JOURNAL ON COMMUNICATIONS

ISSN:1000-436X

Scopus®

REGISTERED

www.jocs.review

Explainability in Image Forensics: A Lime – SVM Based Classification Approach Using LBP and DCT based Feature Extraction

Deepali Joshi [1st]
Dept. of Information Technology
Vishwakarma Institute of Technology
Pune, India

Tejas Wasekar [2nd]
Dept. of Information Technology
Vishwakarma Institute of Technology
Pune, India

Shubham Tambe [3rd]
Dept. of Information Technology
Vishwakarma Institute of Technology
Pune, India

Akanksha Wagaskar [4th]
Dept. of Information Technology
Vishwakarma Institute of Technology
Pune, India

Shruti Sood [5th]
Dept. of Information Technology
Vishwakarma Institute of Technology
Pune, India

Aniket Thenge [6th]
Dept. of Information Technology
Vishwakarma Institute of Technology
Pune, India

Abstract*—Image investigation systems maintain fundamental importance in digital information security. The correct and secure management of digital content depends on image forgery detection techniques. This research presents an effective method with explained functionality to detect forged images by using Local Binary Patterns (LBP) and Discrete Cosine Transform (DCT) features extracted from chrominance components. Local Binary Patterns (LBP) provide image forgery detection capability. The selected features for image forgery detection include LBP together with DCT obtained from image chrominance components. The proposed method segments the C-band Cr channels into blocks and applies LBP followed by DCT to capture textual and frequency-based artifacts indicative of tampering. These features are then classified using a Linear Support Vector Machine (SVM), achieving an accuracy of 92 percent in five-fold cross-validation on the combined CASIA 1.0 and CASIA 2.0 datasets. To enhance model transparency, we integrate the Local Interpretable Model-agnostic Explanations (LIME) framework, providing insight into the classifier's decision-making process. This study proves the utility of the proposed method through measurement results, advancing the interpretability of the proposed approach and the reliability of image forensic systems.

Index Terms*—Image Forgery Detection, Local Binary Patterns (LBP), Discrete Cosine Transform (DCT), Chrominance Features, Linear Support Vector Machine (SVM), Explainable AI (XAI), Local Interpretable Model-agnostic Explanations (LIME), Image Forensics, CASIA Dataset, Image Classification.

I. Introduction

Digital image forensics requires effective image forgery detection within current times when manipulations of images affect journalism and judicial proceedings as well as public belief systems. Advanced image editing software has turned it into a troublesome task to separate authentic images from tampered ones. The present traditional tampering artefact identification tools in image forensics lack interpretability capabilities that become crucial when dealing with high-stakes applications.

Research in Explainable Artificial Intelligence (XAI) has produced powerful technologies that help increase the transparency of artificial intelligence models. A research design presents an explainable image forgery detection framework which integrates handcrafted feature extraction with a transparent classification system.

The system first uses Local Binary Pattern (LBP) for chrominance texture feature extraction from image blocks before Discrete Cosine Transform (DCT) analyzes frequency characteristics. The calculation of transformed feature standard deviation shows how blocks differ from one another. The system obtains separate features from Cr and Cb channels of the YCbCr color space which it combines into a unified feature vector.

The Linear Support Vector Machine (SVM) utilizes the feature vectors obtained to create a training system. The chosen SVM design demonstrates successful implementation for simplicity reasons. We added Local Interpretable Model-agnostic Explanations (LIME) to the model to generate visual explanations of important features that affect each prediction.

Combining LBP features with DCT features and standard deviation-based features alongside Linear SVM explainability using LIME builds an effective detection system where users can understand model decisions. The proposed system demonstrates cross-validation accuracy of 92% which validates its potential practical use for image forensics purposes.

II. Literature Survey

Modern multimedia technologies and conveniently accessible image editing software individuals now face greater risks

for image forgery. Splicing techniques applied to open communication networks allow unauthorized users to edit images by merging different image sections. Such methods create significant consequences for public confidence in addition to exposing problems in court systems. DWT and DRLBP together form an effective detection method through combined use of Discrete Wavelet Transform and Discriminative Robust Local Binary Patterns histogram calculation.Testing occurred using benchmark datasets that demonstrated an accuracy rate of 98.95% while showing better detection performance compared to modern digitized forgery detection systems. The model demonstrates superior abilities to uncover slight image manipulations occurring in spliced areas.[1]

Research has focused on different forgery techniques such as copy-move, splicing, and imitation, typically employing Local Binary Pattern (LBP), Discrete Cosine Transform (DCT), and Support Vector Machines (SVM) for detection. However, existing methods mainly target generic image content rather than official documents. Most methods assess datasets using standard metrics but fail to address real-world document security challenges. This underscores the necessity for more sophisticated, dual-layer detection frameworks that simultaneously evaluate both surface-level and contextual integrity, particularly in the detection of forgery within formal administrative content.[2]

The verification of digital images plays an essential role for journalism and insurance and law because original visual evidence requires authenticity. Passive approaches evaluate an image's statistical data to identify tampering evidence through irregularities. Three major signals that reveal tampering are compression artifacts combined with lighting variations and noise appearance. The wide usefulness of passive methods does not prevent any detection approach from effectively identifying all types of forgery at present. The situation requires innovative approaches which unite two or more detection methods into one analytical solution. The latest research focuses on developing hybrid and machine learning systems to enhance detection capacity for different images and manipulation techniques.[3]

The modern digital environment uses images as its main method of data transmission throughout media platforms and educational institutions and healthcare facilities and political organizations. The easy access to mobile cameras along with advanced editing software enables everyone to manipulate images thus creating doubts about their authenticity. When used for entertainment purposes forged images provide benefits yet their employment in purposes of misinformation and defamation remains unethical. Manipulated images create substantial damage to people while simultaneously lowering the reliability of news sources and impacting what the public believes to be accurate. Detection tools need to remain effective for countering the risks identified. The field of image forensics investigates both classical detection methods and machine learning algorithms as solutions to eliminate image forgery. This assessment includes a detailed study that investigates multiple detection techniques including supervised and unsupervised learning techniques and current deep learning systems. The field requires scalable real-time systems since existing solutions show limited performance under compression and transformations. The author promotes future developments which should unite contextual knowledge with effective computational capabilities.[4]

The rapid progress in multimedia technologies has led to widespread availability of image and video editing tools, which pose serious challenges in verifying media authenticity. These tools, while beneficial for casual users, are also exploited for malicious purposes like identity theft and defamation. Detecting tampered regions in multimedia content remains complex due to subtle alterations. The paper conducts a comprehensive review of common manipulation techniques including splicing, cloning, and deepfakes. It highlights the limitations of current detection models and datasets, emphasizing the need for robust frameworks capable of handling diverse forgery types. The study also evaluates publicly available datasets and calls for standardized, multi-modal datasets that reflect real-world forgery complexities. The long-term vision involves developing privacy-aware, universal detection systems that can assist law enforcement and digital platforms in verifying content authenticity and maintaining trust in digital communications.[6]

Image authentication plays a critical role in distinguishing genuine images from manipulated ones, particularly when minor alterations such as compression are permissible. Traditional approaches often fail to differentiate acceptable modifications from malicious tampering. A technique that exploits the stable relationships of Discrete Cosine Transform (DCT) coefficients across JPEG blocks offers a more precise detection strategy. The model identifies inconsistencies introduced by tampering while tolerating artifacts from legitimate compression. It adapts effectively across different compression levels and recompression cycles, making it robust in real-world applications. This approach sets itself apart by balancing sensitivity to edits with tolerance for routine image processing.[18]

Digital image editing software coupled with image-sharing services allows for more discreet image manipulation which turns into simple transformations like resizes and JPEG compression. An innovative CNN-based detection system detects camera origins while focusing on compression resistance through its networks design. The model becomes capable of generalized detection by receiving training through image data sets with varying compression levels. The technique demonstrates superior performance in accuracy combined with resilience over conventional detection methods according to performance measurements. Model detection processes gain transparency through visual analysis of intermediate layers which enables users to trust predicted outcomes better.[15]

A statistical framework based on wavelet decomposition has been developed as an aid for digital forensic analysis of natural images. Such technique extracts both first-order statistical data and second-order data which defines common image patterns. The method detects tampered regions through its ability to detect unexpected statistical patterns while needing

no information about the forgery locations beforehand. The detection system operates automatically while preventing the requirement for embedded metadata or watermarks. Wavelet analysis proves excellent for detecting very small alterations in images which standard inspection methods miss while performing across multiple tampering techniques including splicing and region duplication. [20]

A passive tampering detection method was developed based on estimating JPEG quantization tables. The technique begins by analyzing AC DCT coefficients and estimating quantization step sizes using Power Spectrum Density (PSD) analysis and Fourier transforms. It then isolates candidate regions presumed to be untampered, excluding suspicious zones. These regions are expanded through a region-growing method to minimize inclusion of altered content. The estimated quantization table serves as a reference for block-level inconsistency checks, employing a Maximum Likelihood Ratio (MLR) classifier to detect tampered areas. Experimental validations show high accuracy in identifying manipulated sections, especially in compressed images.[19]

III. System Architecture

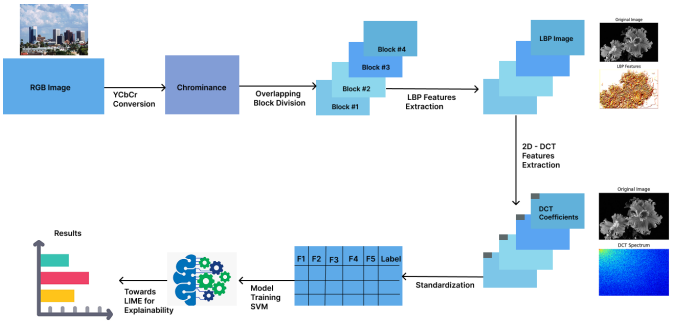

Fig. 1. Proposed System Architecture for Explainable Image Forgery Detection

The system architecture for the proposed image forgery detection method integrates both Local Binary Pattern (LBP) and Discrete Cosine Transform (DCT) based feature extraction techniques for robust chrominance analysis. The RGB input image is first converted to the YCbCr color space, and chrominance channels are processed using overlapping block division. LBP features are extracted and transformed using 2D-DCT, resulting in compact descriptors. These features are standardized and fed into a Linear SVM classifier for binary classification (real vs fake). To enhance interpretability, LIME is applied to explain the SVM model's predictions. This pipeline ensures both accuracy and explainability in digital image forensics.

IV. Methodology

A. Dataset Description

The research used two image forgery detection datasets which were developed by the Institute of Automation Chinese Academy of Sciences (CASIA) and named CASIA Image Tampering Detection Evaluation Database v1.0 and v2.0. These datasets help establish reliable detection tests because they present diverse manipulation methods across numerous image characteristics with regard to content materials and compression techniques and picture resolution types.

The CASIA v2.0 database presents a substantial challenge by providing 14,928 images which include 7,437 real and 7,491 forged samples. This dataset shows complex tampered pictures which include copy-move, splicing and object removal manipulations that human observers find challenging to identify. Advanced editing programs perform tampering that requires JPEG compression to store the data but introduces multiple processing artifacts along with additional image noise to the data.

During the initial stage the dataset contained uneven class distributions that could result in training biases of the model system. The chosen data augmentation techniques included flipping as well as rotation along with minor geometric transformations to achieve balanced classes in the dataset and improve model generalization power.

Our proposed method consisting of chrominance-based separation, block-based Local Binary Patterns and Discrete Cosine Transform extracted features for all images contained in the dataset.

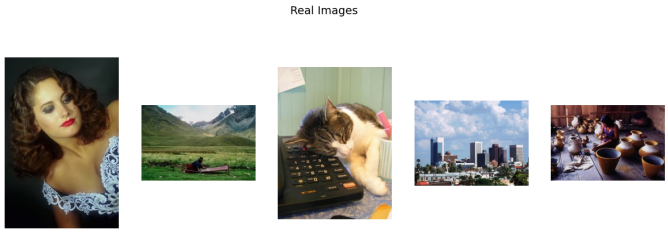

Fig. 2. Images present in the Authentic folder of CASIA 2.0 dataset.

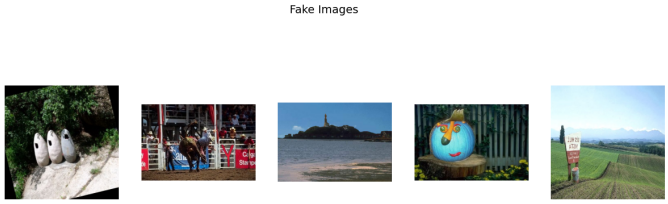

Fig. 3. Shows the Images Present in Tampered Folder of CASIA 2.0

On the other hand, the CASIA v1.0 dataset, though smaller in size, serves as an important complementary benchmark for validating the generalization of our approach. It consists of 1,711 images, with 790 real and 921 tampered samples. The tampering in this dataset primarily involves cut-and-paste (splicing) and copy-move operations, which are relatively simpler compared to the manipulations found in v2.0. However, the images are still useful for training and testing, especially in cases where models are evaluated for robustness across multiple datasets.

To create a more diverse and challenging evaluation environment, we combined both datasets into a single unified dataset. This merged dataset brings together variations in manipulation types, image sources, and quality levels, allowing the model to learn from a richer distribution of forgeries. Such a combination not only improves detection accuracy but also enhances the explainability and robustness of the system when deployed on real-world data.

B. Preprocessing Pipeline

To ensure consistent and meaningful feature extraction, all input images undergo a systematic *preprocessing* pipeline. Initially, each RGB image is converted into the $\mathbf{YC_rC_b}$ color space, a widely adopted color representation in image processing tasks. The $\mathbf{YC_rC_b}$ space separates an image's **luminance component (Y)** from its **chrominance components (Cr and Cb)**, where Cr represents the red-difference and Cb the blue-difference chroma channels.

This separation is crucial for image forgery detection as manipulations often leave subtle traces in the chrominance components due to inconsistencies in compression, blending, or lighting that are less perceptible in luminance.

Mathematically, the transformation from RGB to $\mathbf{YC_rC_b}$ is defined as:

$$Y = 0.299R + 0.587G + 0.114B$$

$$Cr = (R - Y) \times 0.713 + 128$$

$$Cb = (B - Y) \times 0.564 + 128$$

This formulation maps the RGB space into luminance (Y) and two chroma components (Cr and Cb), with added offsets (typically 128) to maintain the dynamic range for digital image representation. For example, consider an RGB pixel with values R = 100, G = 150, and B = 200. **Applying the above transformation yields:**

$$Y = 0.299 \times 100 + 0.587 \times 150 + 0.114 \times 200 = 137.15$$

$$Cr = (100 - 137.15) \times 0.713 + 128 \approx 102.5$$

$$Cb = (200 - 137.15) \times 0.564 + 128 \approx 163.5$$

This step highlights how chrominance encodes *color* deviation information independently of brightness, making it especially sensitive to manipulation artifacts.

Once the $\mathbf{YC_rC_b}$ representation is obtained, the **Y (luminance) channel is discarded**, and only the Cr and Cb channels are retained for further analysis. The rationale behind this is that many tampering operations—like splicing or copy-move forgeries—introduce subtle chromatic aberrations that are hard to detect in the intensity domain but become more evident when examining the *color* distribution patterns.

These two chrominance channels (Cr and Cb), now separated, form the input for the block-wise texture and frequency-based analysis in the subsequent feature extraction stage.

C. Feature Extraction

To effectively distinguish between authentic and tampered images, our system leverages a hybrid feature extraction pipeline based on both texture and frequency domain analysis. The feature extraction process consists of multiple stages, namely: chrominance channel isolation, patch-wise LBP computation, DCT transformation, and statistical aggregation. Each step is designed to amplify the subtle inconsistencies introduced during image manipulation.

The first step involves converting each input image from the RGB color space to the YCrCb color space. This transformation is crucial because it separates luminance (Y) from chrominance information (Cr and Cb). In our approach, we discard the Y channel and retain only Cr and Cb, since forgeries tend to introduce color inconsistencies more noticeably than intensity variations.

Once the Cr and Cb channels are isolated, the image is divided into smaller overlapping blocks using a sliding window approach. For instance, using a block size of 32×32 pixels and a stride of 16 ensures that each region of the image is covered multiple times, enhancing robustness. Each of these blocks is then treated as an independent unit for further processing.

Within each block, we apply the Local Binary Pattern (LBP) operator to extract texture descriptors. LBP is a powerful yet simple method to describe local spatial patterns. For a given central pixel in a grayscale image, LBP works by comparing it with its surrounding neighbors. If the neighbor pixel value is greater than or equal to the center pixel, a 1 is assigned; otherwise, a 0. This results in a binary pattern of length P (number of neighbors), which is then converted into a decimal number. Mathematically, the LBP value at a pixel (x, y) is given by:

$$LBP_{P,R}(x, y) = \sum_{p=0}^{P-1} s(i_p - i_c) \cdot 2^p \tag{1}$$

where i_c is the intensity of the center pixel, i_p is the intensity of the p-th neighbor, and the function $s(x)$ is defined as:

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

For example, suppose a 3×3 patch has a center pixel value of 50, and surrounding pixel values are [55, 60, 45, 49, 51, 30, 20, 70]. Comparing each neighbor with the center (50) results in a binary pattern like 1 1 0 0 1 0 0 1, which converts to the decimal value 201.

Once the LBP features are computed for both the Cr and Cb channels of each block, we apply the Discrete Cosine Transform (DCT) to capture frequency domain information. DCT is particularly useful because it compacts the signal energy into a few coefficients, emphasizing sharp changes or discontinuities—common in tampered regions.

The 2D DCT of a block $f(x, y)$ of size $N \times N$ is defined as:

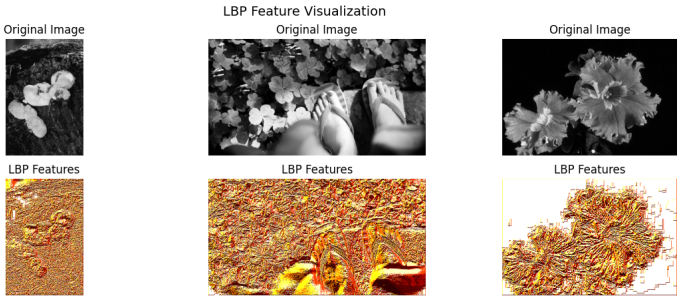

Fig. 4. Visualization of extracted LBP features from Cr and Cb channels

$$F(u,v) = \frac{1}{4}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right) \quad (2)$$

where $u, v = 0, 1, ..., N-1$. The resulting coefficients $F(u,v)$ describe the frequency content in horizontal and vertical directions. In our case, the DCT is applied to the LBP map, enhancing the representation of fine-grained texture changes.

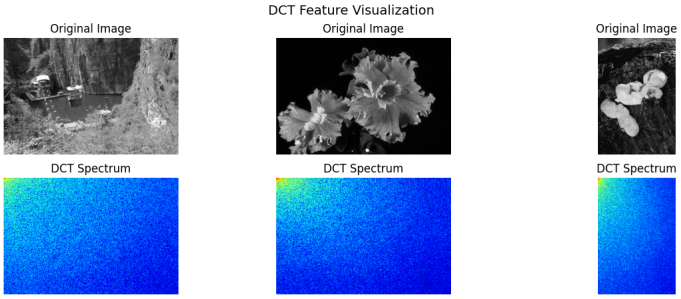

Fig. 5. Visualization of DCT-transformed LBP features representing frequency components

Following DCT, we compute the standard deviation across all blocks for each channel (Cr and Cb) to summarize the variability in texture-frequency response. Finally, we flatten and concatenate the two vectors into a single feature vector for each image. This vector becomes the input for the classification model.

This hybrid extraction process leverages both local texture irregularities (via LBP) and frequency distortions (via DCT), making it well-suited for detecting subtle signs of forgery.

D. Classification Model

In this work, we utilize a *Linear Support Vector Machine (SVM)* classifier to discriminate between real and tampered images based on the extracted LBP-DCT features. The choice of SVM is motivated by its ability to handle high-dimensional data efficiently while providing a robust decision boundary with solid theoretical foundations. The extracted features, derived from chrominance-based LBP and DCT transformations, form a fixed-length vector for each image, which serves as input to the SVM classifier.

The objective of a linear SVM is to find the optimal hyperplane that separates the data into two classes—in our case, *Fake* and *Real* images—with the maximum margin. Mathematically, this is represented by the decision function:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b \quad (3)$$

Here, $\mathbf{x} \in \mathbb{R}^n$ is the input feature vector, $\mathbf{w} \in \mathbb{R}^n$ is the weight vector, and $b \in \mathbb{R}$ is the bias term. The classifier predicts the label of an input sample $\mathbf{x}$ based on the sign of $f(\mathbf{x})$: a positive value indicates one class (e.g., *Real*), while a negative value indicates the other (e.g., *Fake*).

To find the optimal hyperplane, the SVM solves the following optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \ \forall i \quad (4)$$

where $(\mathbf{x}_i, y_i)$ are the training samples and their corresponding class labels ($y_i \in \{-1, +1\}$). This formulation ensures that the margin between the two classes is maximized while correctly classifying all training samples.

In cases where perfect linear separability is not possible, a *soft-margin* SVM introduces slack variables ξ_i and a regularization parameter C to allow some misclassifications:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \quad \text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0 \quad (5)$$

In our implementation, we use the `LinearSVC` class from Scikit-learn, which solves the above optimization using coordinate descent. The regularization parameter C is selected through empirical testing to balance margin maximization and misclassification penalties.

Example: Consider a simplified 2D example where the feature vector $\mathbf{x} = [3, 4]$, the weight vector learned by the model is $\mathbf{w} = [0.5, -0.4]$, and the bias $b = 1.2$. Then, the decision function becomes:

$$\begin{aligned} f(\mathbf{x}) &= (0.5)(3) + (-0.4)(4) + 1.2 \\ &= 1.5 - 1.6 + 1.2 = 1.1 \end{aligned}$$

Since $f(\mathbf{x}) > 0$, this sample is classified as belonging to the *Real* class. If $f(\mathbf{x})$ had been negative, it would be labeled as *Fake*.

Before training the SVM, we normalize the feature vectors using z-score standardization, ensuring that each feature has zero mean and unit variance. This scaling is crucial for SVMs as it ensures that all features contribute equally to the decision boundary.

Furthermore, we adopt 5-fold stratified cross-validation to evaluate the classifier's generalizability. This process splits the dataset into five subsets while preserving the proportion of real and fake samples in each fold, ensuring a fair evaluation.

E. Explainability with LIME

While traditional machine learning models such as Support Vector Machines (SVMs) are powerful for classification tasks, they are often considered "black boxes" due to the lack of transparency in their decision-making process. To address this, we incorporate **LIME (Local Interpretable Model-agnostic Explanations)** into our system to make the forgery detection process more interpretable and trustworthy.

LIME works by approximating the global model locally around the specific data point being predicted. This is achieved by generating perturbed versions of the input (in our case, the LBP-DCT-based feature vector) and observing the impact of these perturbations on the classifier's output. Essentially, LIME fits a simple, interpretable model (such as a sparse linear regression) in the local vicinity of the sample of interest. This local model mimics the behavior of the complex SVM classifier within that localized region, allowing us to understand which features most strongly influenced the final decision.

To understand how the underlying classifier behaves, consider the **decision function of a linear SVM**, which is defined as:

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b \tag{6}$$

Here, $\mathbf{x} \in \mathbb{R}^n$ is the input feature vector (in our case, the extracted features from the image), $\mathbf{w} \in \mathbb{R}^n$ is the weight vector learned by the SVM, and $b \in \mathbb{R}$ is the bias term. The sign of $f(\mathbf{x})$ determines the class label (e.g., Fake or Real), and the magnitude represents the confidence of the decision.

As an example, suppose a sample image results in a 4-dimensional feature vector:

$$\mathbf{x} = [0.8, -1.2, 0.5, 2.0]$$

and assume the learned SVM weight vector and bias are:

$$\mathbf{w} = [1.1, -0.9, 0.4, 0.7], \quad b = -0.5$$

Then the decision function would be:

$$\begin{aligned} f(\mathbf{x}) &= (1.1)(0.8) + (-0.9)(-1.2) + (0.4)(0.5) + (0.7)(2.0) - 0.5 \\ &= 0.88 + 1.08 + 0.2 + 1.4 - 0.5 = 3.06 \end{aligned}$$

Since $f(\mathbf{x}) > 0$, the sample is classified as "Real" with high confidence.

When LIME is applied, it perturbs this input feature vector by randomly modifying values and evaluates the impact on the prediction. By doing this for many such variations, LIME builds a dataset of perturbed instances and corresponding SVM outputs. It then fits a **local surrogate model**, such as a linear regression:

$$g(\mathbf{z}) = \boldsymbol{\theta}^T \mathbf{z} \tag{7}$$

where $\mathbf{z}$ is a perturbed version of the input and $\boldsymbol{\theta}$ represents the importance of each feature locally. The coefficients $\boldsymbol{\theta}$ indicate the strength and direction of influence of each feature for that specific prediction.

In our project, we use this explanation to generate a visual bar plot where top contributing features are highlighted. Positive contributions (favoring "Real") are shown in green, and negative ones (favoring "Fake") in red. This not only helps in auditing the model's decision but also builds confidence in the system's robustness, which is especially important in forensic applications where transparency is critical.

V. Experimental Results

The evaluation of the proposed LIME-SVM-based framework was conducted on two fronts: first using the CASIA 2.0 dataset and then with a combined dataset comprising samples from multiple sources. Evaluation metrics such as precision, recall, F1-score, and AUC were used, along with model explainability through LIME.

A. Evaluation on CASIA 2.0 Dataset

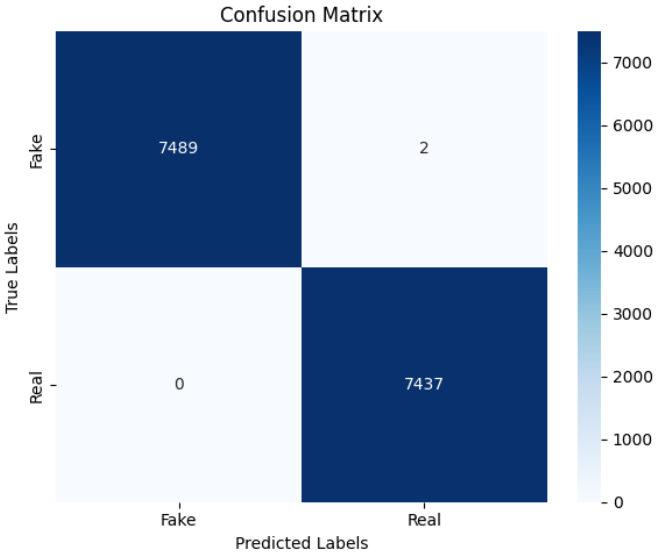

Fig. 6. Confusion Matrix for model trained on CASIA 2.0 dataset.

Fig. 6 shows the confusion matrix for the CASIA 2.0 dataset, where the classifier predicted 7489 fake images and 7437 real images correctly, with only 2 misclassifications.

Classification Report:

	precision	recall	f1-score	support
Fake	1.00	1.00	1.00	7491
Real	1.00	1.00	1.00	7437
accuracy			1.00	14928
macro avg	1.00	1.00	1.00	14928
weighted avg	1.00	1.00	1.00	14928

Fig. 7. Classification Report on CASIA 2.0 dataset.

The classification report in Fig. 7 validates this by showing a precision, recall, and F1-score of 1.00 for both classes, indicating perfect performance on this dataset.

B. Evaluation on Combined Dataset

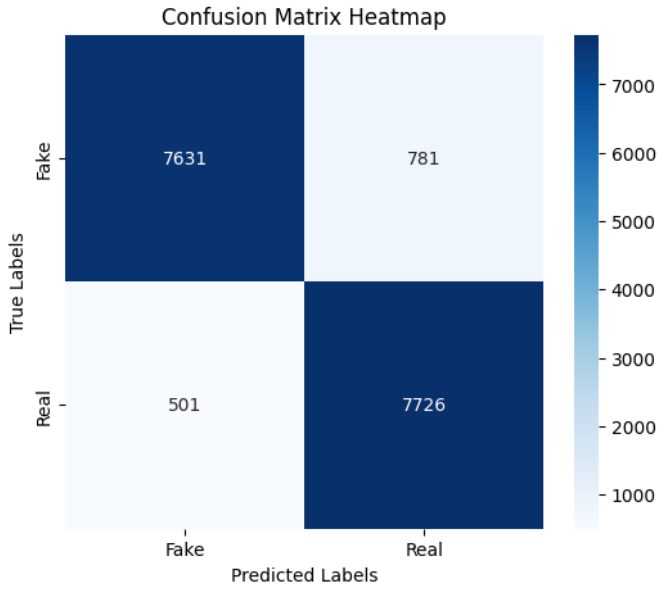

Fig. 8. Confusion Matrix for model trained on combined dataset.

As shown in Fig. 8, the classifier maintained high accuracy on the combined dataset. It correctly identified 7631 fake and 7726 real images, though it misclassified 781 fake and 501 real images.

Classification Report on Combined Dataset:

	precision	recall	f1-score	support
Fake	0.94	0.91	0.92	8412
Real	0.91	0.94	0.92	8227
accuracy			0.92	16639
macro avg	0.92	0.92	0.92	16639
weighted avg	0.92	0.92	0.92	16639

Fig. 9. Classification Report on Combined Dataset.

The classification report in Fig. 9 shows a precision and recall of approximately 0.92 for both classes, demonstrating robust performance under more diverse conditions.

C. Cross-Validation Metrics

In Fig. 10, the precision-recall curve demonstrates an AUC of 0.94, indicating high precision and recall even when the dataset may be slightly imbalanced.

Fig. 11 shows the ROC curve with an AUC of 1.00, reflecting excellent discriminatory power of the classifier between the fake and real classes.

D. Model Explainability Using LIME

In Fig. 12, LIME provides an interpretability layer by identifying and visualizing the most influential features for a prediction. Green bars represent features positively contributing to the classification decision, while red bars signify negative influence. For instance, the feature `feature_1313 > 0.51` had a strong positive impact on classifying the sample as fake, while features such as `feature_953 in (-0.61, 0.02)` had the opposite effect. This level of interpretability is crucial for understanding model behavior in critical forensics applications.

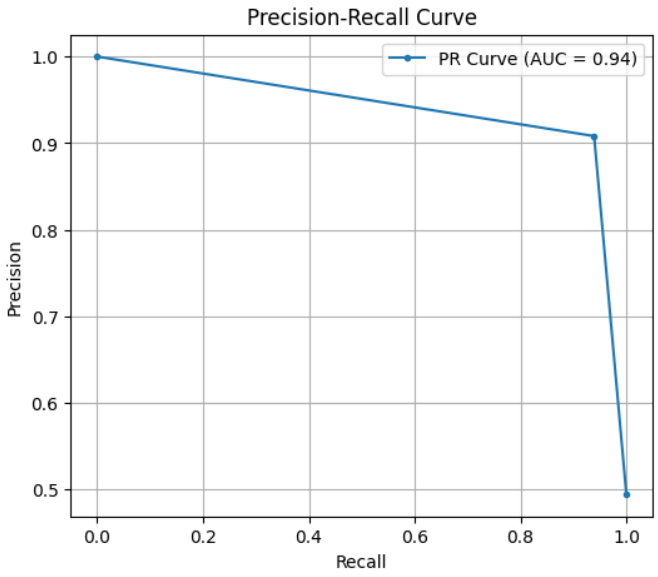

Fig. 10. Precision-Recall Curve with AUC = 0.94.

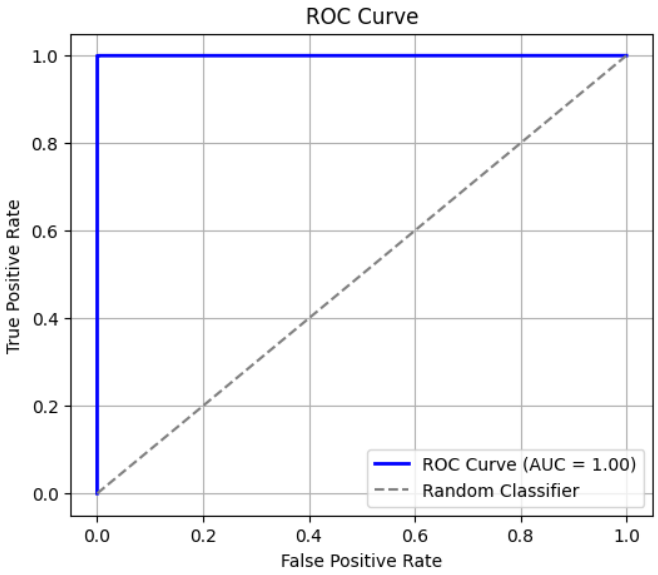

Fig. 11. ROC Curve with AUC = 1.00.

E. Summary of Results

The results show that the model achieves perfect classification on the CASIA 2.0 dataset and maintains strong generalization capabilities on a more complex combined dataset, achieving an overall accuracy of **92%**. The ROC and precision-recall curves indicate reliable classification behavior, and the LIME analysis provides transparency into the model's

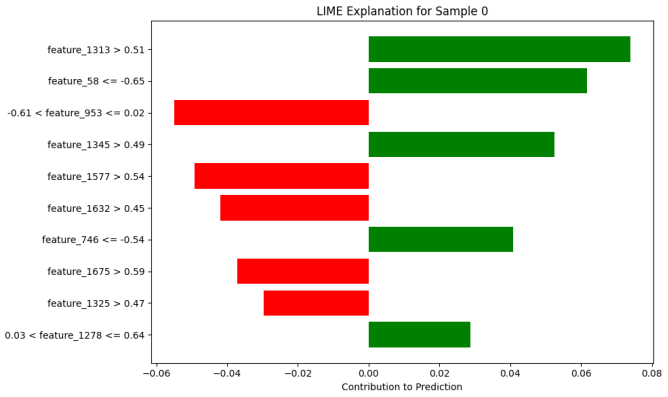

Fig. 12. LIME Explanation showing local feature importance for a test sample.

decision-making process, reinforcing trust and reliability in practical applications.

VI. Future Scope

The current work lays the foundation for explainable image forgery detection by combining handcrafted feature extraction with interpretable machine learning. In the future, the methodology can be extended by integrating deep learning-based feature extraction with LIME or SHAP explainers to further enhance performance and explainability. Additionally, exploring multi-modal datasets and real-world manipulated media from social platforms can improve model robustness. Another potential direction is to develop lightweight versions of the model suitable for deployment on edge devices and mobile platforms, thereby enabling real-time forgery detection in practical scenarios.

VII. Conclusion

In this study, we presented an effective and explainable approach to image forgery detection using chromatic features, LBP-DCT based feature extraction, and a Linear SVM classifier. The system demonstrated strong performance with an accuracy of **92%** on a combined dataset from CASIA 1.0 and CASIA 2.0. By integrating LIME for explainability, we enhanced transparency and interpretability of the model's predictions, which is crucial for real-world forensic applications. This combination of performance and explainability positions our approach as a robust solution for practical deployment in digital media authentication.

References

[1] A. A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, and G. Bebis, "Splicing Image Forgery Detection Based on DCT and Local Binary Pattern," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Austin, TX, USA, 2013, pp. 253–256.

[2] H. Benhamza, A. Djeffal, and A. Cheddad, "Image forgery detection review," in *2021 Int. Conf. Inf. Syst. Adv. Technol. (ICISAT)*, Tebessa, Algeria, 2021, pp. 1–7, doi: 10.1109/ICISAT54145.2021.9678207.

[3] S. D. V and S. D. S, "Detection of Image Forgery Using LBP and DCT Techniques," *Int. J. Sci. Res. Sci. Technol.*, vol. 2, no. 8, pp. 76–81, 2016.

[4] S. Singh and R. Kumar, "Image forgery detection: comprehensive review of digital forensics approaches," *J. Comput. Soc. Sc.*, vol. 7, pp. 877–915, 2024. doi: 10.1007/s42001-024-00265-8.

[5] F. Z. Mehrjardi, A. M. Latif, M. S. Zarchi, and R. Sheikhpour, "A survey on deep learning-based image forgery detection," *Pattern Recognit.*, vol. 144, p. 109778, Dec. 2023. [Online]. Available: https://doi.org/10.1016/j.patcog.2023.109778

[6] S. Tyagi and D. Yadav, "A detailed analysis of image and video forgery detection techniques," *Vis. Comput.*, vol. 39, pp. 813–833, 2023. doi: 10.1007/s00371-021-02347-4.

[7] A. H. Saber, M. A. Khan, and B. G. Mejbel, "A Survey on Image Forgery Detection Using Different Forensic Approaches," *Adv. Sci. Technol. Eng. Syst. J.*, vol. 5, no. 3, pp. 361–370, 2020. doi: 10.25046/aj050347.

[8] S. S. Gaikwad and P. D. Porey, "Copy-Move Forgery Detection Using Efficient Matching and Robust Features," *Int. J. Comput. Appl.*, vol. 178, no. 30, pp. 1–5, Jun. 2019.

[9] S. Walia, K. Kumar, S. Agarwal, and H. Kim, "Using XAI for Deep Learning-Based Image Manipulation Detection with Shapley Additive Explanation," *Symmetry*, vol. 14, no. 8, p. 1611, Aug. 2022. doi: 10.3390/sym14081611.

[10] A. Shiyam and G. Poravi, "Deepfake Low Resource Image Detection with Explainable Reporting," in *Proc. Conf.*, Sep. 2023.

[11] W. H. Abir *et al.*, "Detecting Deepfake Images Using Deep Learning Techniques and Explainable AI Methods," *Intell. Automat. Soft Comput.*, vol. 35, no. 2, pp. 2151–2169, 2023. https://doi.org/10.32604/iasc.2023.029653

[12] V. A. Bhooshan and S. Ghosh, "Robust copy–move image forgery detection using local image permutation interval descriptor and fuzzy clustering," *Expert Syst. Appl.*, vol. 184, p. 115534, 2021. doi: 10.1016/j.eswa.2021.115534.

[13] R. D. B. Filho and M. C. Silveira, "Image forgery detection in lossy compressed images using error level analysis and automatic wavelet soft-thresholding," *Expert Syst. Appl.*, vol. 184, p. 115553, 2021. doi: 10.1016/j.eswa.2021.115553.

[14] Z. Khalid, F. Iqbal, and B. C. M. Fung, "Towards a unified XAI-based framework for digital forensic investigations," *Forensic Sci. Int.: Digital Invest.*, vol. 50, p. 301806, 2024. Available: https://doi.org/10.1016/j.fsidi.2024.301806

[15] B. Diallo, T. Urruty, P. Bourdon, and C. Fernandez-Maloigne, "Robust forgery detection for compressed images using CNN supervision," *Forensic Sci. Int.: Reports*, vol. 2, p. 100112, 2020. doi: 10.1016/j.fsir.2020.100112.

[16] E. Kee, M. K. Johnson, and H. Farid, "Digital Image Authentication From JPEG Headers," *IEEE Trans. Inf. Forensics Secur.*, vol. 6, no. 3, pp. 1066–1075, Sept. 2011. doi: 10.1109/TIFS.2011.2128309.

[17] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, Feb. 2005. doi: 10.1109/TSP.2004.839932.

[18] C.-Y. Lin and S.-F. Chang, "Robust image authentication method surviving JPEG lossy compression," in *Proc. SPIE 3312, Storage and Retrieval for Image and Video Databases VI*, pp. 296–307, 1997. doi: 10.1117/12.298462.

[19] G.-S. Lin and M.-K. Chang, "A passive scheme for tampering detection based on quantization table estimation," in *Proc. SPIE 7744, Visual Commun. Image Process.*, vol. 7744, pp. 774433, 2010. doi: 10.1117/12.863502.

[20] H. Farid and S. Lyu, "Higher-order Wavelet Statistics and their Application to Digital Forensics," in *CVPR Workshop*, Madison, WI, USA, 2003, pp. 94–94. doi: 10.1109/CVPRW.2003.10093.