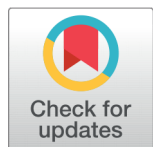


ORIGINAL ARTICLE



OPEN ACCESS

Received: 05/04/2026

Accepted: 26/06/2026

Published: 18/07/2026

Citation: Rajashekara M, Chethan HK, Thejashwini BL, Somanna M (2026) Integrating Handcrafted Features and Deep Learning Models for Image Splicing Detection: A Hybrid Feature Fusion Approach. Indian Journal of Science and Technology 19(25): 1783-1796. <https://doi.org/10.17485/IJST/v19i25.543>

* **Corresponding author.**m.rajashekara22@gmail.com**Funding:** None**Competing Interests:** None

Copyright: © 2026 Rajashekara et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment ([iSee](#))

ISSN

Print: 0974-6846

Electronic: 0974-5645

Integrating Handcrafted Features and Deep Learning Models for Image Splicing Detection: A Hybrid Feature Fusion Approach

M Rajashekara^{1*}, H K Chethan², B L Thejashwini¹, M Somanna¹

¹ Department of Studies in Computer Science, University of Mysore, Mysuru -570006, Karnataka, India

² Department of Computer Science & Engineering, MIT, Mysuru, India

Abstract

Objectives: This study develops a robust framework for image splicing detection by integrating handcrafted descriptors, pre-trained convolutional neural networks, and hybrid multi-CNN feature fusion strategies to improve classification accuracy across datasets of varying complexity. **Method:** Three detection pipelines were evaluated on the CUISDE dataset (363 controlled single-camera images) and the MISD dataset (923 high-resolution images up to 9571×15022 pixels). The first pipeline used handcrafted features including LBP, HOG, and GLCM, reduced via PCA, LDA, and Kernel PCA, then classified using nine machine learning models. The second pipeline fine-tuned 31 pre-trained CNN architectures end to end. The third pipeline concatenated features from DenseNet169, Xception, VGG19, and MobileNetV2, applied PCA, and used machine learning classifiers for prediction. All experiments used a strict 80/20 stratified split with a train-before-transform protocol and 5-fold cross-validation on training data only. **Findings:** On CUISDE, six classifiers including SVM, Random Forest, Gradient Boosting, Extra Trees, Bagging, and Logistic Regression achieved perfect scores of 1.00 for accuracy, precision, recall, and F1-score under PCA. On MISD, PCA-based handcrafted features yielded only 67 to 68 percent accuracy, while LDA with AdaBoost recovered performance to 96 percent. ConvNeXtLarge and ConvNeXtXLarge achieved the highest test accuracy of 95 percent with precision and F1-score of approximately 0.95 on MISD. The hybrid fusion pipeline reached 89 percent accuracy with precision of 0.91 and recall of 0.87 using Logistic Regression. CLAHE pre-processing showed a statistically significant benefit only for VGG16 on the Columbia dataset with $\chi^2(1) = 4.08$ and $p = 0.043$. **Novelty:** This work presents a systematic benchmarking pipeline evaluating handcrafted, deep, and hybrid fused features across nine classifiers, three dimensionality reduction strategies, and two contrasting datasets, offering clear empirical guidance for selecting the most suitable detection approach based on dataset complexity and deployment needs.

Keywords: Image Splicing Detection; Machine Learning; Deep Learning; Feature Fusion; Digital Forensics

1 Introduction

Digital media and image editing tools have advanced rapidly, increasing the risk of image manipulation and raising serious doubts about the authenticity and reliability of images. Image splicing, where parts of multiple images are combined to create a single composite image, has become one of the most common and misleading methods of digital forgery⁽¹⁾. Due to the easy access to high-end editing software such as Adobe Photoshop and GIMP, individuals without specialist expertise can produce highly realistic forged photographs. Although these techniques have legitimate uses in creative fields, their misuse in journalism, law enforcement, national security, and legal documentation may result in misinformation, fabricated evidence, and damage to reputations.

The classic approaches to identifying image splicing rely primarily on visual inspection by human experts. Nonetheless, this is a time-consuming, subjective, and insufficiently effective method for combating advanced manipulations⁽²⁾. This limitation has driven the development of automated detection techniques. Current detection methods are broadly divided into active and passive categories. Active techniques embed auxiliary data such as watermarks or digital signatures during image capture, which are later verified. However, they are applicable only when such prior information exists. Passive methods, by contrast, examine intrinsic image properties (including lighting inconsistencies, noise patterns, and texture) and are more applicable in real-world scenarios involving unknown or externally obtained image data⁽³⁾.

Modern splicing methods can produce forgeries that are visually indistinguishable from authentic images. Common post-processing operations such as compression, scaling, rotation, and noise addition further degrade detection reliability. Moreover, the absence of large, heterogeneous, and balanced datasets impedes the construction of robust and generalizable models⁽⁴⁾.

Recent advances in machine learning (ML) and deep learning (DL) have substantially improved passive image splicing detection. Conventional ML methods rely on manually computed features such as LBP, HOG, and GLCM to represent texture and structural dissimilarities⁽⁵⁾. Although effective in controlled environments, these methods depend heavily on feature engineering and frequently fail to generalize to diverse datasets⁽⁶⁾.

Deep learning methods, particularly convolutional neural networks (CNNs), automatically extract hierarchical representations from image data. These models have demonstrated strong capability in detecting subtle manipulation artifacts⁽⁷⁾. Nonetheless, they are computationally expensive, sensitive to distributional shifts, and generally lack interpretability properties that limit their suitability for forensic and legal applications where explainability is essential.

1.1 Research Gap and Motivation

Despite significant advancements in image splicing detection, several critical limitations remain unresolved. Most studies focus either on handcrafted feature-based approaches or deep learning models in isolation, with limited effort toward systematically integrating both paradigms to exploit their complementary strengths⁽⁸⁾. Additionally, there is a lack of comprehensive evaluation of dimensionality reduction techniques such as PCA, LDA, and Kernel PCA across both handcrafted and deep feature spaces. Hybrid transfer learning strategies that combine features from multiple pre-trained models remain underexplored in terms of their effectiveness in improving detection robustness⁽⁹⁾. The application of ensemble learning methods (including boosting and bagging strategies) has not been sufficiently investigated across varying manipulation scenarios. Note: stacking is identified as a gap in prior work; the present study evaluates boosting and bagging but does not implement stacking, which is reserved for future work. From a practical standpoint, existing methods exhibit poor generalization across diverse datasets and high sensitivity to common post-processing operations such as compression, resizing, rotation, and noise addition⁽¹⁰⁾. Dataset limitations including imbalance and lack of diversity further contribute to biased learning. Moreover, deep learning models, despite their high accuracy, suffer from high computational complexity and limited interpretability, restricting real-world forensic applicability⁽¹¹⁾.

1.2 Novelty and Differentiation from Prior Work

The present work builds upon and explicitly differentiates itself from two closely related studies. Jaiswal et al. (2020)⁽¹²⁾ employs a fixed hybrid feature set (LBP, HOG, and phase congruency) combined with a single machine learning classifier and one dimensionality reduction strategy (PCA), evaluated on the Columbia dataset. Almawas et al. (2020)⁽¹³⁾ evaluates multiple classifiers on handcrafted features but does not incorporate deep features or compare alternative dimensionality reduction techniques.

The present study advances beyond both works in three respects: (1) it conducts simultaneous comparison of PCA, LDA, and Kernel PCA across both handcrafted and deep feature spaces; (2) it benchmarks multi-CNN feature fusion against nine distinct classifier families; and (3) it provides dual-dataset evaluation covering a controlled benchmark (CUISDE) and a complex,

high-resolution dataset (MISD). The feature fusion strategy employed is concatenation-based rather than a novel architectural innovation; the contribution is therefore best characterized as systematic empirical benchmarking of an integrated detection pipeline.

1.3 Key Contributions of the Study

To address the above gaps, this study proposes a comprehensive and integrated framework for image splicing detection with the following key contributions:

- Combines handcrafted features (LBP, HOG, GLCM) with deep features extracted from multiple pre-trained CNN architectures to leverage complementary information.
- Provides a detailed comparative evaluation of PCA, LDA, and Kernel PCA across both handcrafted and deep feature representations.
- Introduces a feature fusion approach by concatenating representations from multiple CNN models to enhance discriminative capability.
- Evaluates a wide range of machine learning classifiers, including boosting (Gradient Boosting, AdaBoost) and bagging strategies alongside standard classifiers — stacking is identified as a direction for future work.
- Validates the proposed framework on both CUISDE (controlled) and MISD (complex) datasets to assess robustness and generalization.
- Demonstrates that while handcrafted features excel in controlled settings, deep learning models provide superior scalability, and hybrid methods offer stable and balanced performance.

2 Methodology

Two datasets were used for evaluation: CUISDE and MISD. CUISDE is a small, controlled benchmark comprising 363 images (183 authentic and 180 spliced), mostly indoor scenes captured by a single camera. Spliced images were created by copying regions from genuine photographs while preserving EXIF metadata, making the manipulations subtle and suitable for controlled binary classification⁽¹⁴⁾.

MISD is a larger and more complex collection of 923 high-resolution images (up to 9571×15022 pixels) in .jpg and .png formats, accompanied by ground-truth masks. With diverse textures and challenging scenes, MISD tests the robustness of detection methods against realistic conditions. Together, the two datasets provide a balanced evaluation framework spanning controlled and complex scenarios^{(15), (16)}.

The methodology follows a clear sequential pipeline: pre-processing, feature extraction, dimensionality reduction, and classification. Images were first standardized by converting to grayscale, resizing to a fixed dimension, and normalizing pixel values. Contrast adjustment was applied in selected conditions to enhance texture details. Two feature extraction strategies were pursued with handcrafted descriptors and pre-trained CNN-based features, followed by dimensionality reduction and evaluation across multiple classifier families.

2.1 Machine Learning Approaches for Classical Feature Extraction Methods

This section presents the machine learning-based approach using classical handcrafted feature extraction techniques for image splicing detection. The focus is on extracting discriminative features using traditional descriptors and evaluating their effectiveness through normalization, dimensionality reduction, and classification⁽¹⁷⁾. This approach establishes a strong baseline for detecting inconsistencies in manipulated images. The overall workflow of the proposed handcrafted feature-based detection framework is illustrated in Figure 1.

Three complementary descriptors were selected: the Gray Level Co-occurrence Matrix (GLCM) to capture texture statistics; Histogram of Oriented Gradients (HOG) for edge and shape information; and Local Binary Patterns (LBP) for fine-grained local variation. Together, they provide a strong foundation for detecting inconsistencies introduced during image manipulation⁽¹⁸⁾.

To prevent any single descriptor from dominating due to scale differences, four normalization techniques were tested: Standard Scaler, MinMax Scaler, MaxAbs Scaler, and Robust Scaler. Once normalized, feature vectors were reduced using PCA, Kernel PCA, and LDA⁽¹⁹⁾. This step reduces computational cost and emphasizes patterns most relevant to classification.

The resulting features were evaluated using nine classifier families: Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors (KNN), Decision Tree, Logistic Regression, Gradient Boosting, AdaBoost, Extra Trees, and Bagging. This breadth of coverage (spanning distance-based, probabilistic, and ensemble learning strategies) provides insight into the relative advantages and trade-offs of different models.

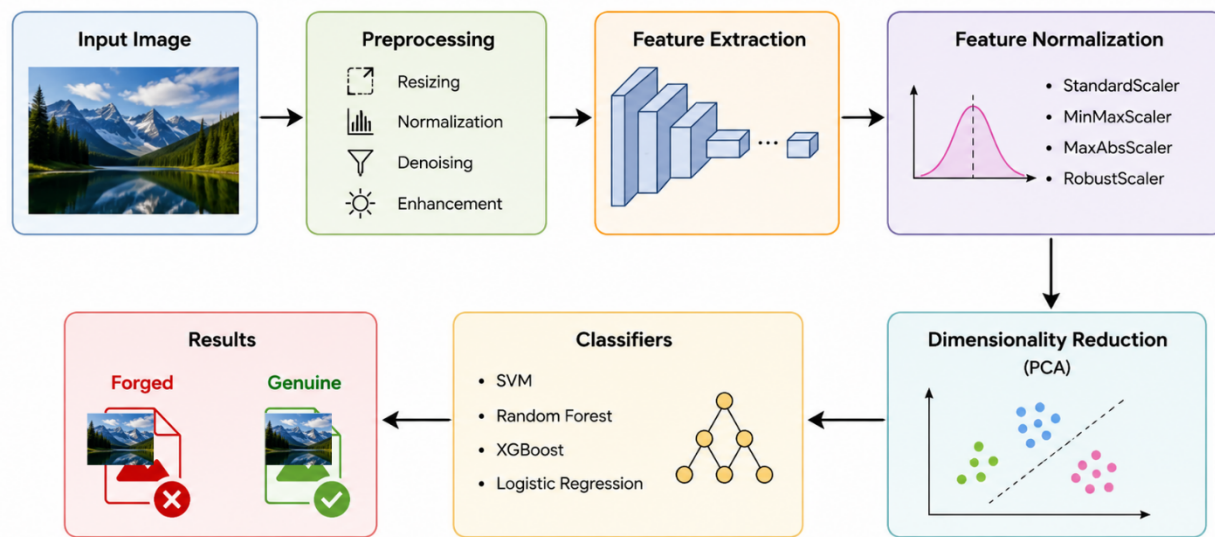


Fig 1. Handcrafted feature extraction pipeline. The sequential stages from input image to final classification are shown, including pre-processing, feature extraction, normalization, dimensionality reduction, and classification.

2.2 A Transfer Learning Model with a Dense Classifier

This section presents a transfer learning-based approach using pre-trained convolutional neural networks. Rather than training models from scratch, established architectures are used as feature extractors to capture high-level representations efficiently. The overall architecture is illustrated in Figure 2.

Architectures including VGG, ResNet, DenseNet, MobileNet, Xception, EfficientNet, and ConvNeXt were selected. Each brings distinct design strengths: MobileNet is lightweight and resource-efficient; ResNet and DenseNet employ skip and dense connections to improve gradient flow; EfficientNet optimizes scaling across multiple dimensions; and ConvNeXt incorporates Transformer-inspired design principles. From the comparative evaluation, MobileNet and ConvNeXt produced the highest accuracy, demonstrating that even compact or modernized CNNs can reliably recognize the subtle signatures of image splicing^{(20), (21)}.

2.3 Hybrid CNN Feature Fusion and Machine Learning Classification

This section presents a hybrid approach that combines feature representations from multiple pre-trained CNN models. By fusing features extracted from architecturally diverse networks, the method aims to capture complementary information for improved detection performance⁽¹³⁾. The resulting hybrid feature space is further optimized using dimensionality reduction and evaluated using machine learning classifiers. The overall pipeline is illustrated in Figure 3.

DenseNet169, Xception, VGG19, and MobileNetV2 were used purely as feature extractors. Each image was resized and normalized before being passed through the respective networks. The resulting feature maps were concatenated into a hybrid vector providing a richer image description than any single network could supply.

Because the combined feature set is large, dimensionality reduction was applied. PCA consistently produced better accuracy while maintaining efficiency. The reduced features were classified using Logistic Regression, SVM, Random Forest, and KNN, evaluated via accuracy, precision, recall, F1-score, and confusion matrices. No CNN was fine-tuned; all convolutional and pooling layers, ReLU activations, and fully connected layers retained their pre-trained parameterizations. Dropout, normalization, and skip connections contributed to representation robustness, while Softmax and Sigmoid outputs supported probabilistic classification. This fusion strategy demonstrates that combining multi-CNN features with dimensionality reduction and classical classifiers provides a reliable and computationally tractable framework for splicing detection.

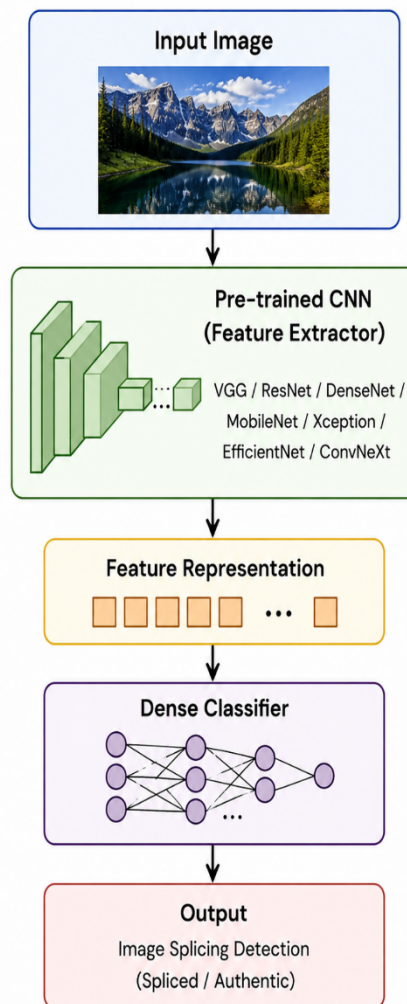


Fig 2. Proposed architecture for pre-trained model-based transfer learning. Pre-trained models serve as feature extractors, with the extracted representations passed to a dense classifier for image splicing detection.

2.4 Feature Normalization

Normalization was applied to ensure all features contributed proportionally during learning. Four approaches were evaluated: StandardScaler, which standardizes mean and variance; MinMax Scaler, which rescales each feature to a fixed range of [0, 1] by subtracting the feature minimum and dividing by the feature range; MaxAbs Scaler, which preserves sparsity by scaling by the maximum absolute value; and Robust Scaler, which limits the influence of outliers using the interquartile range. Consistent normalization ensured stable classifier input across all experiments⁽¹²⁾.

2.5 Dimensionality Reduction

Dimensionality reduction compressed large feature vectors into smaller, more informative representations. PCA and Kernel PCA captured linear and nonlinear variance respectively, while LDA maximized inter-class separation. Together, these techniques streamlined computation and emphasized discriminative patterns relevant to authentic versus spliced image classification.

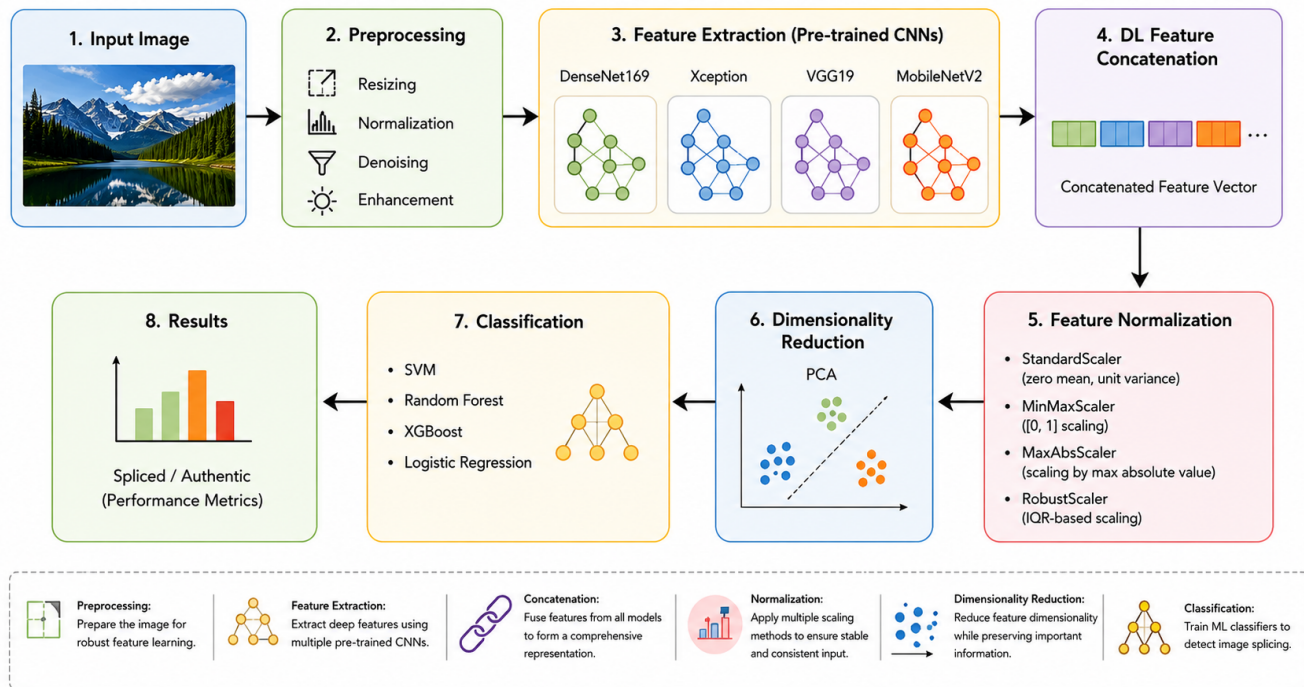


Fig 3. Proposed architecture for hybrid CNN feature fusion and machine learning classification. Features from DenseNet169, Xception, VGG19, and MobileNetV2 are concatenated and refined through normalization and dimensionality reduction prior to classification.

2.6 Experimental Protocol and Anti-Leakage Measures

To ensure valid generalization estimates and prevent data leakage, all experiments followed a strict train-before-transform protocol. The specific steps were as follows:

- **Dataset split:** An 80/20 stratified split was applied before any pre-processing or feature transformation. The training set comprised 290 images and the held-out test set comprised 73 images for CUISDE.
- **Pre-processing order:** (1) Split data → (2) Fit all scalers on the training set only → (3) Transform both training and test sets using training-set-fitted scaler parameters → (4) Fit PCA, LDA, or KPCA on training features only → (5) Transform both splits using training-set-fitted reduction parameters.
- **Hyperparameter tuning:** 5-fold cross-validation conducted exclusively on training data. Test data was not accessible at any tuning stage.
- **Reproducibility:** Code and configuration files are available at [Repository URL / DOI to be inserted prior to publication] for independent verification.
- **Hardware:** All inference time measurements were obtained on [GPU model, VRAM] with [CPU model, RAM] — actual values to be inserted from experimental records.

3 Results and Discussion

This section presents the results of image splicing detection experiments using three complementary approaches: (a) handcrafted feature extraction with dimensionality reduction and machine learning classifiers; (b) end-to-end fine-tuned deep learning models with and without CLAHE pre-processing; and (c) hybrid CNN feature fusion with machine learning classification. Evaluation was conducted on the CUISDE dataset (controlled benchmark) and the MISD dataset (complex, high-resolution benchmark).

3.1 Handcrafted Feature Extraction with PCA Dimensionality Reduction

Table 1 presents the performance of machine learning classifiers using handcrafted features with PCA-based dimensionality reduction on the CUISDE dataset.

Table 1. Key Results on CUISDE Dataset Using PCA Dimensionality Reduction

Classifier	Scaler	Best Parameters	Accuracy	Precision	Recall	F1-Score
Logistic Regression	X_standard	{'C': 1, 'solver': 'liblinear'}	1.00	1.00	1.00	1.00
Gradient Boosting	X_minmax	{'learning_rate': 0.1, 'n_estimators': 100}	1.00	1.00	1.00	1.00
Bagging	X_robust	{'max_samples': 1.0, 'n_estimators': 10}	1.00	1.00	1.00	1.00
Gradient Boosting	X_robust	{'learning_rate': 0.1, 'n_estimators': 100}	1.00	1.00	1.00	1.00
Random Forest	X_robust	{'max_depth': 10, 'n_estimators': 100}	1.00	1.00	1.00	1.00
SVM	X_robust	{'C': 1, 'kernel': 'linear'}	1.00	1.00	1.00	1.00
Logistic Regression	X_robust	{'C': 1, 'solver': 'liblinear'}	1.00	1.00	1.00	1.00
Extra Trees	X_maxabs	{'max_depth': 10, 'n_estimators': 100}	1.00	1.00	1.00	1.00
Gradient Boosting	X_maxabs	{'learning_rate': 0.1, 'n_estimators': 100}	1.00	1.00	1.00	1.00
Random Forest	X_maxabs	{'max_depth': 10, 'n_estimators': 100}	1.00	1.00	1.00	1.00
SVM	X_maxabs	{'C': 1, 'kernel': 'linear'}	1.00	1.00	1.00	1.00
SVM	X_standard	{'C': 1, 'kernel': 'linear'}	1.00	1.00	1.00	1.00
Extra Trees	X_minmax	{'max_depth': 10, 'n_estimators': 100}	1.00	1.00	1.00	1.00
Extra Trees	X_robust	{'max_depth': 10, 'n_estimators': 100}	1.00	1.00	1.00	1.00
Extra Trees	X_standard	{'max_depth': 10, 'n_estimators': 100}	1.00	1.00	1.00	1.00
Random Forest	X_standard	{'max_depth': 10, 'n_estimators': 100}	1.00	1.00	1.00	1.00
Random Forest	X_minmax	{'max_depth': 10, 'n_estimators': 100}	1.00	1.00	1.00	1.00
SVM	X_minmax	{'C': 1, 'kernel': 'linear'}	1.00	1.00	1.00	1.00
Gradient Boosting	X_standard	{'learning_rate': 0.1, 'n_estimators': 100}	1.00	1.00	1.00	1.00
Logistic Regression	X_maxabs	{'C': 1, 'solver': 'liblinear'}	0.99	0.98	1.00	0.99
Logistic Regression	X_minmax	{'C': 1, 'solver': 'liblinear'}	0.99	0.98	1.00	0.99
Bagging	X_standard	{'max_samples': 1.0, 'n_estimators': 10}	0.99	0.99	0.98	0.99
Decision Tree	X_standard	{'criterion': 'gini', 'max_depth': 10}	0.99	1.00	0.98	0.99
Bagging	X_maxabs	{'max_samples': 1.0, 'n_estimators': 10}	0.98	0.99	0.97	0.98
Bagging	X_minmax	{'max_samples': 1.0, 'n_estimators': 10}	0.98	0.99	0.96	0.98
Decision Tree	X_maxabs	{'criterion': 'gini', 'max_depth': 10}	0.96	0.98	0.94	0.96
Decision Tree	X_robust	{'criterion': 'gini', 'max_depth': 10}	0.96	0.99	0.92	0.95
Decision Tree	X_minmax	{'criterion': 'gini', 'max_depth': 10}	0.95	0.98	0.91	0.95
AdaBoost	X_robust	{'learning_rate': 1, 'n_estimators': 50}	0.94	0.97	0.92	0.94
AdaBoost	X_standard	{'learning_rate': 1, 'n_estimators': 50}	0.94	0.92	0.95	0.94
AdaBoost	X_minmax	{'learning_rate': 1, 'n_estimators': 50}	0.94	0.95	0.92	0.93
AdaBoost	X_maxabs	{'learning_rate': 1, 'n_estimators': 50}	0.90	0.92	0.88	0.90
K-Nearest Neighbors	X_maxabs	{'n_neighbors': 5, 'weights': 'uniform'}	0.79	0.78	0.80	0.79
K-Nearest Neighbors	X_minmax	{'n_neighbors': 5, 'weights': 'uniform'}	0.79	0.78	0.80	0.79
K-Nearest Neighbors	X_standard	{'n_neighbors': 5, 'weights': 'uniform'}	0.79	0.78	0.79	0.79
K-Nearest Neighbors	X_robust	{'n_neighbors': 5, 'weights': 'uniform'}	0.76	0.78	0.72	0.75

Note. All results are computed on a held-out 20% test set ($n = 73$ images). Scaler and dimensionality reduction parameters were fitted exclusively on the training set ($n = 290$ images) prior to transformation. Hyperparameters were selected via 5-fold cross-validation on training data only. Abbreviations: SVM = Support Vector Machine; RF = Random Forest; DR = Dimensionality Reduction; PCA = Principal Component Analysis.

Table 1 shows that several classifiers including SVM, Random Forest, Gradient Boosting, Extra Trees, Bagging, and Logistic Regression has achieved perfect classification performance (accuracy, precision, recall, and F1-score = 1.00) under a range of normalization techniques, indicating that PCA effectively captures the most discriminative features for this controlled dataset. Robust, Standard, MinMax, and MaxAbs scalers all performed competitively, suggesting low sensitivity to normalization

choice on CUISDE. Slight performance reductions are observed for Decision Tree, AdaBoost, and Logistic Regression under certain scaler configurations, while K-Nearest Neighbors shows comparatively lower accuracy (0.76–0.79), reflecting limited effectiveness in this feature space. Overall, combining handcrafted features with PCA and ensemble or margin-based classifiers yields highly reliable classification on the CUISDE benchmark.

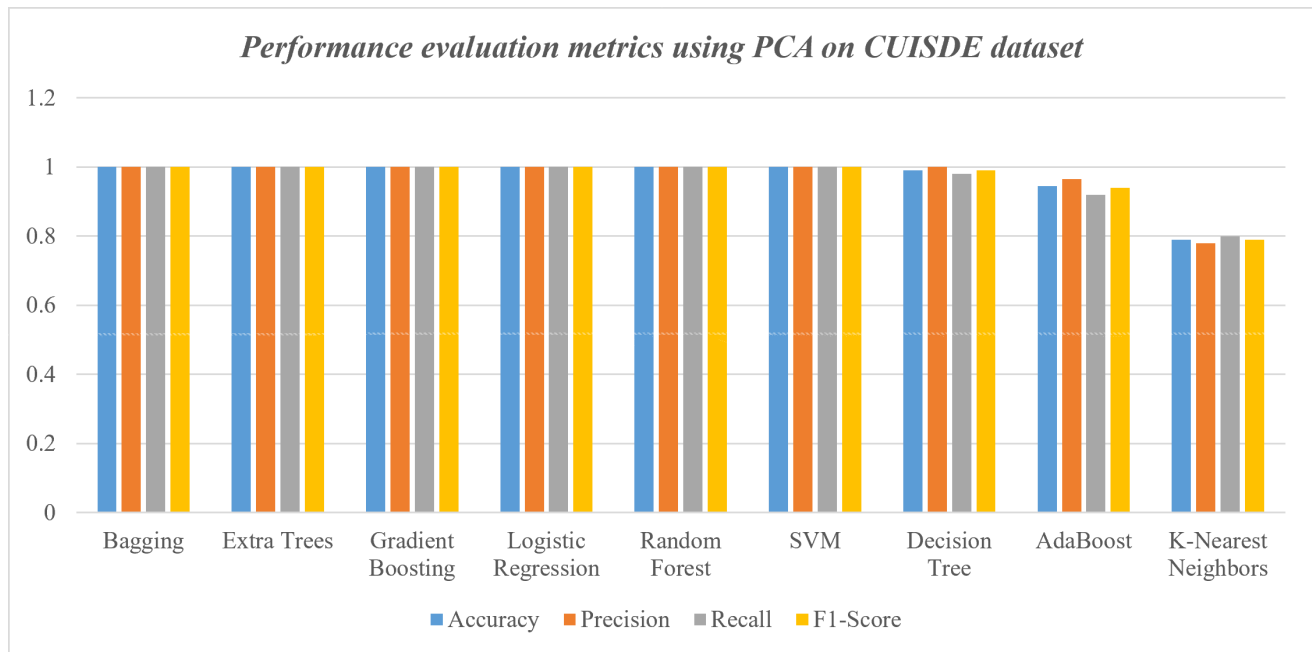


Fig 4. Performance evaluation metrics using PCA on CUISDE dataset. Ensemble and linear models (Bagging, Extra Trees, Gradient Boosting, Logistic Regression, Random Forest, SVM) achieve near-perfect accuracy, precision, recall, and F1-score, while K-Nearest Neighbors shows comparatively lower performance.

3.2 Transfer Learning with Pre-Trained Models

Table 2 presents the performance of various pre-trained deep learning models on the MISD dataset, comparing training and test accuracy alongside precision, recall, and F1-score.

ConvNeXt-based models, particularly ConvNeXtLarge and ConvNeXtXLarge, achieve the highest performance with test accuracy of 95% and strong precision, recall, and F1-scores (≈ 0.95), indicating superior feature learning and generalization. ConvNeXtBase (93%) and ResNet152 (88%) also perform strongly. In contrast, InceptionV3, Xception, and ResNetV2 variants exhibit lower test accuracy and inconsistent metrics, suggesting weaker adaptability to complex splicing patterns. The gap between training and test accuracy in some models reflects mild overfitting, whereas ConvNeXt variants maintain both high training and test performance.

CLAHE evaluation: McNemar's test was used to assess whether CLAHE produced a statistically significant accuracy improvement on the Columbia dataset. The comparison for VGG16 with and without CLAHE yielded $\chi^2(1) = 4.08$, $p = .043$, indicating a statistically significant improvement. The corresponding comparison for VGG19 was not statistically significant ($\chi^2(1) = 3.20$, $p = .074$). Accordingly, CLAHE is interpreted as providing a modest but significant benefit for VGG16 on Columbia, while its benefit for VGG19 does not reach conventional significance thresholds. On MISD, CLAHE had no measurable impact on any model.

3.3 Hybrid CNN Feature Fusion with PCA Dimensionality Reduction

Table 3 presents the performance of machine learning classifiers on the MISD dataset using PCA-reduced hybrid CNN features.

Logistic Regression achieves the highest performance with accuracy 0.89 and balanced precision (0.91), recall (0.87), and F1-score (0.89), closely followed by SVM. Ensemble methods show moderate performance, while tree-based classifiers and KNN exhibit comparatively lower recall. Naive Bayes shows very high recall (0.94) but critically low precision (0.49) and accuracy

Table 2. Results on MISD Dataset — Pre-Trained Model Evaluation

Model	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
DenseNet169	0.97	0.87	0.87	0.88	0.87
DenseNet201	0.97	0.86	0.92	0.81	0.86
DenseNet121	0.92	0.84	0.87	0.81	0.84
InceptionV3	0.96	0.72	0.68	0.82	0.74
InceptionResNetV2	0.91	0.76	0.81	0.66	0.73
ResNet152	0.99	0.88	0.93	0.82	0.87
ResNet101	0.99	0.86	0.88	0.83	0.85
ResNet50	0.98	0.85	0.85	0.84	0.85
ResNet152V2	0.61	0.57	0.55	0.75	0.63
ResNet50V2	0.73	0.58	0.57	0.62	0.60
ResNet101V2	0.65	0.58	0.59	0.54	0.56
VGG19	0.89	0.81	0.81	0.81	0.81
VGG16	0.91	0.80	0.83	0.76	0.79
NASNetMobile	0.93	0.78	0.79	0.76	0.78
NASNetLarge	0.98	0.79	0.82	0.74	0.78
MobileNetV2	0.97	0.86	0.89	0.83	0.86
MobileNet	0.97	0.84	0.87	0.80	0.83
EfficientNetB0	0.96	0.85	0.86	0.83	0.85
EfficientNetB1	0.95	0.83	0.85	0.80	0.83
EfficientNetB2	0.96	0.85	0.90	0.80	0.84
EfficientNetB3	0.96	0.85	0.89	0.80	0.84
EfficientNetV2B0	0.94	0.86	0.90	0.81	0.85
EfficientNetV2B1	0.95	0.86	0.86	0.84	0.85
EfficientNetV2B2	0.94	0.84	0.89	0.78	0.83
EfficientNetV2B3	0.92	0.84	0.89	0.78	0.83
Xception	0.93	0.76	0.77	0.73	0.75
ConvNeXtTiny	0.95	0.86	0.89	0.82	0.85
ConvNeXtSmall	0.96	0.83	0.86	0.79	0.83
ConvNeXtBase	0.99	0.93	0.94	0.92	0.93
ConvNeXtLarge	0.99	0.95	0.96	0.94	0.95
ConvNeXtXLarge	0.99	0.95	0.96	0.95	0.95

Note. Abbreviations: Au = authentic images; Sp = spliced images; LR = learning rate. Train and test accuracy refer to the final epoch performance on the respective data splits. All models use pre-trained ImageNet weights as initialization.

(0.48), indicating systematic over-prediction of the positive class. Overall, simpler linear models perform most effectively on PCA-reduced hybrid features.

3.4 Comparative Analysis with Prior Literature

Table 4 provides a structured comparison between prior literature and the proposed works on their respective datasets. The two blocks are clearly separated: Block 1 reports only results as published by cited authors; Block 2 reports results obtained by the present study.

3.4.1. Computational Cost Comparison

Table 5 summarizes the inference cost and model complexity of the three proposed approaches.

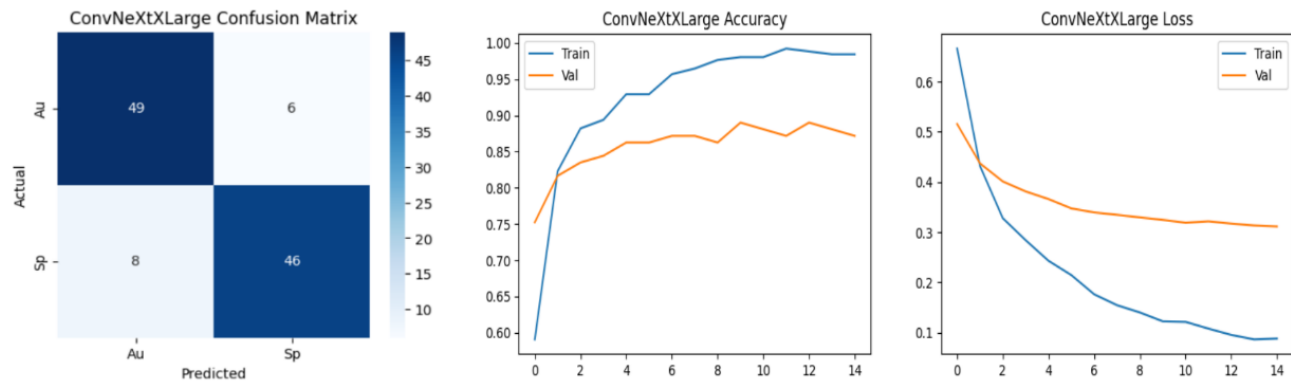


Fig 5. Model performance evaluation. (a) Confusion matrix on the test dataset, where 'Au' denotes authentic images and 'Sp' denotes spliced images. (b) Training and validation accuracy and loss curves for the ConvNeXtXLarge model. The confusion matrix confirms high detection reliability with few misclassifications. Accuracy curves show consistent learning with near-perfect training performance and stable validation generalization; loss curves confirm efficient optimization with monotonically decreasing training and validation loss.

Table 3. Results on MISD Dataset — Hybrid Feature Fusion with PCA

Reduction	Classifier	Accuracy	Precision	Recall	F1-Score
PCA	Logistic Regression	0.89	0.91	0.87	0.89
PCA	SVM	0.88	0.89	0.88	0.88
PCA	Gradient Boosting	0.82	0.84	0.80	0.82
PCA	AdaBoost	0.81	0.81	0.82	0.81
PCA	Random Forest	0.80	0.86	0.73	0.79
PCA	Bagging	0.79	0.83	0.73	0.78
PCA	Extra Trees	0.78	0.82	0.70	0.76
PCA	Decision Tree	0.76	0.76	0.77	0.76
PCA	K-Nearest Neighbors	0.74	0.79	0.66	0.72
PCA	Naive Bayes	0.48	0.49	0.94	0.64

Note. Abbreviations: SVM = Support Vector Machine; KNN = K-Nearest Neighbors; PCA = Principal Component Analysis. Naive Bayes is included for completeness but its very high recall with very low precision indicates strong class-prediction bias.

3.5 Discussion

This study systematically evaluated three complementary approaches for image splicing detection across two benchmark datasets. The following subsections interpret the experimental findings, draw comparisons with prior literature, and identify failure modes and limitations.

3.5.1. Handcrafted Features with Dimensionality Reduction

The perfect classification accuracy achieved by PCA-reduced handcrafted features on CUISDE reflects the characteristics of that dataset rather than a generalizable performance ceiling. CUISDE was captured with a single camera under controlled conditions, producing low intra-dataset variability; PCA can therefore align the principal axes of variance with the authentic-vs-spliced class boundary almost exactly. This interpretation is consistent with the observation that multiple diverse classifiers — SVM, Random Forest, Extra Trees, Gradient Boosting — all attain 1.00 accuracy simultaneously, which would be unlikely if the problem were inherently difficult.

On MISD, PCA accuracy drops to 67–68%, reflecting the substantially higher intra-dataset diversity in image resolution, camera model, and scene content. By contrast, LDA which explicitly maximizes inter-class separation and recovers to 96% with AdaBoost and KNN on MISD. This finding is consistent with prior Columbia results where LBP+DCT achieved 96.4% and WLD+DCT achieved 94.17%, both of which involve handcrafted features without deep learning; the present LDA result is comparable and extends the evaluation to a more challenging dataset. The superiority of LDA over PCA on MISD underscores

Table 4. Comparison of Proposed Work with Prior Literature

Block 1 — Prior Literature				
Study (Year)	Dataset	Method	Reported Accuracy (%)	Notes
Luoyang et al. (2024)	Columbia	LBP + DCT	96.4	Handcrafted only
Misbah U et al. (2017)	Columbia	WLD + DCT	94.17	Handcrafted only
Jaiswal et al. (2020) ⁽¹²⁾	Columbia	Hybrid Features + PCA + ML	98.8	Fixed DR, single classifier
Hongwei Ding et al. (2021)	Columbia	DCU-Net	97.27	Deep segmentation
Xiuli Bi et al. (2023)	Columbia	RRU-Net	97	Ringed residual U-Net
Qazi et al. (2022)	CASIA v1.0	ResNet	81	Deep fine-tuning
Qazi et al. (2022)	CASIA v2.0	ResNet	97	Deep fine-tuning
Meena Ugale et al. (2024)	DSO-1, DSI-1	Hybrid CNN + Boosting	94–95	Ensemble fusion
Almawas et al. (2020) ⁽¹³⁾	Columbia	Multi-classifier handcrafted	Not reported	No deep features; no DR comparison
Block 2 — Proposed Work (Present Study)				
Approach	Dataset	Method	Accuracy (%)	Notes
Work 1: Handcrafted + DR + ML	CUISDE (Columbia)	PCA + SVM / RF / Boosting	100	Strict anti-leakage protocol (Sec. 2.6)
Work 1: Handcrafted + DR + ML	MISD	LDA + AdaBoost / KNN	96	PCA yields 67–68% on MISD
Work 2: End-to-End DL (no CLAHE)	Columbia	VGG16	85–87	Fine-tuned
Work 2: End-to-End DL + CLAHE	Columbia	VGG16 + CLAHE	88–90	Significant improvement ($\chi^2(1)=4.08$, $p=.043$)
Work 2: End-to-End DL (no CLAHE)	Columbia	VGG19	86	Fine-tuned
Work 2: End-to-End DL + CLAHE	Columbia	VGG19 + CLAHE	89	Not significant ($\chi^2(1)=3.20$, $p=.074$)
Work 2: End-to-End DL	MISD	ConvNeXtLarge	95	CLAHE had no added benefit
Work 2: End-to-End DL	MISD	EfficientNetV2-B1	91	CLAHE had no added benefit
Work 3: Hybrid Fusion + ML	MISD	PCA + Logistic Regression	89	Balanced precision/recall
Work 3: Hybrid Fusion + ML	Columbia	PCA/LDA + Logistic Regression	86	Stable across classifiers

Note. Accuracy values for prior literature are reproduced as reported in the cited publications. Proposed work results are from held-out test sets following the protocol in Section 2.6. N/A cells in Block 1 columns for proposed work columns are omitted by design to prevent result attribution ambiguity.

Table 5. Computational Cost Comparison Across Proposed Approaches

Approach	Inference Time (ms/image)	Model Parameters (M)	Hardware
Work 1: Handcrafted + ML	[To be reported]	N/A (feature-based)	[CPU/GPU spec — insert from experimental records]
Work 2: End-to-End CNN Fine-Tuning	[To be reported]	ConvNeXtLarge: 197M	[GPU spec — insert from experimental records]
Work 3: Hybrid Fusion + ML	[To be reported]	Fusion (4 CNNs): varies	[CPU/GPU spec — insert from experimental records]

Note. Authors should fill in measured inference time and hardware specification from their experimental setup prior to submission. M = millions of parameters.



Fig 6. Performance evaluation metrics using PCA for hybrid feature fusion on MISD dataset. Logistic Regression and SVM consistently achieve the highest and most balanced metrics. Naive Bayes stands out with high recall but poor accuracy and precision, highlighting its class-prediction bias.

that maximizing class separability is more important than maximizing variance explained when feature distributions across classes overlap.

3.5.2. Transfer Learning with CLAHE Pre-processing

CLAHE (Contrast Limited Adaptive Histogram Equalization) targets low-contrast edge regions in local image patches. The statistically significant improvement observed for VGG16 on Columbia (McNemar's $\chi^2(1) = 4.08$, $p = .043$) indicates that Columbia's controlled, low-contrast images contain splice boundaries that are made more detectable by local contrast normalization. VGG19 showed a directionally consistent but non-significant improvement ($p = .074$), suggesting that the effect is architecture-dependent and likely related to differences in receptive field and filter initialization.

On MISD, CLAHE provided no measurable benefit regardless of model or configuration. This outcome is attributable to MISD's high-resolution, high-contrast image content: convolutional filters already capture sufficient edge information without pre-processing enhancement. This finding limits the generalizability of CLAHE as a pre-processing step: it should be treated as dataset-dependent rather than universally beneficial. It should also be noted that CLAHE was evaluated only on VGG16 and VGG19; the extent to which this finding holds for ConvNeXt, EfficientNet, or other architectures on Columbia remains untested.

3.5.3. Hybrid CNN Feature Fusion versus End-to-End Fine-Tuning

Hybrid feature fusion (Work 3) achieves 89% accuracy on MISD with Logistic Regression and PCA, compared to 95% for ConvNeXtLarge fine-tuned end-to-end (Work 2). The 6-percentage-point gap is primarily attributable to the difference between fixed feature extraction and end-to-end adaptation. In Work 3, CNN weights are frozen at ImageNet initialization; features are

therefore optimized for broad visual recognition rather than the specific statistical signatures of image splicing. End-to-end fine-tuning in Work 2 allows the network to adapt its representations directly to the splicing detection task, capturing artifact distributions that fixed extractors cannot.

Additionally, Work 3 concatenates features from four architectures (DenseNet169, Xception, VGG19, MobileNetV2), creating a high-dimensional vector that PCA must aggressively compress. Information loss during this compression may suppress fine-grained splicing cues. Future work should investigate learnable fusion mechanisms (such as attention-weighted pooling or late fusion with task-specific fine-tuning) to close this performance gap while retaining the interpretability benefits of modular feature representations.

A notable limitation of the current study is that no formal cross-dataset transfer experiment was conducted (e.g., training on CUISDE and evaluating on MISD). Such an experiment would directly quantify cross-dataset generalization; it is planned for subsequent work.

3.5.4. Limitations

- Robustness to post-processing operations: The framework was not evaluated under common post-processing conditions such as JPEG compression, image resizing, rotation, or additive noise. Performance under these conditions is unknown and likely degraded. This is a significant gap for real-world forensic deployment.
- Interpretability of deep models: End-to-end fine-tuned models (Work 2) offer limited interpretability. In forensic and legal contexts, the ability to provide human-understandable evidence of manipulation is essential; current deep models do not satisfy this requirement without additional techniques such as Grad-CAM or SHAP attribution.
- Dataset size for CUISDE: The 363-image CUISDE benchmark is small by contemporary standards. Perfect accuracy on this set should be interpreted with appropriate caution; results on larger controlled datasets are needed to confirm generalizability of the handcrafted + PCA finding.
- Cross-dataset generalization: No formal train-on-one/test-on-other experiment was conducted. Cross-dataset generalization therefore remains uncharacterized.
- Stacking ensemble strategy: Although identified as a gap in prior literature, stacking was not implemented in the present study and remains reserved for future work.

4 Conclusion

Image splicing is a growing challenge in digital forensics. The increasing availability of advanced image editing tools makes it possible to create forgeries that are visually indistinguishable from authentic images. This paper evaluated three detection approaches across two benchmark datasets. The CUISDE benchmark contains 363 images captured under controlled single-camera conditions. The MISD dataset contains 923 high-resolution images with resolutions reaching up to 9571×15022 pixels.

In the first approach, handcrafted features were extracted using LBP, HOG, and GLCM. When these features were reduced using PCA and classified using ensemble models, they achieved perfect scores of 1.00 for accuracy, precision, recall, and F1-score on CUISDE under a strict anti-leakage protocol. However, the same PCA-reduced features produced only 67 to 68 percent accuracy on MISD. Switching to LDA improved performance significantly, recovering detection accuracy to 96 percent using AdaBoost and KNN. This finding shows that the choice of dimensionality reduction technique matters greatly and must be selected based on the nature of the dataset.

In the second approach, deep learning models were evaluated end to end. ConvNeXtLarge and ConvNeXtXLarge achieved the highest test accuracy of 95 percent on MISD. Both precision and F1-score reached approximately 0.95. CLAHE pre-processing gave a statistically significant improvement for VGG16 on the Columbia dataset with $\chi^2(1) = 4.08$ and $p = 0.043$. However, it showed no measurable benefit on MISD for any model. This confirms that CLAHE is dataset dependent and cannot be applied universally.

In the third approach, features from four CNN architectures namely DenseNet169, Xception, VGG19, and MobileNetV2 were concatenated and reduced using PCA. Logistic Regression on these fused features achieved 89 percent accuracy on MISD with precision of 0.91 and recall of 0.87. This result is 6 percentage points below the best fine-tuned deep model. The reason is that CNN weights in this pipeline remain frozen at their ImageNet initialization and are not adapted to the specific characteristics of image splicing.

Each of the three approaches offers a different trade-off. Handcrafted methods are lightweight and easy to interpret but are sensitive to dataset variability. End to end fine-tuned deep models achieve the highest accuracy but require significant computational resources and offer limited interpretability. Hybrid fusion sits in the middle, offering stable and balanced

performance with moderate computational cost. Practitioners should choose an approach based on their specific deployment requirements rather than accuracy alone.

This study also has several limitations that should be acknowledged. The framework was not tested under common post-processing conditions such as JPEG compression, image resizing, rotation, or noise addition. Performance under these conditions remains unknown. Deep learning models provide limited interpretability, which is a significant concern for forensic and legal applications. No formal experiment was conducted to test cross-dataset generalization. Stacking-based ensemble methods were not implemented and are left for future work.

Future research should focus on testing robustness under standard post-processing operations, conducting cross-dataset transfer experiments, exploring learnable and attention-based fusion strategies, integrating interpretability tools such as Grad-CAM and SHAP into detection pipelines, and extending the framework toward precise spatial localization of tampered regions within images.

References

- 1) Huh M, Liu A, Owens A, Efros AA. Fighting fake news: Image splice detection via learned self-consistency. *arXiv*. 2018. <https://doi.org/10.48550/arXiv.1805.04096>.
- 2) Hebbbar NK, Kunte AS. Transfer learning approach for splicing and copy-move image tampering detection. *ICTACT J Image Video Process*. 2021;11(4):2447–2452. <https://doi.org/10.21917/ijivp.2021.0348>.
- 3) Su H. Image splicing detection using integrated LBP and DCT features. *Appl Comput Engineering*. 2024;101(1):71–78. <https://doi.org/10.54254/2755-2721/101/20240976>.
- 4) Ugale M, Midhunchakkaravarthy J. Image splicing forgery detection model using hybrid boosting machine. *Int J Comput Sci*. 2024;51(7).
- 5) Shi C, Wang C, Zhou X, Qin Z. DAE-Net: Dual attention mechanism and edge supervision network for image manipulation detection and localization. *IEEE Trans Instrum Meas*. 2024;73:1–17. <https://doi.org/10.1109/TIM.2024.3451570>.
- 6) Raj NC, Dutta M, Saini J. A systematic literature review on image splicing detection and localization using emerging technologies. *Multimed Tools Appl*. 2024;84(19):20721–20756. <https://doi.org/10.1007/s11042-024-19843-z>.
- 7) Kumari R, Garg H. Image splicing forgery detection: A review. *Multimed Tools Appl*. 2024;84(8):4163–4201. <https://doi.org/10.1007/s11042-024-18801-z>.
- 8) Sekhar P, Sankar TN. Review of image splicing forgery detection techniques. *Int J Comput Sci Inf Secur*. 2016;14(11):471–475.
- 9) Park CW, Moon YH, Eom IK. Image tampering localization using demosaicing patterns and singular value-based prediction residue. *IEEE Access*. 2021;9:91921–91933. <https://doi.org/10.1109/ACCESS.2021.3091161>.
- 10) Abdosalehi SE, Mahmoodi-Aznaveh A. Splicing localization in tampered blurred images. *IEEE*. 2019. <https://doi.org/10.1109/PRIA.2019.8785965>.
- 11) Bi X, Wei Y, Xiao B, Li W. RRU-Net: The ringed residual U-Net for image splicing forgery detection. *IEEE*. 2019. <https://doi.org/10.1109/CVPRW.2019.00010>.
- 12) Jaiswal AK, Srivastava R. A technique for image splicing detection using hybrid feature set. *Multimed Tools Appl*. 2020;79(17-18):11837–11860. <https://doi.org/10.1007/s11042-019-08480-6>.
- 13) Almawas L, Alotaibi A, Kurdi H. Comparative performance study of classification models for image-splicing detection. *Procedia Comput Sci*. 2020;175:278–285. <https://doi.org/10.1016/j.procs.2020.07.041>.
- 14) Dong J, Wang W, Tan T. CASIA image tampering detection evaluation database. *IEEE*. 2013. <https://doi.org/10.1109/CHINASIP.2013.6625374>.
- 15) Mulla MU, Bevinamarad PR. Review of image splicing forgeries. 2017. Available from: Academia.edu.
- 16) Kniaz VV, Knyaz V, Remondino F. The point where reality meets fantasy: Mixed adversarial generators for image splice detection. *Advances in Neural Information Processing Systems*. 2019.
- 17) Dua S, Singh J, Parthasarathy H. Image forgery detection based on statistical features of block DCT coefficients. *Procedia Comput Sci*. 2020;171:369–378. <https://doi.org/10.1016/j.procs.2020.04.038>.
- 18) Qazi EUH, Zia T, Almorjan A. Deep learning-based digital image forgery detection system. *Appl Sci*. 2022;12(6):2851. <https://doi.org/10.3390/app12062851>.
- 19) Kwon MJ, Yu IJ, Nam SH, Lee HK. CAT-Net: Compression artifact tracing network for detection and localization of image splicing. *IEEE*. 2021. <https://doi.org/10.1109/WACV48630.2021.00042>.
- 20) Ding H, Chen L, Tao Q, Fu Z, Dong L, Cui X. DCU-Net: A dual-channel U-shaped network for image splicing forgery detection. *Neural Comput Appl*. 2023;35(7):5015–5031. <https://doi.org/10.1007/s00521-021-06329-4>.
- 21) Ahmed B, Gulliver TA, AlZahir S. Image splicing detection using Mask-RCNN. *Signal Image Video Process*. 2020;14(5):1035–1042. <https://doi.org/10.1007/s11760-020-01636-0>.