pixel-level reconstruction.

Table 2: Encoder in VGG16 + U-Net architecture

Block	Layer Name	Output Shape	Description
Block 1	conv1 → conv2	(256, 256, 64)	Extracts low-level textures, i.e., edges, colors, etc.
Block 2	conv1 → conv2	(128, 128, 128)	Detects more complex patterns (e.g., corners)
Block 3	conv1 → conv3	(64, 64, 256)	Extracts mid-level object features
Block 4	conv1 → conv3	(32, 32, 512)	Captures deeper semantic information
Block 5	conv1 → conv3	(16, 16, 512)	Bottleneck with highest abstraction

Table 3: Decoder in VGG16 + U-Net architecture

Step	Operation	Output Shape	Description
Step 1	Conv2DTranspose (512, strides = 2) + Skip (Block 4)	(32, 32, 512)	Upsamples to 32×32 and combines deep features
Step 2	Conv2DTranspose (256, strides = 2) + Skip (Block 3)	(64, 64, 256)	Adds mid-level object features
Step 3	Conv2DTranspose (128, strides = 2) + Skip (Block 2)	(128, 128, 128)	Restores detailed textures
Step 4	Conv2DTranspose (64, strides = 2) + Skip (Block 1)	(256, 256, 64)	Refines spatial structure
Step 5	Conv2DTranspose (32, strides = 2)	(512, 512, 32)	Restores full image resolution

The Bottleneck Layer (Block 5) is very important, as it is used for extracting feature maps from the VGG16 module. Further, the decoder (U-Net) is used to up-sample the feature map back to size 512x512 with help of transpose convolutions (Conv2DTranspose) and skip connections to recover the spatial details for precise segmentation and prevent excessive blurring. Also, each step in the decoder includes Conv2D layers to extract refine features after upsampling. The efficiently designed up-sampling model reduces information loss compared to direct interpolation. The output layer of the VGG16 + U-Net architecture consists of Conv2D (1, (1,1)) with the activation function as sigmoid to produce a single-channel segmentation mask to define the output as forged or authentic image. This architecture is congenial for Image forgery localization also, where localization of fine details is crucial and more generalization for unseen data with less computation time.

Results

Datasets Used

In the IEEE IFS-TC Image Forensic Challenge dataset as shown in Fig. 6a, 1,474 images are considered for training, testing, and validation of all the models, out of which 1050 are authentic while 424 are fake images (approx. 2.5:1). Splitting was done at the image level, before any patch extraction and each full image was assigned to train, validation, or test first, and only then patches were obtained from image through splitting. No patch from a given source image appears in more than once. Since data is imbalanced,

Augmentation (Flipping, Transpose, resize optical distortion, brightness contrast, grid distortion, etc.) were applied only after splitting, and only to the training. Validation and test images were left untouched, so evaluation metrics reflect performance on completely unseen data. Binary mask is also available for these fake images to train the model. The sample of fake image along with its sample binary mask is shown in Fig. 7. The second dataset used is CASIA V2 Mormille et al. (2023) as shown in Fig. 6b with 12,614 images, having authentic (original) and tampered (manipulated) categories. Out of these images, the cleaned dataset has 9895 images with 5937 images are used for training of proposed models, while 1979 are used for testing and validating of models individually. The training set was used for learning while validation set was used during training to tune the models and to decide early stopping.

The test set was used once, at the end, to evaluate final performance. Like IEEE IFS-TC, this split was done at the image level before any patch extraction, so no image (or its patches) appears in more than one split. The total images are divided into 60%: Training, 20%: Testing, and 20%: Validation. The model learns patterns, features, etc. from the training dataset by adjusting weights of its internal parameters (weights). The modeling in training process updates weights for minimizing loss function. However, for generalization and tuning hyper-parameters, a validation dataset is used to avoid the condition of overfitting or under-fitting and further tested on testing (unseen) data.

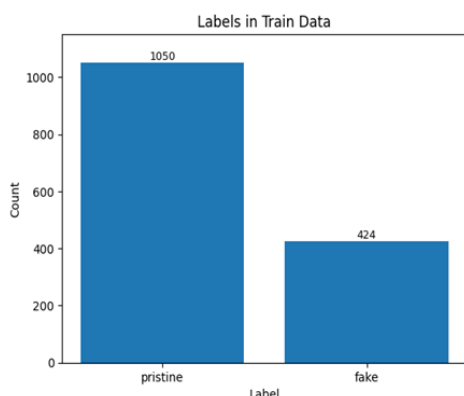


Fig. 6 a: IEEE IFS TC Dataset

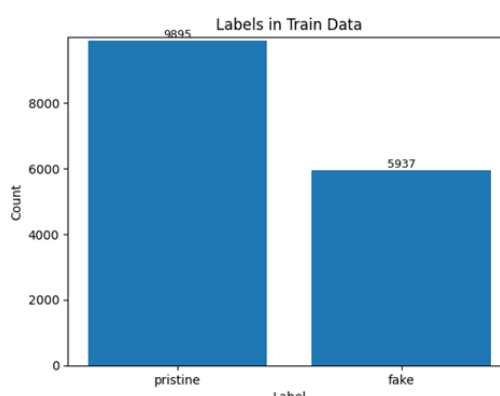


Fig. 6 b: CASIA V2 Dataset

Fig. 6 a-b: Bar graph showing imbalance in dataset

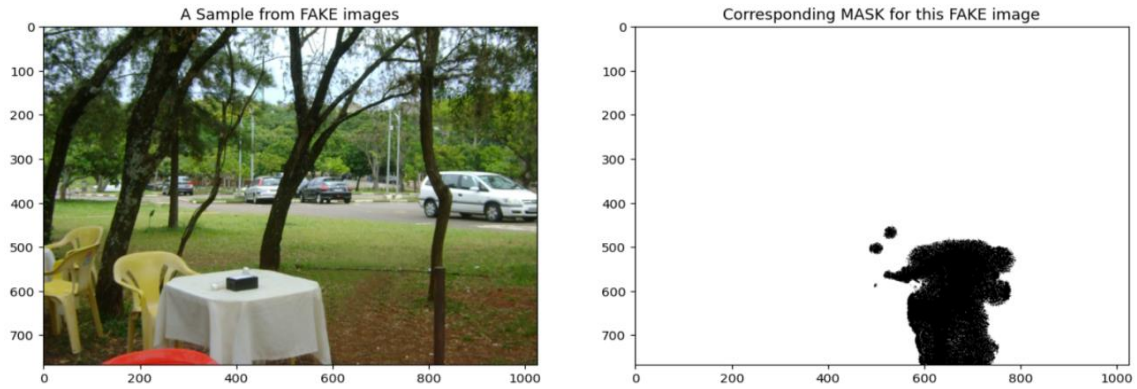


Fig. 7: Fake image with its binary mask

Hyperparameter Tuning

Optimizer

The AdamW optimizer has been employed owing to its stability with encoder-decoder architectures with batch normalization that benefit from adaptive moment estimation. However, SGD with Momentum has also been used but the convergence was slow with high computation time, as the number of epochs was drastically increased for training model.

Learning Rate and Weight Decay

Initially, 1×10^{-4} is used with 0.0005 weight decay to prevent overfitting and to maximize generalization in models.

Batch Size and Epochs

Training is conducted for 8-20 epochs with early stopping applied and it has been observed that Memory constraints limited the batch size to 4. To avoid crashing of RAM, the proposed models were trained at Batch Size = 4.

Loss Functions

To handle severe class imbalance between forged and authentic pixels, Weighted Binary Cross-Entropy (WBCE) combined with Dice loss has been used. Binary Cross-Entropy (BCE) loss provides pixel-wise binary classification, while Dice Loss measures overlapping in between predicted and ground truth masks. It is especially effective for imbalanced data. This hybrid formulation of Loss function enhanced the detection of subtle manipulation boundaries while maintaining global consistency.

The formula for WBCE is

$$(y, \hat{y}) = -[w_p * \log(\hat{y}) + w_n * (1 - \hat{y}) * \log(1 - \hat{y})] \quad (3)$$

Where y = true label (0 or 1), $\hat{y}$ = predicted probability w_p = weight for positive class, w_n = weight for negative class, $\log$ = natural logarithm.

The formula for Dice loss is:

$$LDice(y, \hat{y}) = 1 - Dice(y, \hat{y}) \quad (4)$$

$$LDice(y, \hat{y}) = \frac{\sum_{i=1}^N y_i \hat{y}_i + \epsilon}{\sum_{i=1}^N y_i + \sum_{i=1}^N \hat{y}_i} \quad (5)$$

Where:

- y_i = Ground truth pixel (0 or 1)
- $\hat{y}_i$ = Predicted probability for pixel
- N = Total number of pixels
- ϵ = Small constant

Standard cross-entropy (which in general used in models) minimizes a per-pixel classification error under an implicit assumption of class balance and does not hold in forgery localization, where forged pixels constitute a small minority of the spatial domain; consequently, gradient contributions from the majority (authentic) class dominate optimization and bias the decision boundary toward under-segmentation of manipulated regions. The weighted BCE term rescales the per-class loss contribution to counteract this prior imbalance, while the Dice coefficient, defined over the overlap between predicted and ground-truth masks, imposes a region-level structural constraint that is largely invariant to pixel-count disparity. The composite objective therefore jointly enforces pixel-wise classification fidelity and spatial coherence of the predicted mask, addressing complementary failure modes that neither term resolves independently and improves the performance.

Data Augmentation

Data augmentations, including random Gamma, rotation; Horizontal and Vertical flipping, optical Distortion, Grid distortion, Transpose, Elastic

Transform, random Brightness Contrast, and Resizing, were incorporated. Quantitative analysis revealed that augmentation improved accuracy and F1-score by approximately 4% and IoU by 3.5%, particularly on partially forged samples.

Training Environment

All the proposed models are run on NVIDIA Tesla T4 GPU is versatile and energy-efficient graphics processing unit designed primarily for deep learning neural networks and light training workloads. It is built with 16 GB, GDDR6 memory making it suitable for training small to medium-sized models, transfer learning, and performing high-speed inference. The benefit of T4's support for mixed-precision training is that it helps to accelerate the model training and to reduce memory usage.

The following section explains the graphs of Loss function, Accuracy and other evaluation metrics of each proposed model. The basic expression of these metrics is as follows:

- Pixel-wise accuracy: represents ratio of correctly classified pixels to total number of pixels:

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (6)$$

Where TP: True Positives, TN: True Negatives, FP: False Positives, FN: False Negatives:

- Precision, Recall, and F1-Score: To handle data imbalance and emphasize model robustness in detecting forgery regions, Precision, Recall, and F1-score are computed as follows:

$$Precision = \frac{(TP)}{(TP+FP)} \quad (7)$$

$$Recall = \frac{(TP)}{(TP+FN)} \quad (8)$$

$$F1 = \frac{(2 \times (Precision \times Recall))}{(Precision + Recall)} \quad (9)$$

The F1-score provides a harmonic mean of Precision and Recall. It is the most effective metrics for the model for imbalance dataset.

Intersection Over Union (IoU)

IoU /Jaccard Index measures the spatial overlap in between the predicted and ground truth masks:

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

High IoU parameter indicate strong pixel-wise agreement between prediction and truth.

Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC).

The ROC relates True Positive Rate (TPR) and the False Positive Rate (FPR) at different thresholds and are used as classification thresholds, visualizing the balance between sensitivity and specificity:

$$TPR = \frac{TP}{(TP+FN)}, FPR = \frac{FP}{(FP+TN)} \quad (11)$$

The AUC indicates area under the curve, providing a threshold-independent measure of classification ability (indicates better overall discrimination between tampered and authentic regions):

$$AUC = \int_0^1 TPR(x) d(PR(x)) \quad (12)$$

Results for Proposed Models

Results for IEEE IFS-TC Dataset: U-Net architecture has proven effective for image forgery detection, particularly in Copy-Move and Image Splicing forgery, due to its ability to localize tampered regions at the pixel level.

U-Net architecture provides accuracy in detection labeled as forged/authentic. In Fig. 8, the blue line (training loss) in the loss graph steadily remains low and indicates well trained model. The red line/ validation loss drops quickly and stabilizes close to the training loss. In the accuracy graph, the blue line/training accuracy starts from 78% and rises up to 95.14%, which represents the fitting of the training data well. The red line (validation accuracy) shows an initial drop in accuracy but rapid improvement after epoch 3 starts, and by epoch 7, it attains accuracy of 95.79%. As Training and validation in both the functions have stable and high-performing convergence. Thus, the model provides good generalization and no issues of overfitting.

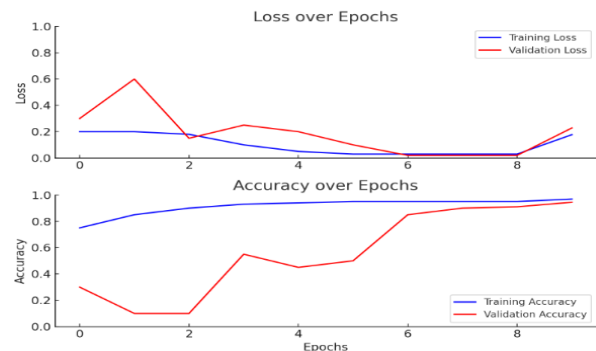


Fig. 8: Loss and accuracy graph for U- Net model

In Fig. 9, blue line (Training Loss) in the Loss Graph starting at 0.43 in epoch 1 and gradually decreasing up to 0.24 by 10th epoch, indicating the model is learning well on the training set. However, at final epoch, it slightly increases to 0.22 but still is quite low. Validation loss (red line) initially starts at 0.53, higher than the training loss, and finally converges at 0.15 (low).

It may fluctuate initially due to unstable early learning but eventually stabilizes (no over fitting) and reaches a final value of 0.22, which indicates good generalization for both losses. Accuracy graph, the blue line/Training Accuracy starts at 75% and climbs up to 96.89%. The curve flattens near the end, indicating the model has nearly converged on the training dataset while Validation accuracy, as shown in red line, starts low with nearly 30% and dips at 10% at 2nd epoch but finally stabilizes at 94.48% and remains constant till the last epoch, showing good generalization as validation accuracy is near training accuracy.

The Blue Line (Training Loss) in the Loss Graph, as shown in Fig. 10, represents model learning and fitting is better for training data with each epoch. Drop starts from 0.6, and the final value achieved is 0.21. The Red Line (Validation Loss) decreases with good generalization in early epochs and finally converges at 0.32. However, there is a small fluctuation from epoch 6 to 9 due to the randomness of data, but since the variation is small, it does not represent any sign of overfitting, thus the model is learning effectively. In the accuracy graph, the blue line (training accuracy) increases from 75% to 94.28%, representing well learning of model for training data. The Red Line (Validation Accuracy) initially jumps from 50% to 96.29 % in 1 epoch, representing a model that learned fast. High and stable accuracy for both training and validation data represents good generalization. Accuracy achieved for training is 94.28% and validation data is 96.29%.

The Blue Line (Training Loss) as shown in Fig. 11 represents loss graph which starts with a very high value of 0.95 and decreases steadily to 0.22 which represents that model is learning progressively. Red Line (Validation Loss) follows training loss closely however; it is slightly lower at all points and converges at 0.23. Overall, the loss function graph shows no fluctuation and no divergence which indicates stable and consistent validation performance and represents that it generalizes well. In Accuracy graph, Blue Line (Training Accuracy) starts from 5% and progresses to 96.96% by epoch 2 which represents fast learning on training data. Red Line (Validation Accuracy) reaches perfectly classifies the validation set early on. 96.26% after the 1st epoch and retains the same value.

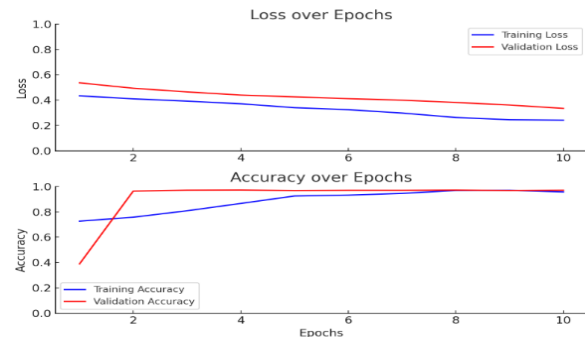


Fig. 9: Loss and accuracy graph for buster net model

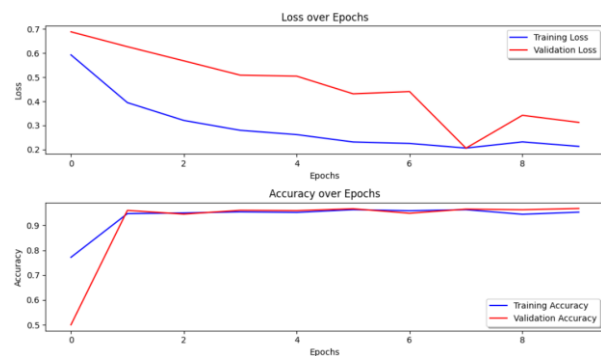


Fig. 10: Loss and accuracy graph for efficient B0+ U-Net model

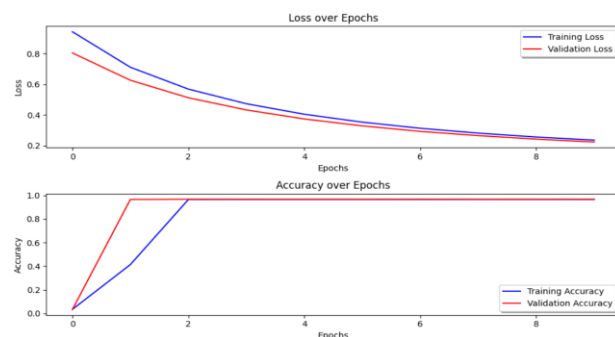


Fig. 11: Loss and accuracy graph for VGG 16+ U-Net model

Results for CASIA V2 Dataset

U-Net architecture has proven effective, particularly in both datasets, due to its ability to localize tampered regions at the pixel level. In Fig. 12, accuracy graph, the blue line /training accuracy) starts from 93.6% and rises up to 97.80%, this represents the fitting of the training data well. The red line (validation accuracy) shows rapid improvement after epoch 3 starts, and by epoch 7, it attains accuracy by 96.07%. Training and validation in both the functions have stable and high-performing convergence. Thus, the model provides good generalization and no issues of over-fitting.

The blue line (training loss) in the loss graph steadily remains low, and the red line (validation loss) drops quickly and stabilizes above the training loss.

In Fig. 13, accuracy graph, the blue line/training accuracy increases from 60% to 96.19% representing model is learning well on training data. The Red Line /Validation Accuracy initially jumps from 30% to 94.3% with the model learning fast indicating good generalization. The accuracy achieved 96.19% and 94.3%, for training and validation data respectively. The blue line/training Loss in Loss Graph starts at 0.6 in epoch 1 and gradually decreases to 0.046 by epoch 8, indicating the model is learning well on training set. Validation Loss/red line initially starts at 0.53, higher than the training loss and finally converges at 0.15 (low).

It may fluctuate initially due to unstable early learning but eventually stabilizes (no overfitting) and reaches a final value of 0.22 which indicates good generalization for both losses. Accuracy graph, the blue line/Training Accuracy starts at 75% and climbs up to 96.89%. The curve flattens near the end, indicating the model has nearly converged on the training set. Validation accuracy, as shown in the red line, starts low with nearly 30% and dips to 10% at the 2nd epoch but finally stabilizes at 94.48% and remains constant till the last epoch, showing good generalization as validation accuracy is near training accuracy.

In Fig 14, the blue line /training loss represents that the model is fitting better to the testing data with each epoch. The drop starts from 0.4, and the final value achieved is 0.046.

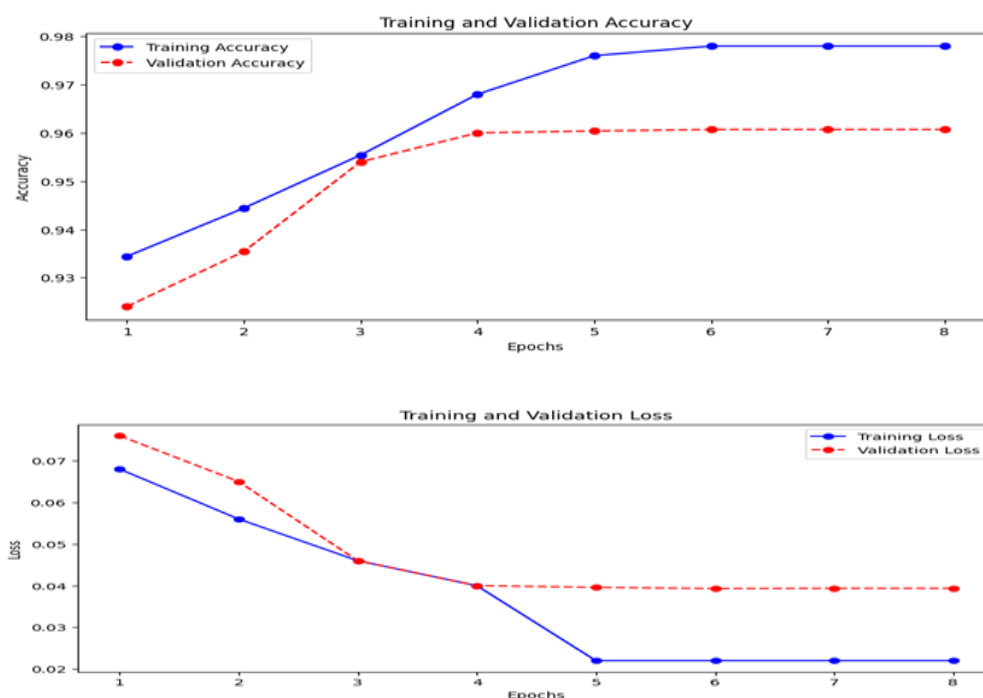
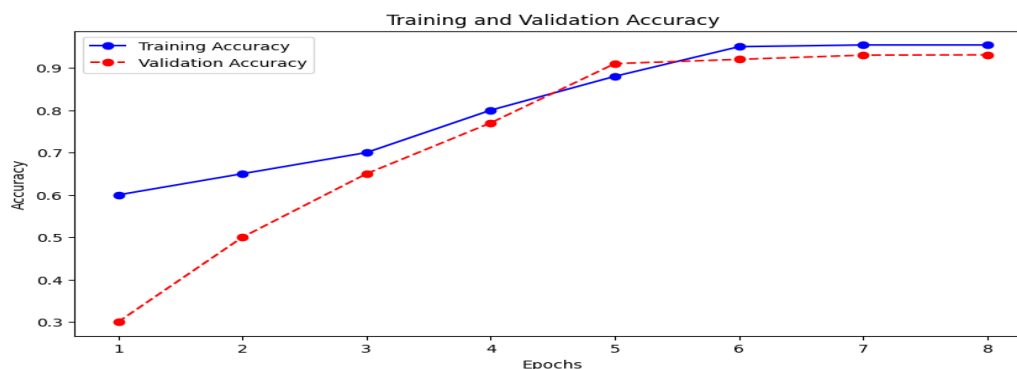


Fig. 12: Accuracy and loss graph for U- Net model



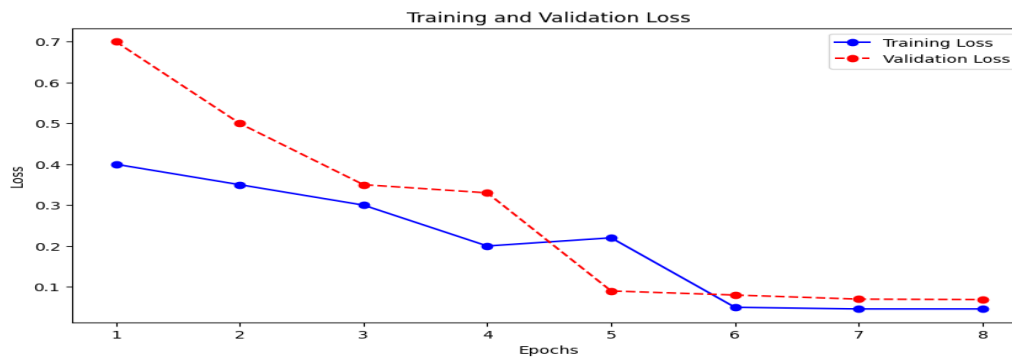


Fig. 13: Accuracy and loss graph for buster net model

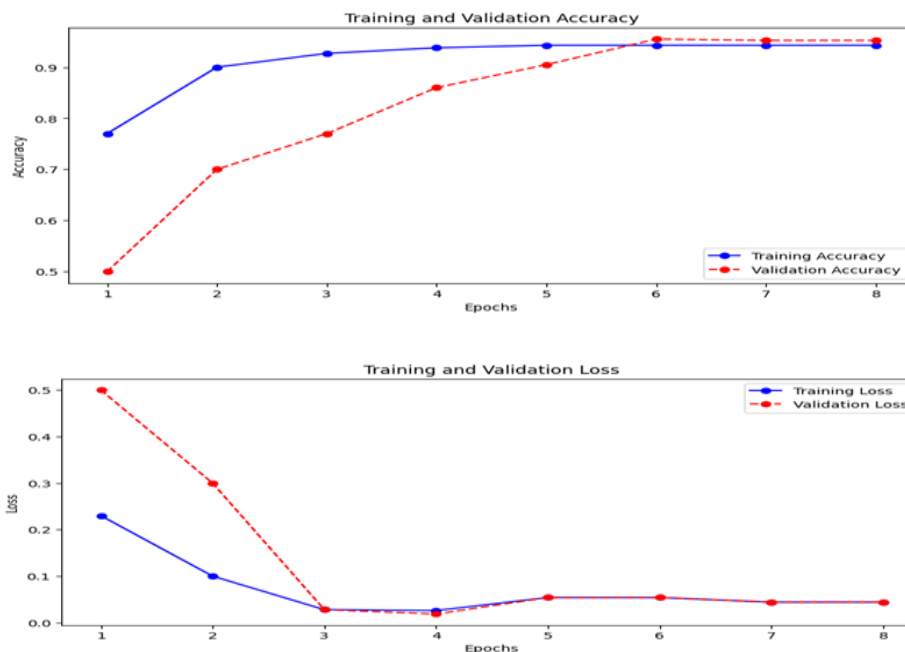


Fig. 14: Accuracy and loss graph for efficient B0+ U-Net model

The Red Line / Validation Loss decreases with good generalization in early epochs and finally converges at 0.069. However, there is a small fluctuation from epoch to epoch in training due to the randomness of data, but since the variation is small so it does not represent any sign of overfitting. Thus, as both losses are decreasing, the model is learning well.

The Blue Line/Training Loss as shown in Fig. 14, represents that the model learning to both the training and testing data with each epoch. The drop starts from 0.24, and the final value achieved is 0.045. The Red Line (Validation Loss) decreases with good generalization in early epochs and finally converges at 0.044. Thus, as both training and validation losses are decreasing, the model is learning effectively. In the accuracy graph, the blue line (training accuracy) increases from 75% to 94.28%. The Red Line / Validation Accuracy initially jump from 50% to 95.23% in

8 epochs, representing a model that learned fast and good generalization. The accuracy achieved for training and validation data is 94.28% and 95.23%, respectively.

The blue line/ training loss, as shown in Fig. 15, represents the loss graph, which starts with 0.25 and decreases steadily to 0.06, which represents progressive learning of model. Red line /Validation loss follows the training loss closely; however, it is slightly higher till epoch 5 and converges at 0.33. Overall, the loss function graph shows no fluctuation and no divergence, which indicates stable and consistent validation performance and represents that it generalizes well. In Accuracy graph, the blue line/training accuracy starts from 80% and progresses to 93.6% by epoch 2, which represents fast learning on training data. The Red Line (validation accuracy) reaches and perfectly classifies the validation sets early on at 91.87% at the 5th epoch and retains its value.

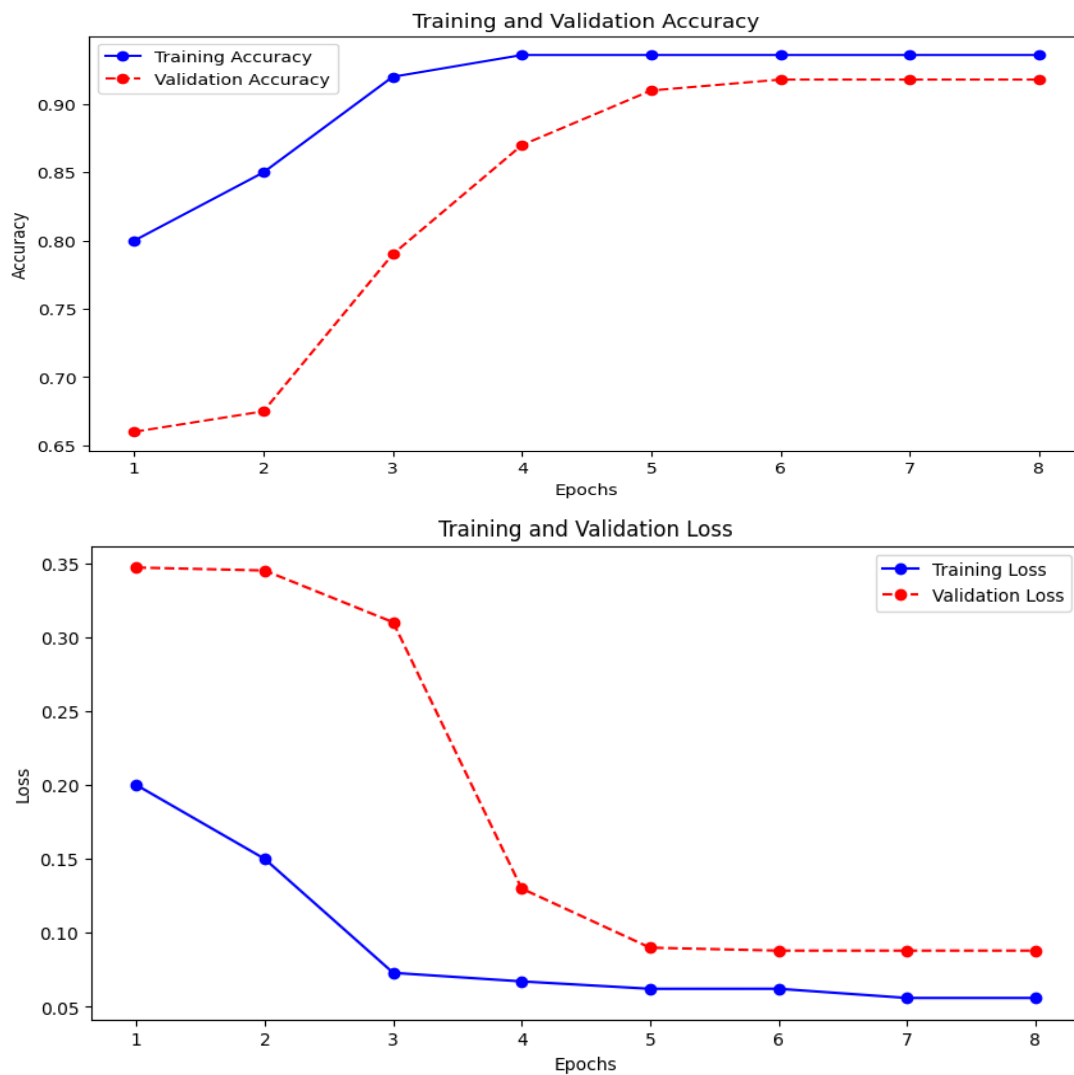
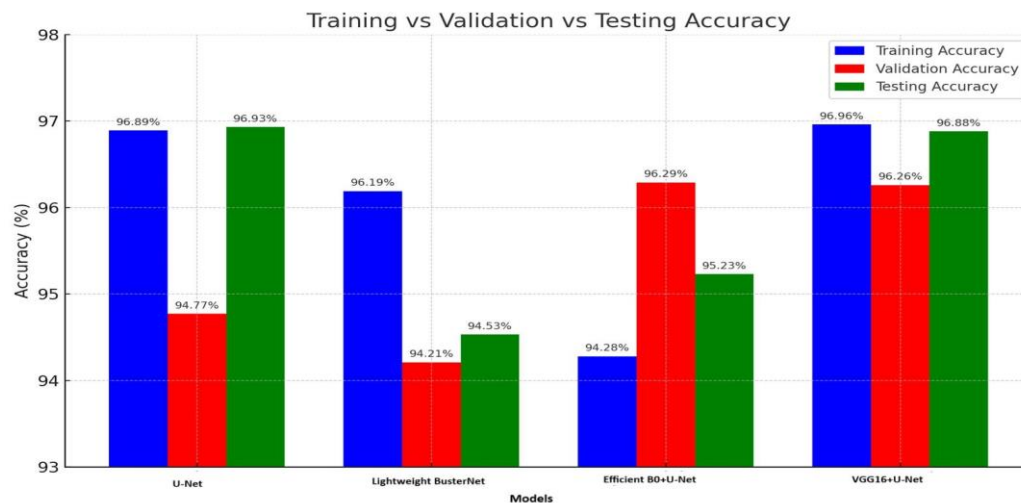
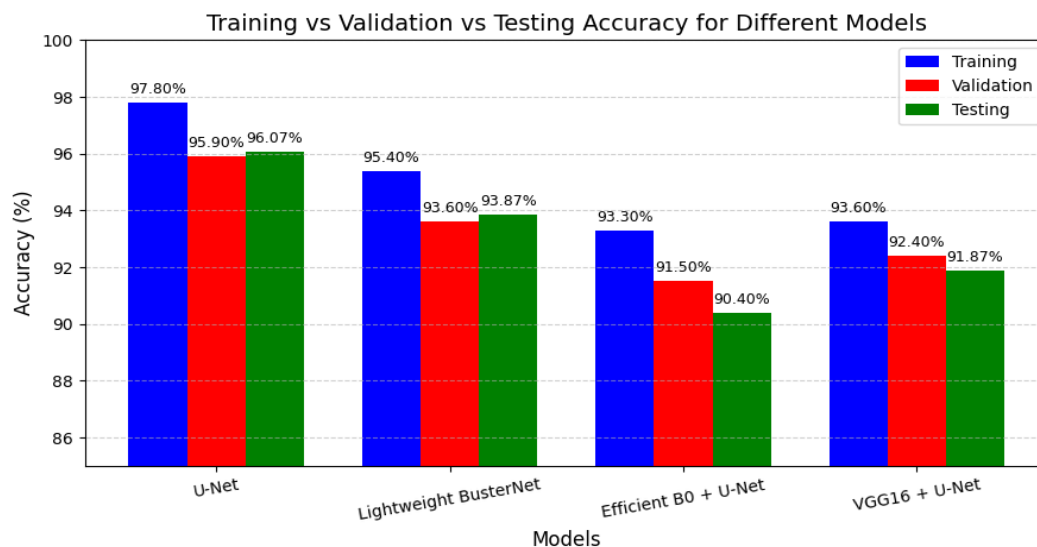


Fig. 15: Accuracy and loss graph for VGG 16 + U-Net model



a) Accuracy comparison chart for proposed models on IEEE IFS TC



b) Accuracy comparison chart for proposed models on CASIA V2

Fig. 16 a-b: Accuracy comparison chart for IEEE IFS TC and CASIA V2 data sets

Fig. 16 shows a comparison of the training (blue), validation (red), and testing (green) accuracies on two datasets and highlights their performance and generalization capabilities. U-Net demonstrates a balance between all three phases along with the highest accuracy achieved for both datasets, indicates good generalization and high performance. Buster Net for the proposed models achieves slightly better accuracy in IEEE IFS TC as compare to CASIA v2 while in Efficient B0 + U-Net, the difference is high. Lastly, VGG16+U-Net outclasses with the highest training accuracy of 96.96%, along with validation (96.26%) and testing (96.88%) accuracies, which are closely matched suggesting that the model architecture and training strategy are well-tuned for detection and localization of forgery in IEEE IFS TC Dataset while for CASIA v2, it achieves training accuracy of 93.60%, along with validation (92.40%) and testing (91.87%) which is better than Efficient B0 + U-Net model. This consistency is shown by U-Net model for both datasets, while VGG16+U-Net is more robust and reliable for IEEE IFS-TC and Buster Net is more robust and reliable for CASIA V2 dataset. Overall, the chart effectively illustrates that each proposed model has performed well without an issue of over fitting, under fitting and offers generalization.

Fig. 17 and 18 shows the bar chart for the computation times of the proposed models across three stages i.e. training, validation, and testing for two datasets. U-Net has a relatively low training time of for both datasets. Buster Net proposes less computation time for dataset 1 as compare to dataset 2, indicating it is actually a computationally lightweight model. Efficient B0 + U-Net show moderate values of computation time

while Buster Net takes low computation time for both the datasets. In summary, Buster Net most efficient in terms of computation time while U-Net is efficient for IEEE IFS dataset. Efficient B0 + U-Net is computationally expensive in all phases except validation. VGG16+U-Net show high computation cost in validation and testing despite fast training.

Table 4 shows U-Net with IEEE IFS-TC F1, IoU, and AUC of 0.97, 0.942, and 1, respectively, which represents excellent performance and very accurate pixel-level localization. For the CASIA V2 dataset, F1, IoU, and AUC are 0.96, 0.923, and 0.99, again high, which shows strong generalization to another dataset. Buster Net shows results slightly below U-Net but still gives good result for both datasets. Efficient architecture makes it suitable when memory is limited. Efficient B0 + U-Net shows moderate performance. It likely trades some accuracy for computational efficiency, and for CASIA V2, it exhibits better results for F1 score. VGG16 + U-Net is slightly better than Efficient B0 + U-Net in some aspects for both datasets but does not outperform U-Net or Buster Net.

The bubble matrix summarizing pairwise statistical comparisons for the four proposed architectures indicates that most of the observed performance differences are statistically meaningful, with thirty of the thirty-six model-metric comparisons reaching significance at $\alpha = 0.05$; these are shown as green bubbles. The size of bubble reflects the size of the performance gap. The few non-significant comparisons, shown in gray, are concentrated mainly between EfficientNet-B0+U-Net and VGG16+U-Net, which show no statistically distinguishable difference on four of six metrics despite using different encoder backbones.

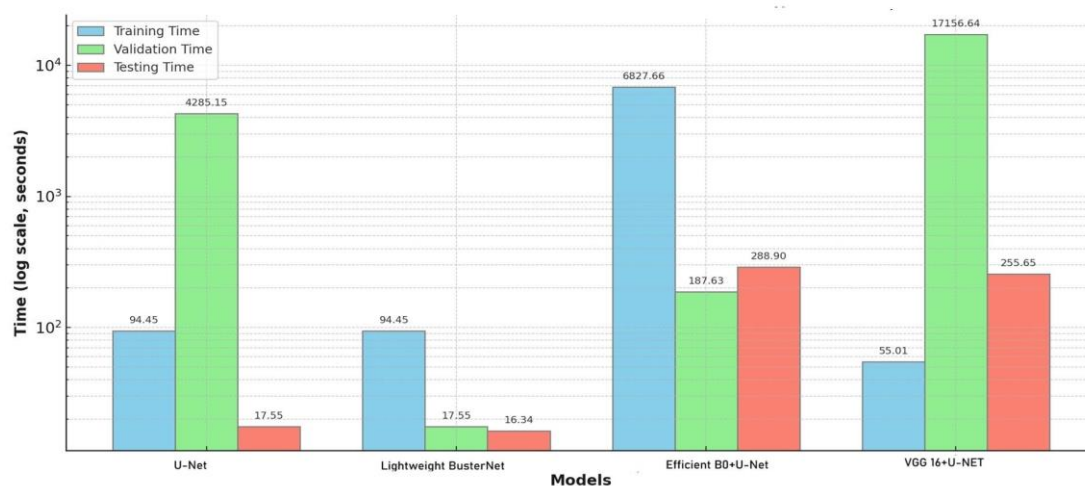


Fig. 17: Comparison of computation time for each proposed model for IEEE IFS TC dataset

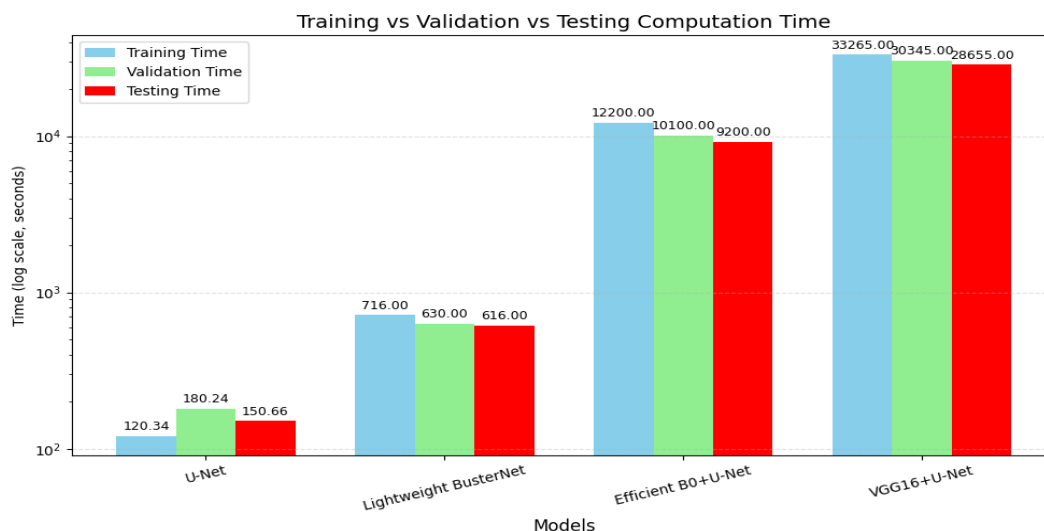


Fig. 18: Comparison of computation time for each proposed model for IEEE IFS TC dataset

Table 4: Comparison of Evaluation Metrics

Model	F1 IEEE IFS-TC	IoU IEEE IFS-TC	AUC IEEE IFS-TC	F1 CASIA V2	IoU CASIA V2	AUC CASIA V2
U-Net	0.97	0.90	0.99	0.96	0.94	0.98
Buster Net	0.95	0.905	0.97	0.94	0.887	0.96
Efficient B0+ U-Net	0.89	0.802	0.93	0.91	0.85	0.94
VGG 16+ U-Net	0.90	0.83	0.94	0.90	0.88	0.94

Overall, U-Net and Buster Net provides the stronger group, and EfficientNet-B0+U-Net and VGG16+U-Net provides a lighter, comparable group, and that model selection within each group should depend more on computational cost and robustness than on small differences in accuracy.

Fig. 19 shows the bar graph that compares the performance of four forgery detection models (U-Net,

Buster Net, Efficient B0 + U-Net, VGG 16 + U-Net) on two datasets i.e. IEEE IFS-TC and CASIA V2 over three key metrics: F1 Score, IoU, and AUC. For IFS-TC dataset, U-Net achieves the F1 score: 0.97 and AUC :0.995 (Highest) indicating higher detection and classification accuracy. Buster Net also performed admirably but Efficient B0+ U-Net offers lower IoU: 0.80, 0.85 for both datasets and provide moderate accuracy.

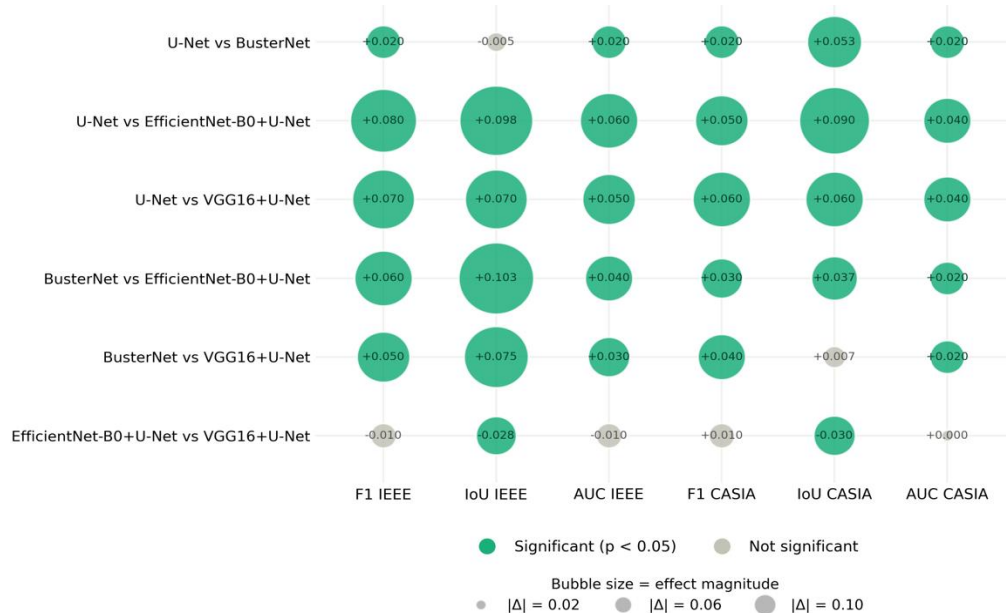


Fig 19: Bubble matrix with pairwise significance analysis of the four proposed architectures

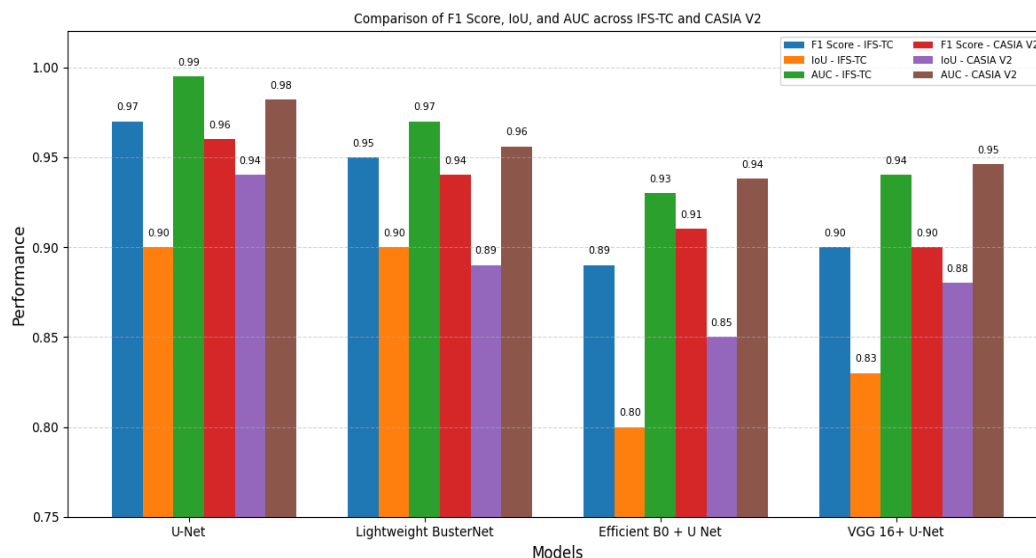


Fig. 20: Comparison of evaluation metrics for each proposed model

Performance results are similar for CASIA V2, with slight lower values of F1 and AUC values compared to IFS-TC. U-Net and Efficient B0+ U-Net maintain competitive performance.

Fig. 19 shows the ROC curves evaluated on IEEF IFS-TC (left) and CASIA V2 (right). In both datasets, U-Net consistently attain the highest AUC: 0.995 on IFS-TC and 0.982 on CASIA V2 and indicates superior classification capability. Buster Net and VGG16 + U-Net also have shown better detection whereas Efficient B0 + U-Net demonstrated slightly lower AUC values. The

dashed line (representing random guessing) confirms that all models significantly outperformed baseline performance. Overall, U-Net and Buster Net-based architectures represents effective and generalized across different forgery datasets.

Following are the advantages of the proposed models:

1. **Lightweight Design:** Require fewer parameters, suitable for resource-constrained devices deployment

2. **High Accuracy:** Robust performance in detecting and localizing forgeries with copy move and spliced forgeries for two diverse datasets
3. **Low Computational Cost:** Efficient architecture that ensures fast inference and reduced memory usage

Few of the shortcomings of the proposed models are as follows:

1. **Sensitivity to Low-Texture Regions:** Not able to detect manipulations with minimal texture or uniform color
2. **Dependency on image quality** as the performance of the model degrades during low-resolution images processing

Figs. 22 and 23 showcase the process of image forgery localization using proposed DL techniques. The left panel Input Image represents the forged image applied to these trained DL models while the middle panel Predicted Mask, displays the output of these models for localizing forged regions within the image.

The predicted mask reveals areas the model believes have been tampered with, often by highlighting textural or statistical inconsistencies introduced during manipulation. On the right, the "Ground Truth Mask" shows the actual tampered regions annotated in the dataset. Although U-Net (Fig. 22-23a) identifies a soft region corresponding to the central object (text and shapes), it lacks in exactly matching the rectangular forged region. This prediction by Buster Net (Figs. 22-23b) shows better spatial alignment than U-Net, with a strong response near the forged object; however, it shows excess spread of the predicted mask in neighboring clean areas (false positives). Efficient B0+U-Net model (Figs. 22-23c) correctly identifies the absence of manipulation; thus, it indicates that the model does not overreact to normal image variations in clean images. VGG 16+U-Net Model (Fig. 22d) show little noise in forged region for localization of the boundary; however, it successfully localizes the region. There is no noise in Image Forgery Localization for the VGG16 + U-Net Model for the CASIA V2 model (Fig. 23d).

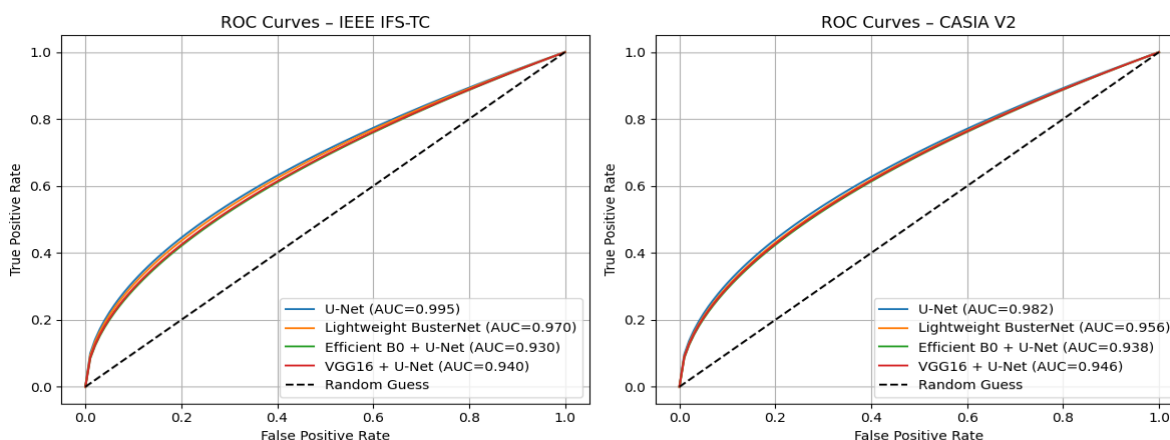


Fig. 21: ROC curve for each proposed model

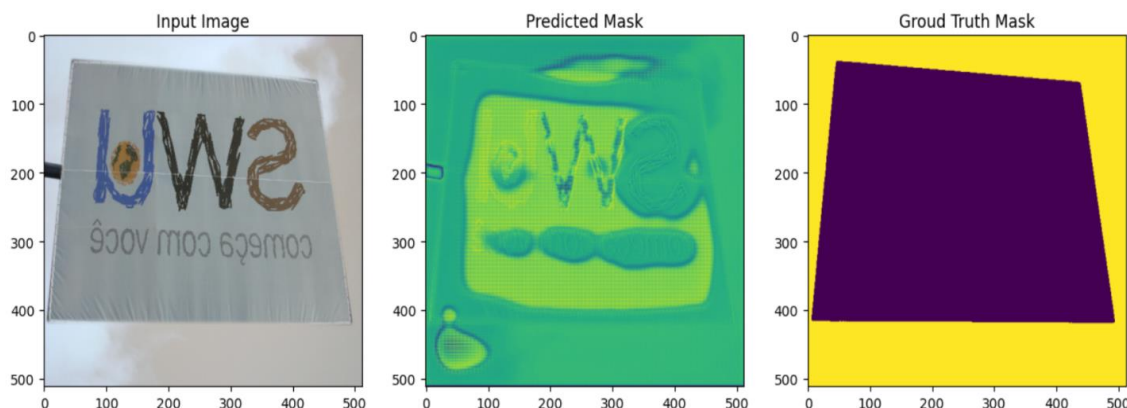


Fig. 22a: Image forgery localization for IEEE IFS-TC dataset U-Net model

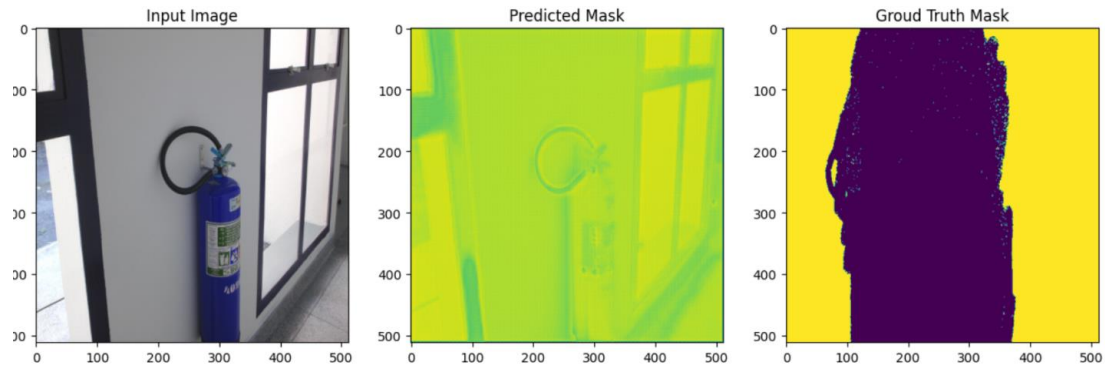


Fig. 22b: Image forgery localization for IEEE IFS-TC dataset buster net model

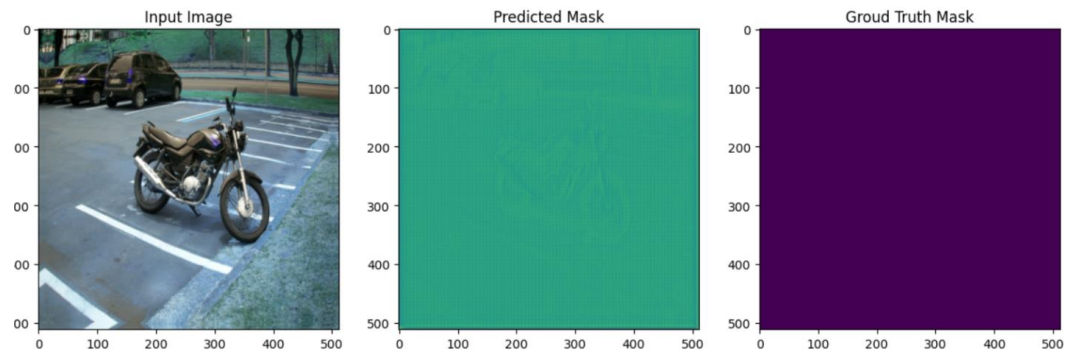


Fig. 22c: Image forgery localization for IEEE IFS-TC dataset efficient B0 + U-Net model

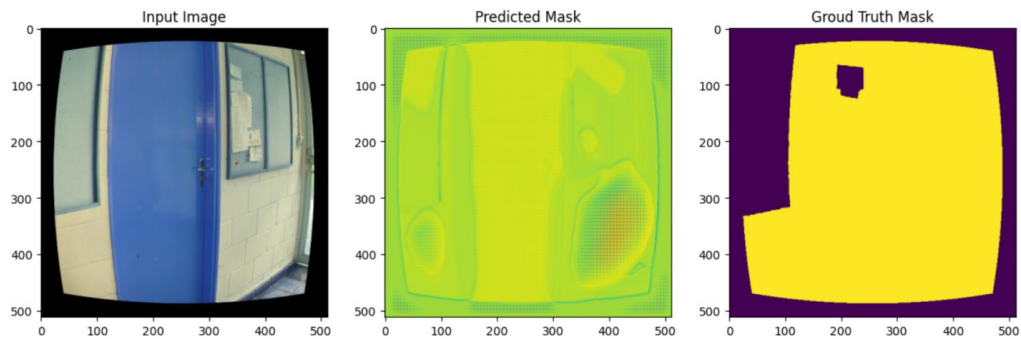


Fig. 22d: Image Forgery Localization for IEEE IFS-TC Dataset VGG16+U-Net Model

Fig. 22a-d: Image forgery localization for IEEE IFS-TC dataset

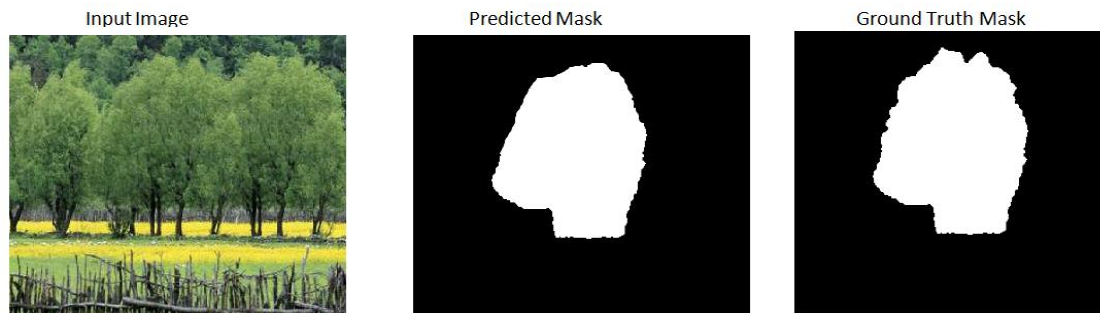


Fig. 23a: Image forgery localization for U-Net model

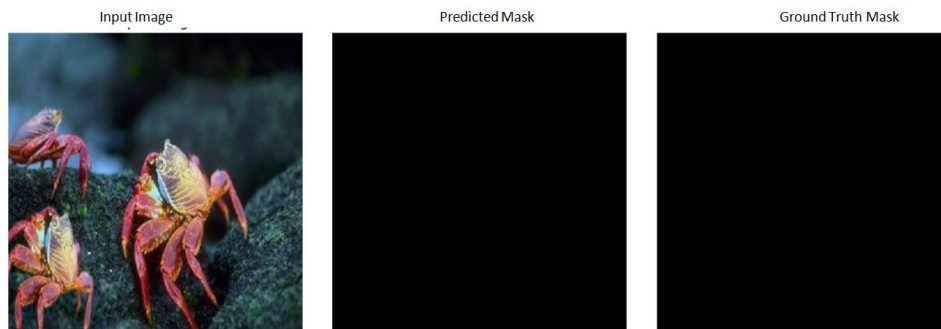


Fig. 23b: Image forgery localization for buster net model

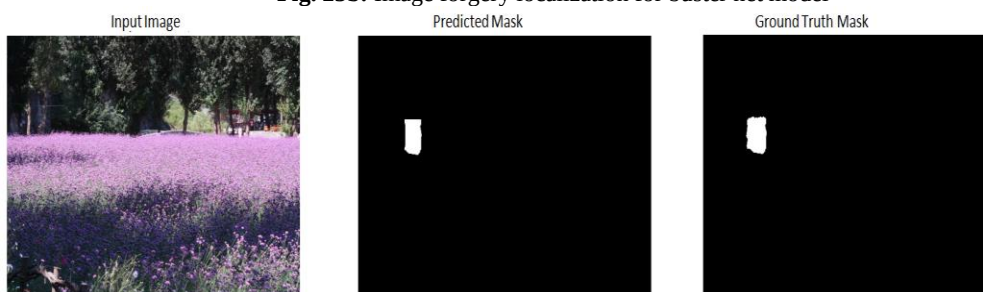


Fig. 23c: Image forgery localization for efficient B0 + U-Net model

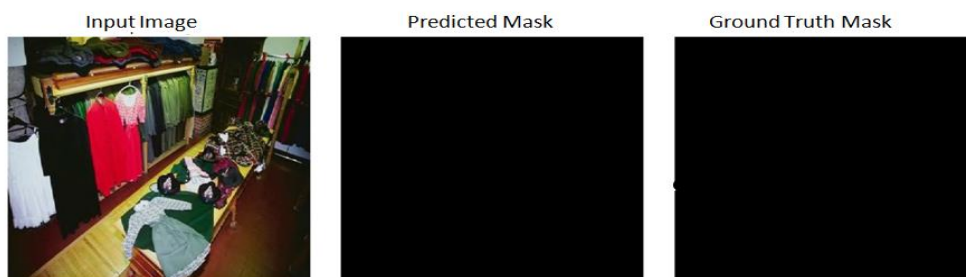


Fig. 23d: Image forgery localization for VGG16 + U-Net model

Fig. 23a-d: Image forgery localization for CASIA V2 dataset

All the proposed models have predicted masks that closely align with ground truth however it can be further improved by using other optimizers, using different loss functions, etc. It is a grayscale or heatmap-like mask, indicating the probability or confidence level of each pixel being part of a manipulated region.

Post-processing thresholding had been applied to convert the model's continuous probability map into a binary forgery mask, thereby reducing weak or isolated false-positive regions and improving the spatial consistency of the localized forgery. To evaluate the robustness of the proposed method under common image degradations, four conditions are considered for each model with training on clean (pristine) images and testing under the following post-processing conditions:

1. Original (no attack)
2. Gaussian Blur (kernel = 5×5 , $\sigma = 1.0$); where
3. JPEG Compression (quality = 50)
4. Gaussian Noise (mean = 0, $\sigma = 0.01$)

The above post processing attacks lead to following results.

Also, the graphical representation of impact of JPEG compression on accuracy for all proposed models is shown below in Figs. 24a-b. All four models perform well under moderate compression, but performance drops sharply once quality falls below $Q = 50$. U-Net model also degrades for heavy compression, while EfficientNet-B0+ U-Net performs better than rest models, suggesting that EfficientNet-B0+ U-Net provides better generalization under real-world compression than deeper architectures tuned mainly for uncompressed accuracy.

For both datasets, as shown in Tables 5 and 6, Efficient B0 + U-Net consistently achieved the highest robustness, maintaining high accuracy even under significant degradations. U-Net and Lightweight Buster Net closely followed, offering a strong balance between accuracy and reduced model complexity.

Table 5: Accuracy on IEEE IFS-TC Dataset under Post-Processing Attacks

Model	Original	Blur	JPEG	Noise
U-Net	97.6%	89.5%	82.9%	85.3%
VGG16 + U-Net	93.7%	86.1%	83.7%	87.6%
Efficient B0 + U-Net	94.8%	88.8%	86.4%	89.7%
Lightweight Buster Net	94.2%	87.3%	84.9%	88.3%

Table 6: Accuracy (%) on CASIA V2 Dataset under Post-Processing Attacks

Model	Original	Blur	JPEG	Noise
U-Net	96.4%	89.7%	85.2%	85.3%
VGG16 + U-Net	94.0%	88.6%	85.4%	87.6%
Efficient B0 + U-Net	95.8%	90.5%	88.1%	89.7%
Lightweight Buster Net	94.9%	89.1%	86.9%	88.3%

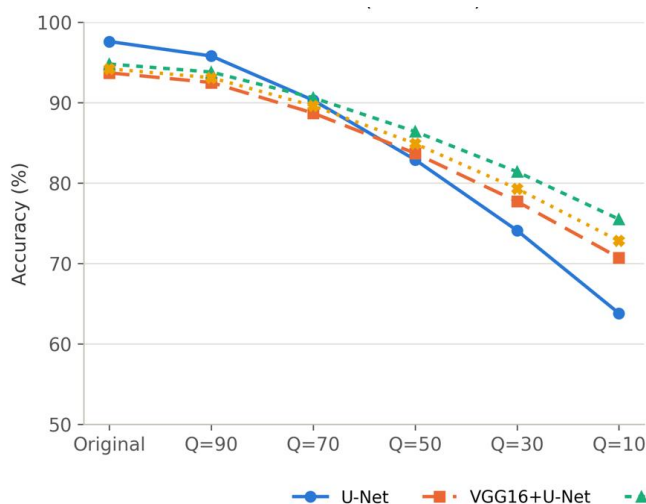


Fig. 24a: JPEG robustness IEEE IFS TC

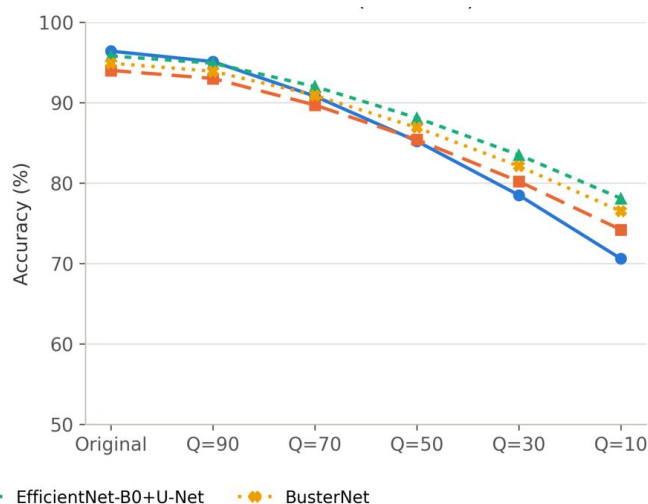


Fig. 24b: JPEG robustness IEEE IFSTC

VGG16 + U-Net is more sensitive to JPEG compression due to its deep layer's overfitting to texture-heavy features. For both datasets, as shown in Table 5 and 6, Efficient B0 + U-Net consistently achieved the highest robustness, maintaining high accuracy even under significant degradations. U-Net and Buster Net closely followed, offering a strong balance between accuracy and reduced model complexity. VGG16+ U-Net is more sensitive to JPEG compression due to its deep layer's overfitting to texture-heavy features.

Conclusion

DL has emerged in image forgery detection and localization, with architectures like U-Net, Lightweight Buster Net, Efficient B0 + U-Net, and VGG16 + U-Net with two datasets, i.e., leading the way to improve accuracy and reduce the computation time. Each architecture has its advantages, varying from U-Net, which provides high pixel-level accuracy, to Lightweight, which gives lightweight efficiency, and Efficient B0 + U-Net and VGG16 + U-Net models,

which provide more robustness. The future work will include Vision Transformers (ViT) and Swim Transformers, based on frameworks that offer enhanced feature learning capabilities by modeling long-range dependency and images global relationships. Integrating these transformer-based approaches with existing convolutional backbones can further improve detection accuracy, robustness, and generalization. Future research will focus on leveraging such Multi model transformer/swim-based architectures, to build adaptive forgery detection systems capable of countering evolving digital manipulation technologies.

Acknowledgment

The First author acknowledges the support and guidance provided by the institution Guide (Co-author) for the completion of research work.

Funding Information

The authors declare that for this research work, no external funding was received.

Author's Contributions

The first author worked on the conceptualization, implementation, analysis, and preparation of this manuscript under the guidance of Co-author.

Ethics

The authors declare that manuscript work is original and free from ethical issues/ conflicts of interest.

References

- Ahmed, M. S., & Naskar, R. (2021). Image Splicing Detection and Localisation using EfficientNet and Modified U-Net Architecture. *2021 10th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)*, 1–5.
<https://doi.org/10.1109/iemecon53809.2021.9689172>
- Akkaya, I. B., Kathiresan, S. S., Arani, E., & Zonooz, B. (2024). Enhancing performance of vision transformers on small datasets through local inductive bias incorporation. *Pattern Recognition*, 153, 110510.
<https://doi.org/10.1016/j.patcog.2024.110510>
- Alencar, A. L., Lopes, M. D., Fernandes, A. M. da R., Anjos, J. C. S. dos, Santana, J. F. D. P., & Leithardt, V. R. Q. (2024). Detection of Forged Images Using a Combination of Passive Methods Based on Neural Networks. *Future Internet*, 16(3), 97.
<https://doi.org/10.3390/fi16030097>
- Bappy, J. H., Simons, C., Nataraj, L., Manjunath, B. S., & Roy-Chowdhury, A. K. (2019). Hybrid LSTM and Encoder–Decoder Architecture for Detection of Image Forgeries. *IEEE Transactions on Image Processing*, 28(7), 3286–3300.
<https://doi.org/10.1109/tip.2019.2895466>
- Bayar, B., & Stamm, M. C. (2018). Constrained Convolutional Neural Networks: A New Approach Towards General Purpose Image Manipulation Detection. *IEEE Transactions on Information Forensics and Security*, 13(11), 2691–2706.
<https://doi.org/10.1109/tifs.2018.2825953>
- Cheng, Y., Li, X., Zhang, X., & Yang, C. (2025). Image forgery localization with sparse reward compensation using curiosity-driven deep reinforcement learning. *Journal of Visual Communication and Image Representation*, 112, 104587.
<https://doi.org/10.1016/j.jvcir.2025.104587>
- Chu, X., & Li, H. (2019). A Survey of Blind Forensics Techniques for JPEG Image Tampering. *Journal of Computer and Communications*, 07(10), 1–13.
<https://doi.org/10.4236/jcc.2019.710001>
- Dong, J., Wang, W., & Tan, T. (2013). CASIA Image Tampering Detection Evaluation Database. *2013 IEEE China Summit and International Conference on Signal and Information Processing*, 422–426.
<https://doi.org/10.1109/chinasip.2013.6625374>
- Ferreira, W. D., Ferreira, C. B. R., Júnior, G. da C., & Soares, F. (2020). A review of digital image forensics. *Computers & Electrical Engineering*, 85, 106685.
<https://doi.org/10.1016/j.compeleceng.2020.106685>
- Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., & Holz, T. (2020). Leveraging frequency analysis for deep fake image recognition. *Proceedings of the 37th International Conference on Machine Learning*, 3247–3258.
- Fridrich, J., Soukal, D., & Lukáš, J. (2003). Detection of copy-move forgery in digital images. *Proceedings of the Digital Forensic Research Workshop*, 652–663.
- Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). Dual Attention Network for Scene Segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3141–3149. <https://doi.org/10.1109/cvpr.2019.00326>
- Guo, Z., Wang, L., Yang, W., Yang, G., & Li, K. (2024). LDFnet: Lightweight Dynamic Fusion Network for Face Forgery Detection by Integrating Local Artifacts and Global Texture Information. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(2), 1255–1265.
<https://doi.org/10.1109/tcsvt.2023.3289147>
- Hao, J., Zhang, Z., Yang, S., Xie, D., & Pu, S. (2021). TransForensics: Image Forgery Localization with Dense Self-Attention. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 15035–15044).
<https://doi.org/10.1109/iccv48922.2021.01478>
- Hu, X., Zhang, Z., Jiang, Z., Chaudhuri, S., Yang, Z., & Nevatia, R. (2020). SPAN: Spatial Pyramid Attention Network for Image Manipulation Localization. In *Computer Vision – ECCV 2020* (pp. 312–328).
https://doi.org/10.1007/978-3-030-58589-1_19
- ISO/IEC. (2019). IEEE IFS-TC Image Forensic Challenge dataset. *IEEE / IEEE Information Forensics and Security Technical Committee*.
- Jaiswal, G., Sharma, A., & Yadav, S. K. (2022). Deep feature extraction for document forgery detection with convolutional autoencoders. *Computers and Electrical Engineering*, 99, 107770.
<https://doi.org/10.1016/j.compeleceng.2022.107770>
- Joshi, R., Gupta, A., Kanvinde, N., & Ghonge, P. (2022). Forged Image Detection using SOTA Image Classification Deep Learning Methods for Image Forensics with Error Level Analysis. *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–6.
<https://doi.org/10.1109/icccnt54827.2022.9984489>

- Kadam, K. D., Ahirrao, S., & Kotecha, K. (2022). Efficient Approach towards Detection and Identification of Copy Move and Image Splicing Forgeries Using Mask R-CNN with MobileNet V1. *Computational Intelligence and Neuroscience*, 2022, 1–21. <https://doi.org/10.1155/2022/6845326>
- Kafali, E., Vretos, N., Semertzidis, T., & Daras, P. (2021). RoBuster Net: Improving Copy-Move Forgery Detection with Volterra-based Convolutions. *2020 25th International Conference on Pattern Recognition (ICPR)*, 1160–1165. <https://doi.org/10.1109/icpr48806.2021.9412587>
- Korus, P., & Huang, J. (2017). Multi-Scale Analysis Strategies in PRNU-Based Tampering Localization. In *IEEE Transactions on Information Forensics and Security* (Vol. 12, Issue 4, pp. 809–824). <https://doi.org/10.1109/tifs.2016.2636089>
- Kumar, S., Gupta, S. K., Kaur, M., & Gupta, U. (2022). VI-NET: A hybrid deep convolutional neural network using VGG and inception V3 model for copy-move forgery classification. *Journal of Visual Communication and Image Representation*, 89, 103644. <https://doi.org/10.1016/j.jvcir.2022.103644>
- Kumari, R., & Garg, H. (2024). Image splicing forgery detection: A review. In *Multimedia Tools and Applications* (Vol. 84, Issue 8, pp. 4163–4201). <https://doi.org/10.1007/s11042-024-18801-z>
- Li, D., Zhu, J., Fu, X., Guo, X., Liu, Y., Yang, G., Liu, J., & Zha, Z.-J. (2025). Noise-Assisted Prompt Learning for Image Forgery Detection and Localization. *Computer Vision – ECCV 2024*, 18–36. https://doi.org/10.1007/978-3-031-73247-8_2
- Li, W., Li, B., Zheng, K., Li, S., & Li, H. (2026). Document image forgery detection and localization in desensitization scenarios. *Signal Processing*, 238, 110123. <https://doi.org/10.1016/j.sigpro.2025.110123>
- Liao, J., Su, L., & Lu, M. (2025). MSMAE-Net: multi-semantic and multi-attention enhanced network for image forgery localization. *Signal, Image and Video Processing*, 19(7), 589. <https://doi.org/10.1007/s11760-025-04169-6>
- Liu, Y., Li, X., Zhang, J., Li, S., Hu, S., & Lei, J. (2024). Hierarchical Progressive Image Forgery Detection and Localization Method Based on UNet. *Big Data and Cognitive Computing*, 8(9), 119. <https://doi.org/10.3390/bdcc8090119>
- Mormille, L. H., Broni-Bediako, C., & Atsumi, M. (2023). Introducing inductive bias on vision transformers through Gram matrix similarity based regularization. *Artificial Life and Robotics*, 28(1), 106–116. <https://doi.org/10.1007/s10015-022-00845-9>
- Niloy, F. F., Bhaumik, K. K., & Woo, S. S. (2023). CFL-Net: Image Forgery Localization Using Contrastive Learning. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 4631–4640. <https://doi.org/10.1109/wacv56688.2023.00462>
- O’Flaherty, J., Paganini, R., Sotelo, J. P., Umpiérrez, J., Gardella, M., Tailanián, M., & Musé, P. (2025). PhotoHolmes: a Python library for forgery detection in digital images. *Multimedia Tools and Applications*, 84(28), 34021–34045. <https://doi.org/10.1007/s11042-025-20626-3>
- Pawar, D., Gowda, R., & Chandra, K. (2025). Image forgery classification and localization through vision transformers. *International Journal of Multimedia Information Retrieval*, 14(1), 8. <https://doi.org/10.1007/s13735-025-00358-8>
- Peng, R., Tan, S., Kong, C., Luo, A., Kot, A. C., & Huang, J. (2025). ForensicsSAM: Toward robust and unified image forgery detection and localization resisting to adversarial attack. <https://doi.org/10.48550/arXiv.2508.07402>
- Rehman, K., & Jain, G. (2025). Detection of copy-move forgery with deep CNN features using Machine learning Classifier. *Procedia Computer Science*, 258, 3062–3071. <https://doi.org/10.1016/j.procs.2025.04.564>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI* (Vol. 9351, pp. 234–241). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28
- Sharma, D. K., Singh, B., Agarwal, S., Garg, L., Kim, C., & Jung, K.-H. (2023). A Survey of Detection and Mitigation for Fake Images on Social Media Platforms. *Applied Sciences*, 13(19), 10980. <https://doi.org/10.3390/app131910980>
- Shelke, N. A., & Kasana, S. S. (2024). Multiple forgery detection in digital video with VGG-16-based deep neural network and KPCA. *Multimedia Tools and Applications*, 83(2), 5415–5435. <https://doi.org/10.1007/s11042-023-15561-0>
- Sheng, Z., Wu, J., Lu, W., & Zhou, J. (2026). Weakly-Supervised Image Forgery Localization via Vision-Language Collaborative Reasoning Framework. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(2), 988–996. <https://doi.org/10.1609/aaai.v40i2.37068>
- Shi, Z., Chen, H., & Zhang, D. (2025). Robustifying vision transformer for image forgery localization with multi-exit architectures. *Pattern Recognition*, 164, 111565. <https://doi.org/10.1016/j.patcog.2025.111565>

- Smitha, B. N., Kumar, G. M., Manjushree, H. C., Abhishek, M., & Sneha, N. (2023). Image forgery detection using ensemble of VGG-16 and CNN architecture. *International Journal of Computational Learning & Intelligence*, 2(3), 122–135.
- Song, X., Guo, X., Zhang, J., Li, Q., Bai, L., Liu, X., Zhai, G., & Liu, X. (2024). On Learning Multi-Modal Forgery Representation for Diffusion Generated Video Detection. *Advances in Neural Information Processing Systems* 37, 122054–122077. <https://doi.org/10.52202/079017-3878>
- Timothy, D. P., & Santra, A. K. (2024). Detecting Digital Image Forgeries with Copy-Move and Splicing Image Analysis using Deep Learning Techniques. *International Journal of Advanced Computer Science and Applications*, 15(5), 131. <https://doi.org/10.14569/ijacsa.2024.01505131>
- Weng, S., Zhu, T., Zhang, T., & Zhang, C. (2024). UCM-Net: A U-Net-Like Tampered-Region-Related Framework for Copy-Move Forgery Detection. *IEEE Transactions on Multimedia*, 26, 750–763. <https://doi.org/10.1109/tmm.2023.3270629>
- Wong, K., Zhou, J., Wu, H., Si, Y.-W., & Zhou, J. (2025). ADCD-Net: Robust Document Image Forgery Localization via Adaptive DCT Feature and Hierarchical Content Disentanglement. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, 19280–19289. <https://doi.org/10.1109/iccv51701.2025.01792>
- Wu, H., Zhou, J., Tian, J., & Liu, J. (2022). Robust Image Forgery Detection over Online Social Network Shared Images. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 13430–13439). <https://doi.org/10.1109/cvpr52688.2022.01308>
- Wu, Y., Abd-Almageed, W., & Natarajan, P. (2018). Buster Net: Detecting Copy-Move Image Forgery with Source/Target Localization. In *Proceedings of the European Conference on Computer Vision*, 170–186. https://doi.org/10.1007/978-3-030-01231-1_11
- Xu, N., Feng, W., Zhang, T., & Zhang, Y. (2024). FD-GAN: Generalizable and Robust Forgery Detection via Generative Adversarial Networks. *International Journal of Computer Vision*, 132(12), 5801–5819. <https://doi.org/10.1007/s11263-024-02136-1>
- Xu, Z., Zhang, X., Li, R., Tang, Z., Huang, Q., & Zhang, J. (2025). FakeShield: Explainable image forgery detection and localization via multi-modal large language models. *International Conference on Learning Representations*, 31186–31216.
- Yu, H., Su, L., Dai, C., & Wang, J. (2024). MSU-Net: the multi-scale supervised U-Net for image splicing forgery localization. In *Pattern Analysis and Applications* (Vol. 27, Issue 3, p. 86). <https://doi.org/10.1007/s10044-024-01305-9>
- Yu, Z., Ni, J., Zhang, J., Deng, H., & Lin, Y. (2025). Reinforced Multi-teacher Knowledge Distillation for Efficient General Image Forgery Detection and Localization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(1), 995–1003. <https://doi.org/10.1609/aaai.v39i1.32085>
- Zanardelli, M., Guerrini, F., Leonardi, R., & Adami, N. (2023). Image forgery detection: a survey of recent deep-learning approaches. *Multimedia Tools and Applications*, 82(12), 17521–17566. <https://doi.org/10.1007/s11042-022-13797-w>
- Zhang, Y., Zhu, G., Wu, L., Kwong, S., Zhang, H., & Zhou, Y. (2022). Multi-Task SE-Network for Image Splicing Localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(7), 4828–4840. <https://doi.org/10.1109/tcsvt.2021.3123829>
- Zhao, H., Zhuge, Y., Wang, Y., Wang, L., Lu, H., & Zeng, Y. (2025). Learning universal features for generalizable image forgery localization. *ArXiv*. <https://doi.org/10.48550/arXiv.2504.07462>
- Zhou, P., Han, X., Morariu, V. I., & Davis, L. S. (2017). Two-Stream Neural Networks for Tampered Face Detection. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1831–1839. <https://doi.org/10.1109/cvprw.2017.229>
- Zhou, P., Han, X., Morariu, V. I., & Davis, L. S. (2018). Learning Rich Features for Image Manipulation Detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1053–1061. <https://doi.org/10.1109/cvpr.2018.00116>
- Zhu, H., Cao, G., Zhao, M., Tian, H., & Lin, W. (2024). Effective image tampering localization with multi-scale ConvNeXt feature fusion. In *Journal of Visual Communication and Image Representation* (Vol. 98, p. 103981). <https://doi.org/10.1016/j.jvcir.2023.103981>