Enhancing the Image Forgery Detection based Machine Learning Approach using Multiple Datasets

Heba Adnan Raheem
Computer Science Department, College of Computer Science and Information Technology, University of Kerbala, Iraq
hiba.adnan@uokerbala.edu.iq (corresponding author)

Mohammed Abdallazez Mohammed
Computer Science Department, College of Computer Science and Information Technology, University of Kerbala, Iraq
mohammed.abdallazez@uokerbala.edu.iq

Ameer Sameer Hamood Mohammed Ali
Presidency of the University of Babylon, University of Babylon TOEFL Center, Babylon, Iraq
pre225.ameer.sameir@uobabylon.edu.iq

Received: 7 January 2025 | Revised: 31 January 2025 and 1 March 2025 | Accepted: 6 March 2025

 DOI: https://doi.org/10.48084/etasr.10151

ABSTRACT

Nowadays, detecting forged images has become increasingly important because of the widespread use of advanced image editing tools. Splicing is one common form of forgery, where parts or different images are combined to create misleading images. However, detecting this type of forgery poses a challenge because it often appears highly realistic and is difficult to distinguish from authentic images. This study presents a method for detecting forged images. The proposed system aims to enhance forgery detection by carefully analyzing images using preprocessing, such as resizing, converting colors to HSV, analyzing histograms, converting images into binary numeric values, and visualizing the original and forged images and their respective hues based on grayscale, RGB, and HSV histograms. The proposed method used three machine learning algorithms, namely Multilayer Perceptron (MLP), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), and the results demonstrate its effectiveness in rapidly discerning forged images while maintaining high accuracy of 99.72% on the MISD, 99.53 % on the CASIA2, 97.44 % on the NC2016, and 94.30 % on the CoMoFoD datasets.

Keywords-image forgery; KNN; MLP; preprocessing; SVM

I. INTRODUCTION

The impact of an image is stronger than millions of words, as images are used as evidence in celebrity magazines, political campaigns, scientific research, and courts since they represent an efficient and natural method for people to communicate [1, 2]. For example, the translation of images between languages is unnecessary, as ease of use, rapid availability, and the abundance of low-cost devices contribute to the process of capturing, storing, and sending images [3]. At the same time, the rapid accessibility of software packages used to edit and modify images makes it very easy even for users with limited experience to change or create new images [4]. This increases the possibility of forgery and manipulation of images, which is not limited to experts and specialists. Therefore, the weight and integrity of digital images are weakened by the progress of digital technology [5, 6]. Image forgery detection is a critical field aimed at identifying manipulated or altered images. With the rise of sophisticated editing tools, detecting image forgeries has become increasingly challenging [7, 8]. Various techniques, including digital watermarking, statistical analysis, and machine learning, are employed to analyze images for signs of manipulation [9, 10]. The main objective of this study is to develop an algorithm capable of detecting sophisticated image manipulations.

Previous studies have highlighted various methods for image splicing detection, including the utilization of geometry invariants and camera characteristics, as well as deep learning methods that leverage Convolutional Neural Networks (CNNs) and autoencoders. However, these methods often rely on datasets that do not adequately represent the complexities

associated with multiple splicing. The MISD not only includes a diverse array of spliced images but also offers ground-truth masks, enhancing the accuracy of detection algorithms by providing clear references for identifying spliced regions.

In [11], an MWC-Net multitask wavelet correction network was proposed to learn more comprehensive and representative features to detect link forgery and distinguish its location based on wavelet collection, decompression, and reconstruction of the features of forgery images. MWC-Net was based on a multitasking strategy that contributed to more comprehensive learning to improve image forgery detection. The results showed that this system outperformed other techniques based on the same image forgery features. In [12], a technique was proposed to detect image forgery according to the color distribution of pixels near the edge of the images, by extracting them using the contour transformation technique to accurately distinguish the original and manipulated edges based on the IQR quartile range criteria. The chromatic histograms and borders were distributed in the YCbCr color space. This method was used to reduce the computation time and enhance the localization performance. The results showed the superiority of this method in the accurate detection of image forgery by about 97% with 100%.

In [13] a modified CASIA dataset was used along with others because image splicing detection did not involve multiple-spliced images. This study provided a high-quality image dataset available to the public. In [14], a CNN model was proposed to detect forged images with high accuracy in real-time based on a small number of variables. The lightweight model was based on a set of convolutional layers that are suitable for working in a limited-resource environment. The results proved the sensitivity of the algorithm on the CASIA 1.0 and 2.0 databases with an accuracy of 99.1% and 99.3%, and forgery detection in the CUISDE database with an accuracy of 100%. In [15], a DCU-Net-channel UNIT network was proposed to detect image forgery and manipulation. High-frequency filters were utilized to extract the remnants of the forged and manipulated images. Then, the manipulated and original images were used, and the deep features extracted from the encoding were merged, as well as the features that were tampered with by convolution and dilation using the secondary fusion technique. The results of the proposed system on the Kazeh and Colombia datasets showed that the algorithm performed the best in detecting and resisting noise and recompression attacks on JPEG images.

II. METHODOLOGY

The proposed method implements a splicing forgery detection technique for digital images. It comprises several key steps aimed at enhancing the effectiveness and accuracy of the splicing forgery detection algorithm. Figure 1 illustrates the proposed system. The following steps outline the execution of the proposed image forgery detection algorithm. The aim is to deliver a comprehensive analysis of image forgery detection, utilizing a diverse range of image features and metrics.

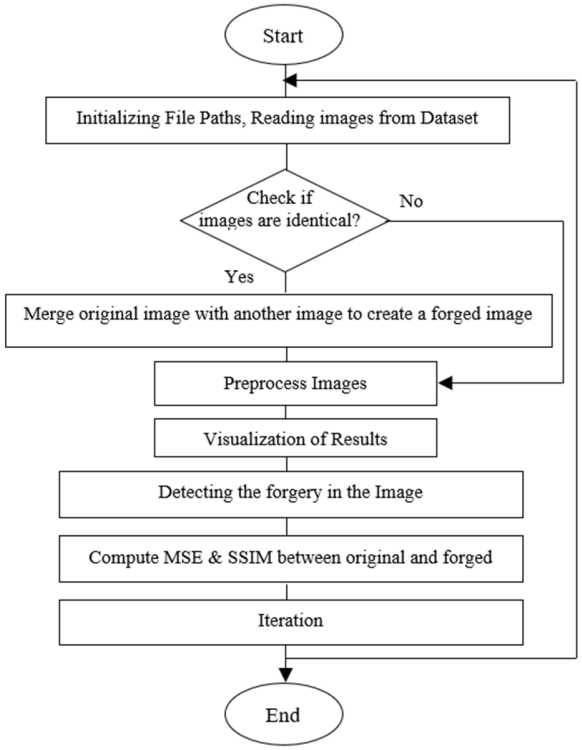

Fig. 1. Flowchart of the proposed image splicing forgery detection algorithm.

A. Define File Paths, Read Images from the Dataset

The initial stage begins by setting the file paths for the original and forged image folders. Then, images from both folders are read to prepare for further analysis. The database is converted from the MISD, CASIA2, NC2016, and CoMoFoD datasets to a CSV file, based on a set of Java libraries to read and process images by converting to grayscale, changing size, extracting pixel values, converting the data to a binary digital format in the form of pixel values, and writing the images so that each image corresponds to one row in the database.

B. Examine Image Identity

Ensure that the number of images in both folders is equal and verify their identity. If the images are identical create a forged image by adding another image to the original one.

C. Image Preprocessing

Convert images to the HSV color space using the rgb2hsv function, calculate hue differences by subtracting the forged image from the original, and compute histograms in grayscale, RGB, and HSV for both original and forged images. In the preprocessing stage, the raw input images undergo several operations to enhance their quality and prepare them for further analysis and processing. These operations include:

- Image resizing: This operation involves changing the dimensions of each image. In forgery detection, resizing ensures that both the original and forged images have the same dimensions, which is necessary for accurate comparison and analysis.

- RGB to HSV color space conversion: Images are represented in RGB (Red, Green, Blue) color space and converted to HSV (Hue, Saturation, Value). H is the type of color (Red, Green, Blue), S is the intensity of color, and V is the brightness of the color.
- Hue component extraction: The hue component represents the color tone of an image. By extracting this component, forgery detection algorithms can focus specifically on color differences between original and forged images, which are often indicative of forgery. Mathematically, hue extraction involves isolating the hue channel from the HSV representation of the image.
- Histogram calculation: A histogram is a graphical representation of the distribution of pixel values in an image. Calculating histograms for grayscale and RGB color spaces helps to analyze the overall brightness, contrast, and color intensity distributions in the images. Histograms in the HSV color space provide insights into the distribution of hue, saturation, and value components, aiding in the detection of color-based forgeries [16].
- Grayscale conversion: Converting images to grayscale simplifies the analysis by removing color information and representing each pixel with a single intensity value. Grayscale images are often used in forgery detection to focus on structural and luminance differences between original and forged images. The grayscale conversion was applied using the following formula: $0.299 \times Red + 0.587 \times Green + 0.114 \times Blue$.
- Normalization is based on measuring pixel values within a specific range, for example, between zero and one, or applying a z-score to improve model performance during training.
- Binary conversion involved specific libraries in the Java programming language to implement functions such as threshold, to convert grayscale images directly to binary images.

D. Visualization of Results

The original and forged images were visualized alongside their respective hue difference image, and grayscale, RGB, and HSV histograms.

E. Calculate MSE and SSIM Between Original and Forged Images

The Mean Squared Error (MSE) and Structural Similarity Index (SSIM) are used to quantitatively measure the difference between the original and forged images. In evaluating the effectiveness of forgery detection techniques, it is essential to employ robust evaluation measures that provide insight into the quality and similarity of images.

MSE is a commonly used metric to quantify the difference between two images, calculating the average squared difference between the corresponding pixel intensities in the original and forged images. Mathematically, MSE is defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N} \left(I_{Original}(i) - I_{Forged}(i)\right)^2 \quad (1)$$

where $I_{Original}(i)$ and $I_{Forged}(i)$ represent the pixel intensities of the original and forged images, respectively, and NN is the total number of pixels [17]. MSE values range from 0 to ∞, so a lower MSE value indicates a smaller difference between the original and forged images, implying higher similarity [18].

SSIM is a perception-based metric that measures the similarity between two images, taking into account luminance, contrast, and structure. Unlike MSE, SSIM considers the human visual system's characteristics. SSIM is calculated by comparing three components: luminance (ll), contrast (cc), and structure (ss). The overall SSIM index is calculated as the product of these components [19]:

$$SSIM(X,Y) = \frac{2\,\mu_X*\mu_Y+C_1}{\mu_X^2+\mu_Y^2+C_1} \times \frac{2\,\sigma_{XY}+C_2}{\sigma_X^2+\sigma_Y^2+C_2} \times \frac{2\,\sigma_{XY}+C_3}{\sigma_X+\sigma_Y+C_3} \quad (2)$$

where μx, μy, σx, σy, and σxy are the means and covariances of x and y, and $C1$, $C2$, and $C3$ are constants to stabilize the division with weak denominators. SSIM values range from -1 to 1, with 1 indicating identical images [20].

F. Iteration

The above steps are repeated for each pair of original and forged images in the dataset, ensuring thorough analysis and detection of image forgeries. In addition, the proposed image forgery detection method trains three machine learning algorithms to detect forged images from the original images in the dataset [21]:

- The Multilayer Perceptron (MLP) is a neural network that can be used effectively to detect image forgery [22]. MLP can learn complex patterns in the data, making them suitable for distinguishing between original and adjacent images. It is based on a hidden layer, an input layer, and an output layer, allowing it to learn complex features from the input data and is trained according to the error between the expected and actual outputs. This process is repeated to contribute to improving the accuracy of detecting forged images over time.
- The Support Vector Machine (SVM) is used to detect image forgery by finding ideal super levels to separate classes, making it particularly effective in distinguishing between original and forged images. It relies on preprocessing to convert images to grayscale to simplify the data and reduce computational complexity by applying noise removal techniques such as median filtering to improve image quality. The support vector mechanism extracts relevant features to find classifications by analyzing pixels.
- The K-Nearest Neighbors (KNN) is used to detect image forgery based on the principle of determining a point class for data based on the classes of its nearest neighbors in the feature space [23]. It includes changing the size, normalization, and converting images to grayscale to extract relevant features to determine similarity based on several techniques, including color schemes that contribute to analyzing the distribution of colors in the image, and texture features, such as the co-occurrence matrix, and

detecting edges within the image to highlight the important features. The proposed system indicates the possibility of KNN's high accuracy rates in detecting forged images, which indicates its effectiveness in distinguishing between forged and original images.

III. DATASETS

The validity of digital forgery detection techniques depends on the standard data. Previous research trends on detecting image forgery were based on a standard dataset to link images. The proposed system is based on a dataset collection dedicated to this purpose.

A. Multiple Image Splicing Dataset (MISD)

The database contains a different set of images from CASIA V1.0 [24], which were merged into a single dataset containing a set of realistic images in natural colors. It consists of 618 original images in JPG format and 300 multi-link images in JPG format with dimensions of 384×256 pixels [13].

B. CASIA2 Dataset

CASIA 2 is an image tampering detection evaluation database. It is designed to evaluate AI algorithms in detecting splicing, copying, moving, and other types of tampering [25]. It contains two types of images, original and fake, altered with manipulation techniques such as copying, moving, and others [26].

C. NC2016 Dataset

NC2016 is a collection of forensic images [27], developed for a collection of original and manipulated images, such as parts of images that have been combined using copy and move, removing some elements, or changing their visual appearance [28].

D. Copy-Move Forgery Detection (CoMoFoD) Dataset

CoMOFoD is a specialized dataset of original and forged images, with different types of manipulations to simulate the forgery of realistic copies [29]. It includes rotation, gradation, reflection, brightness, contrast, and blurring, which makes it difficult to detect image forgery [30].

Table I shows the used datasets and their attributes after the preprocessing stage.

TABLE I. THE USED FEATURES OF DATASETS

Dataset	Number of records	Number of attributes
MISD	918	5
CASIA2	4795	7
NC2016	13000	12
CoMoFoD	10400	5

IV. RESULTS

MATLAB was used to implement the forgery detection technique through the following steps on.

A. Forged Image by Addition

If images are identical, create a forged image by merging the original and another image as shown in Figure 2.

Fig. 2. Add image (panda) to an original image.

B. Hue Difference Image

The hue difference image is a visualization that highlights areas where the hue (color) of the forged image differs significantly from that of the original image. It is calculated by extracting the hue component from both the original and forged images and then calculating the absolute difference between them. Areas of significant hue difference indicate potential regions of manipulation or forgery, such as color alterations or additions, as shown in Figure 3.

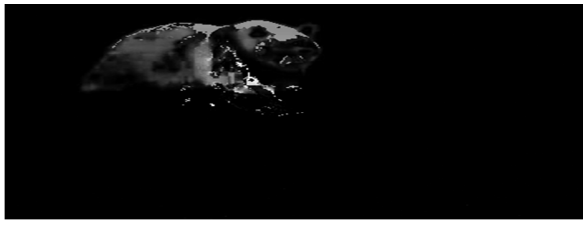

Fig. 3. Hue difference between the original and forged images.

C. Grayscale Histogram

Grayscale histograms provide insights into the distribution of pixel intensities in the images, representing their overall brightness and contrast. Histograms for both the original and forged images are plotted, showing the frequency of pixel intensity values. Differences in the histogram shapes or peaks may indicate areas where modifications have been made, affecting the brightness or contrast of the images, as shown in Figure 4. The x-axis of a grayscale histogram represents the intensity values of the pixels, typically ranging from 0 (black) to 255 (white) for 8-bit images. The y-axis shows the frequency or number of pixels with each intensity value.

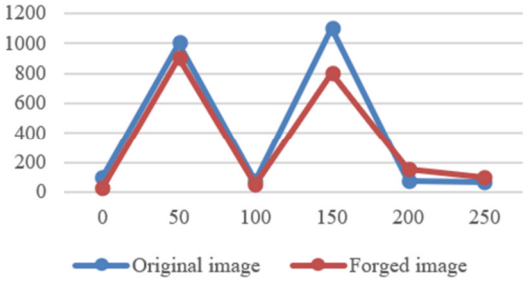

Fig. 4. Grayscale histogram for original and forged images.

D. RGB Histogram

RGB histograms are generated separately for each color channel (red, green, blue) in both the original and forged images. These histograms illustrate the distribution of color

intensities across different color channels, revealing potential alterations in specific color components. Variations in histogram patterns or shifts in color distribution may indicate manipulations or modifications in certain color channels.

E. HSV Histogram

The HSV histograms depict the distribution of the hue, saturation, and value components in the images. Similar to RGB histograms, HSV histograms provide insight into the color characteristics of the images but in a different color space. Changes in the distribution of hue or saturation values may signal alterations in color tones or saturation levels, suggesting a possible forgery. The x-axis represents the values of the channels in the HSV color space. Hue ranges from 0 to 179 in OpenCV, saturation (S) ranges from 0 to 255, and value (V) ranges from 0 to 255.

F. Difference Image and Suspicious Regions

The difference image is computed by calculating the absolute difference between the original and forged images. Thresholding the grayscale difference image highlights areas where significant differences exist between the two images. These suspicious regions indicate potential regions of interest where image manipulation or forgery may have occurred.

G. Results

Tables II-VII and Figure 5 show the performance evaluation results of different machine learning algorithms in detecting forged images across multiple databases. The results show significant differences in accuracy and processing time, as the MLP algorithm achieved accuracy rates of 99.6% on MISD with a processing time of 427 ms. On the contrary, the SVM algorithm showed similar accuracy on MISD but with a significantly faster processing time of 56 ms. The KNN algorithm outperformed the others, achieving the highest accuracy of 99.72% on MISD, with the lowest processing time of only 4 ms. KNN outperformed MLP and SVM algorithms due to its simplicity, ease of implementation, speed of building the trained model, and classifying the dataset based on the majority class among the nearest neighbors without the need for complex mathematical formulations.

The results also show that the balance of precision is the highest for KNN, with a value of 0.9950, highlighting its ability to effectively reduce false positives compared to the other algorithms. All models achieved a perfect recall of one, which indicates that they were able to identify all positive cases without misses. In addition, KNN maintained balance with the Kappa rate in superior consistency of correct detections for both fake and real images on different datasets. The detection results of KNN confirm its effectiveness in identifying fake images. In terms of false alert rate, the best tool was the KNN algorithm with a rate of 0.8333, which highlights its reliability in reducing false alerts. Regarding the area under the Euclidean curve, the MLP algorithm obtained the highest score at 0.0205, while KNN and SVM obtained lower values, indicating a potential need to improve discrimination between classes in some cases. Regarding MSE, KNN showed the best value of 0.0035, as a result of its greater prediction accuracy, indicating its efficiency.

TABLE II. EVALUATION METRICS ON MISD

Evaluation parameters	MISD		
	MLP	**SVM**	**KNN**
Precision	0.9964	0.9963	0.99745
Recall	1	1	1
F-Measure	1	0.9982	0.99872
Kappa Coefficient	0.9854	0.9854	0.9893
Detection Rate (DR)	1	1	1
False Alert Rate (FAR)	0.8952	0.7802	0.71831
Area Under Curve (AUC)	0.025	0.025	0.0285
Mean Absolute Error	0.0132	0.2503	0.00610
Relative Absolute Error (RAE)	8.9706	169.34	4.0201
Root Relative Squared Error (RRSE)	18.150	123.68	17.4809
Error Rate	0.0036	0.0036	0.00272

TABLE III. EVALUATION METRICS OF CASIA2 DATASET

Evaluation parameters	CASIA2		
	MLP	**SVM**	**KNN**
Precision	0.99448	0.99444	0.99502
Recall	1	1	1
F-Measure	0.99723	0.99721	0.99750
Kappa Coefficient	0.9885	0.98859	0.99023
Detection Rate (DR)	1	1	1
False Alert Rate (FAR)	0.84732	0.84825	0.83333
Area Under Curve (AUC)	0.0205	0.0117	0.0130
Mean Absolute Error	0.00512	0.2402	0.00353
Relative Absolute Error (RAE)	2.56043	119.941	1.75444
Root Relative Squared Error (RRSE)	14.8986	100.472	13.7329
Error Rate	0.00550	0.00556	0.00469

TABLE IV. EVALUATION METRICS ON NC2016

Evaluation Parameters	NC2016		
	MLP	**SVM**	**KNN**
Precision	0.8666	1.0	1.0
Recall	0.097	0.0977	0.125
F-Measure	0.1756	0.1780	0.222
Kappa Coefficient	0.1699	0.1731	0.217
Detection Rate (DR)	1	1	1
False Alert Rate (FAR)	0.8	0.902	0.875
Area Under Curve (AUC)	0.2229	0.1600	0.321
Mean Absolute Error	0.0338	0.0275	0.022
Relative Absolute Error (RAE)	117.74	0.122	82.47
Root Relative Squared Error (RRSE)	94.323	94.914	90.02
Error Rate	0.0312	0.0307	0.025

TABLE V. EVALUATION METRICS ON COMOFOD

Evaluation parameters	CoMoFoD		
	MLP	**SVM**	**KNN**
Precision	0.942	0.9331	0.9432
Recall	1	1	1
F-Measure	0.970	0.9654	0.9708
Kappa Coefficient	0	0	0.0004
Detection Rate (DR)	1	1	1
False Alert Rate (FAR)	0.857	1	1
Area Under Curve (AUC)	0.071	0.0375	0.0477
Mean Absolute Error	0.030	0.2461	0.0301
Relative Absolute Error (RAE)	91.68	656.10	90.850
Root Relative Squared Error (RRSE)	99.20	198.90	102.37
Error Rate	0.056	0.0692	0.0569

TABLE VI. ACCURACY AND TIME TO BUILD THE CLASSIFICATION MODEL

Algorithm	MISD		CASIA2		NC2016		CoMoFoD	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
MLP	99.6363 %	427 ms	99.4436 %	2215 ms	96.8717 %	5987 ms	94.3269 %	38 ms
SVM	99.6364 %	56 ms	99.4437 %	131 ms	96.9231 %	1015 ms	93.0769 %	292 ms
KNN	99.7275 %	4 ms	99.5308 %	6 ms	97.4423 %	11 ms	94.3029 %	11 ms

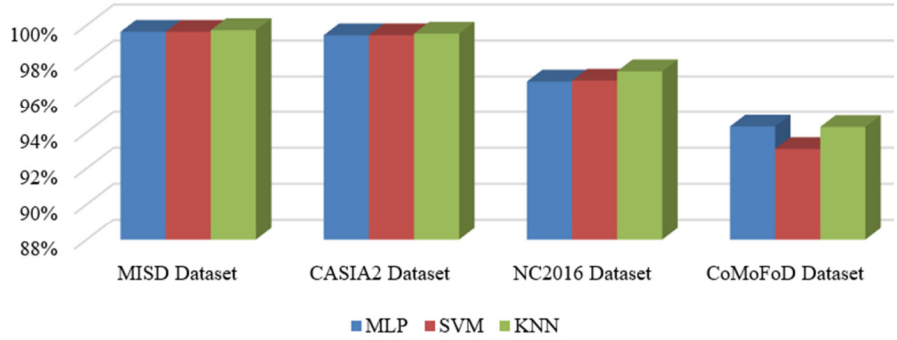

Fig. 5. Accuracy comparison of the proposed algorithms on different datasets.

TABLE VII. COMPARISON WITH RELATED WORKS

Ref. - Year	Dataset	Tool used	Algorithm	Best accuracy
[31] - 2020	CASIA 2	Multimedia Tools Apps	Forgery clustering algorithm	96.9%
[32] - 2022	Mix of CASIA- 2.0 and NC2016	Multimedia Tools and Applications	CNN	72.9%
[33] - 2021	CASIA-V2	Multimedia Tools and Applications	Deep neural networks (CNN)	93.04%
	CoMoFoD and CASIA-V2			88.90%
	CoMoFoD, CASIA-V2, and NIST 2018			89.01%
Proposed system	MISD	Java and Python	KNN	99.72%
	CASIA2			99.53%
	NC2016			97.44%
	CoMoFoD			94.30%

V. CONCLUSIONS

The detection of forgery in digital images is an extremely important field of research as a result of tremendous development with the advancement of technology. The hue difference image and grayscale histograms reveal noticeable differences between the original and forged images, especially in areas where manipulation or forgery may have occurred. RGB and HSV histograms provided insight into the color distribution and characteristics of the images, aiding in the identification of suspicious regions. The difference in image and calculated metrics (MSE and SSIM) further validate the effectiveness of forgery detection techniques, with lower MSE and higher SSIM values indicating a closer resemblance between the original and forged images. The implementation results demonstrate the efficacy of the forgery detection technique in detecting multiple image splicing. The proposed system contributes to the advancement of the image forensics field and provides valuable insights to detect and identify image forgeries with high accuracy based on the KNN algorithm. The novelty of the proposed system depends on the integration of algorithms to identify subtle discrepancies between original and forged images, providing a more powerful and accurate solution after testing it on diverse datasets (MISD, CASIA2, NC2016, and CoMoFoD). This contributes to its ability to process comprehensively and be compatible with novelties in the nature of images in the future. Some potential directions for future work include exploring deep learning techniques, such as CNNs and RNNs, for automatic feature extraction to enhance detection accuracy, and investigating methods for real-time detection of image forgeries, allowing swift action in applications such as live streaming and video surveillance.

REFERENCES

[1] A. Cepak and T. J. Mesyn, "Fakes, Forgery, and Facebook," in *Handbook of Visual Communication*, 1st ed., S. Josephson, J. D. Kelly, and K. Smith, Eds. Routledge, 2020, pp. 465–480.

[2] C. Nastasi and S. Battiato, "Defamation 2.0: New Threats in Digital Media Era - An Overview on Forensics Approaches in the Social Network Ecosystem:," in *Proceedings of the International Conference on Image Processing and Vision Engineering*, Online Streaming, 2021, pp. 121–127, https://doi.org/10.5220/0010463601210127.

[3] S. S. Ali, I. I. Ganapathi, N. S. Vu, S. D. Ali, N. Saxena, and N. Werghi, "Image Forgery Detection Using Deep Learning by Recompressing Images," *Electronics*, vol. 11, no. 3, Jan. 2022, Art. no. 403, https://doi.org/10.3390/electronics11030403.

[4] N. K. Rathore, N. K. Jain, P. K. Shukla, U. Rawat, and R. Dubey, "Image Forgery Detection Using Singular Value Decomposition with Some Attacks," *National Academy Science Letters*, vol. 44, no. 4, pp. 331–338, Aug. 2021, https://doi.org/10.1007/s40009-020-00998-w.

[5] A. H. Saber, M. A. Khan, and B. G. Mejbel, "A Survey on Image Forgery Detection Using Different Forensic Approaches," *Advances in*

Science, Technology and Engineering Systems Journal, vol. 5, no. 3, pp. 361–370, 2020, https://doi.org/10.25046/aj050347.

[6] A. Kuznetsov, "Digital image forgery detection using deep learning approach," *Journal of Physics: Conference Series*, vol. 1368, no. 3, Aug. 2019, Art. no. 032028, https://doi.org/10.1088/1742-6596/1368/3/032028.

[7] A. Alzahrani, "Digital Image Forensics: An Improved DenseNet Architecture for Forged Image Detection," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13671–13680, Apr. 2024, https://doi.org/10.48084/etasr.7029.

[8] O. Mayer and M. C. Stamm, "Exposing Fake Images With Forensic Similarity Graphs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 1049–1064, Aug. 2020, https://doi.org/10.1109/JSTSP.2020.3001516.

[9] S. Chen, T. Yao, Y. Chen, S. Ding, J. Li, and R. Ji, "Local Relation Learning for Face Forgery Detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, pp. 1081–1088, May 2021, https://doi.org/10.1609/aaai.v35i2.16193.

[10] O. Mayer and M. C. Stamm, "Forensic Similarity for Digital Images," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1331–1346, 2020, https://doi.org/10.1109/TIFS.2019.2924552.

[11] X. Bi, Z. Zhang, Y. Liu, B. Xiao, and W. Li, "Multi-Task Wavelet Corrected Network for Image Splicing Forgery Detection and Localization," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, Shenzhen, China, Jul. 2021, pp. 1–6, https://doi.org/10.1109/ICME51207.2021.9428466.

[12] M. Habibi and H. Hassanpour, "Splicing Image Forgery Detection and Localization Based on Color Edge Inconsistency using Statistical Dispersion Measures," *International Journal of Engineering*, vol. 34, no. 1, Feb. 2021, https://doi.org/10.5829/ije.2021.34.02b.16.

[13] K. D. Kadam, S. Ahirrao, and K. Kotecha, "Multiple Image Splicing Dataset (MISD): A Dataset for Multiple Splicing," *Data*, vol. 6, no. 10, Sep. 2021, Art. no. 102, https://doi.org/10.3390/data6100102.

[14] K. M. Hosny, A. M. Mortda, N. A. Lashin, and M. M. Fouda, "A New Method to Detect Splicing Image Forgery Using Convolutional Neural Network," *Applied Sciences*, vol. 13, no. 3, Jan. 2023, Art. no. 1272, https://doi.org/10.3390/app13031272.

[15] H. Ding, L. Chen, Q. Tao, Z. Fu, L. Dong, and X. Cui, "DCU-Net: a dual-channel U-shaped network for image splicing forgery detection," *Neural Computing and Applications*, vol. 35, no. 7, pp. 5015–5031, Mar. 2023, https://doi.org/10.1007/s00521-021-06329-4.

[16] Monika, D. Bansal, and A. Passi, "Image Forgery Detection and Localization Using Block Based and Key-Point Based Feature Matching Forensic Investigation," *Wireless Personal Communications*, vol. 127, no. 4, pp. 2823–2839, Dec. 2022, https://doi.org/10.1007/s11277-022-09898-2.

[17] G. Zhou, X. Tian, and A. Zhou, "Image copy-move forgery passive detection based on improved PCNN and self-selected sub-images," *Frontiers of Computer Science*, vol. 16, no. 4, Aug. 2022, Art. no. 164705, https://doi.org/10.1007/s11704-021-0450-5.

[18] F. Akdeniz and Y. Becerikli, "Detecting audio copy-move forgery with an artificial neural network," *Signal, Image and Video Processing*, vol. 18, no. 3, pp. 2117–2133, Apr. 2024, https://doi.org/10.1007/s11760-023-02856-w.

[19] M. Uma Devi and U. Ravi Babu, "Grey wolf assisted SIFT for improving copy move image forgery detection," *Evolutionary Intelligence*, vol. 15, no. 2, pp. 1097–1108, Jun. 2022, https://doi.org/10.1007/s12065-019-00304-8.

[20] A. H. Mohammed, D. H. Badr, and F. Ali, "Detection of Image Forgery Using Information Standard Method With SVM," *Journal of Physics: Conference Series*, vol. 1818, no. 1, Mar. 2021, Art. no. 012212, https://doi.org/10.1088/1742-6596/1818/1/012212.

[21] S. Singh and R. Kumar, "Image forgery detection: comprehensive review of digital forensics approaches," *Journal of Computational Social Science*, vol. 7, no. 1, pp. 877–915, Apr. 2024, https://doi.org/10.1007/s42001-024-00265-8.

[22] F. Z. Mehrjardi, A. M. Latif, M. S. Zarchi, and R. Sheikhpour, "A survey on deep learning-based image forgery detection," *Pattern Recognition*, vol. 144, Dec. 2023, Art. no. 109778, https://doi.org/10.1016/j.patcog.2023.109778.

[23] J. Zhang, Y. Li, S. Niu, Z. Cao, and X. Wang, "Improved Fully Convolutional Network for Digital Image Region Forgery Detection," *Computers, Materials & Continua*, vol. 60, no. 1, pp. 287–303, 2019, https://doi.org/10.32604/cmc.2019.05353.

[24] K. D. Kadam, S. Ahirrao, and K. Kotecha, "Multiple Image Splicing Dataset (MISD): A Dataset for Multiple Splicing," *Data*, vol. 6, no. 10, Oct. 2021, Art. no. 102, https://doi.org/10.3390/data6100102.

[25] D. Goel, "CASIA 2.0 Image Tampering Detection Dataset." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/divg07/casia-20-image-tampering-detection-dataset.

[26] J. Dong, W. Wang, and T. Tan, "CASIA Image Tampering Detection Evaluation Database," in *2013 IEEE China Summit and International Conference on Signal and Information Processing*, Beijing, China, Jul. 2013, pp. 422–426, https://doi.org/10.1109/ChinaSIP.2013.6625374.

[27] "Dataset NC2016 - Open Media Forensics Challenge." NIST, [Online]. Available: https://www.nist.gov/itl/iad/mig/open-media-forensics-challenge.

[28] P. Zhuang, H. Li, S. Tan, B. Li, and J. Huang, "Image Tampering Localization Using a Dense Fully Convolutional Network," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2986–2999, 2021, https://doi.org/10.1109/TIFS.2021.3070444.

[29] "CoMoFoD." University of Zagreb, [Online]. Available: https://www.vcl.fer.hr/comofod/.

[30] D. Tralic, P. L. Rosin, X. Sun, and S. Grgic, "Copy-Move Forgery Detection Using Cellular Automata," in *Cellular Automata in Image Processing and Geometry*, vol. 10, P. Rosin, A. Adamatzky, and X. Sun, Eds. Springer International Publishing, 2014, pp. 105–125.

[31] N. T. Pham, J. W. Lee, and C. S. Park, "Structural Correlation Based Method for Image Forgery Classification and Localization," *Applied Sciences*, vol. 10, no. 13, Jun. 2020, Art. no. 4458, https://doi.org/10.3390/app10134458.

[32] D. Mallick, M. Shaikh, A. Gulhane, and T. Maktum, "Copy Move and Splicing Image Forgery Detection using CNN," *ITM Web of Conferences*, vol. 44, 2022, Art. no. 03052, https://doi.org/10.1051/itmconf/20224403052.

[33] S. Jabeen, U. G. Khan, R. Iqbal, M. Mukherjee, and J. Lloret, "A deep multimodal system for provenance filtering with universal forgery detection and localization," *Multimedia Tools and Applications*, vol. 80, no. 11, pp. 17025–17044, May 2021, https://doi.org/10.1007/s11042-020-09623-w.