Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.042755

ARTICLE

Image Splicing Forgery Detection Using Feature-Based of Sonine Functions and Deep Features

Ala'a R. Al-Shamasneh[1] and Rabha W. Ibrahim[2,3,4,*]

[1]Department of Computer Science, College of Computer & Information Sciences, Prince Sultan University, Riyadh, 11586, Saudi Arabia

[2]Department of Computer Science and Mathematics, Lebanese American University, Beirut, 1102 2801, Lebanon

[3]Department of Mathematics, Mathematics Research Center, Near East University, Nicosia, 99138, Turkey

[4]Information and Communication Technology Research Group, Scientific Research Center, Alayen University, Dhi Qar, 64001, Iraq

*Corresponding Author: Rabha W. Ibrahim. Email: rabhaibrahim@yahoo.com

Received: 11 June 2023 Accepted: 13 September 2023 Published: 30 January 2024

ABSTRACT

The growing prevalence of fake images on the Internet and social media makes image integrity verification a crucial research topic. One of the most popular methods for manipulating digital images is image splicing, which involves copying a specific area from one image and pasting it into another. Attempts were made to mitigate the effects of image splicing, which continues to be a significant research challenge. This study proposes a new splicing detection model, combining Sonine functions-derived convex-based features and deep features. Two stages make up the proposed method. The first step entails feature extraction, then classification using the "support vector machine" (SVM) to differentiate authentic and spliced images. The proposed Sonine functions-based feature extraction model reveals the spliced texture details by extracting some clues about the probability of image pixels. The proposed model achieved an accuracy of 98.93% when tested with the CASIA V2.0 dataset "Chinese Academy of Sciences, Institute of Automation" which is a publicly available dataset for forgery classification. The experimental results show that, for image splicing forgery detection, the proposed Sonine functions-derived convex-based features and deep features outperform state-of-the-art techniques in terms of accuracy, precision, and recall. Overall, the obtained detection accuracy attests to the benefit of using the Sonine functions alongside deep feature representations. Finding the regions or locations where image tampering has taken place is limited by the study. Future research will need to look into advanced image analysis techniques that can offer a higher degree of accuracy in identifying and localizing tampering regions.

KEYWORDS

Image forgery; image splicing; deep learning; Sonine functions

1 Introduction

Image forgery refers to the process of altering or manipulating digital images with the intent of deceiving or misleading viewers. Various methods and software tools, such as Photoshop or other photo editing software, can be used for image manipulation [1].

Digital image tampering can take many forms, including adding, removing, or modifying objects, changing colors, or altering the lighting and contrast of an image. Some common examples of digital image tampering include removing or adding objects, changing the background, or retouching images to make them look more attractive [2]. Digital image manipulation can have serious consequences, especially where accuracy and reliability are essential, like in journalism or legal proceedings. As a result, a range of tools and techniques are available to help detect digital image tampering, such as algorithms designed to identify inconsistencies in digital images [3]. In general, image manipulation is divided into two approaches. The blind approach is another name for the passive method because it requires no additional information to detect image forgery. This method is based on features extracted directly from images [4,5]. Copy-move [6,7] is the process of adding new content from the same image, while image splicing involves incorporating new elements from another image into the original. Usually, there are no overt indications that an image has been changed; however, some image statistics may have been altered. The act of combining and merging portions of other images to create a composite fabricated image is known as image splicing, as opposed to copy-move forgery, as shown in Fig. 1. The instantaneous texture change caused by the splicing process provides a numerical basis for distinguishing between authentic and fake images.

Figure 1: Sample of forgery spliced image (CASIA2 dataset)

Several works have been published in the literature to help with forgery detection. The lack of statistical data for feature extraction, especially in forged image areas, is one of these studies' limitations [8]. The main goal of this study is to enhance image splicing detection by combining conventional and deep-based feature extraction.

2 Related Works

This section reviews the studies on methods for splicing forgery detection, which includes a variety of viewpoints and methodologies.

2.1 Traditional Methods of Feature Extraction

The traditional splicing forgery detection approach is a method used to identify image tampering through splicing or combining parts of different images. The traditional splicing forgery detection

approach involves analyzing the image's content and identifying discrepancies in the image's metadata to detect any tampering. The ability to identify fake images makes detecting image splicing a crucial task in digital forensics. The traditional splicing forgery detection method is efficient at detecting image splicing and can handle various kinds of image manipulation. Various approaches might be more successful in different situations, so it is crucial to carefully assess any method's performance before applying it to a real-world application [9,10].

One popular technique for detecting image splicing is error level analysis (ELA). In this technique, the difference is computed between the input image and the resaved image, and then a threshold is applied to identify areas of the altered image. A novel method using the relationship between noise variance and image block sharpness is proposed by [11]. Based on the observation the noise level in the image varies across different areas of the image, the blind image splicing detection method was developed. The technique works by first dividing the image into small blocks and computing the "noise level function" (NLF) for each block. The NLF is a function that relates the variance of the pixel values in a block to the mean of the pixel values in that block. In an unmanipulated image, the NLF is expected to be consistent across all blocks, but in a spliced image, the NLF will show inconsistencies across different image blocks. The technique was implemented using various algorithms, including the "color filter array" (CFA) algorithm and the "singular value decomposition" (SVD) algorithm. The CFA-based algorithm computes the NLF, while the SVD estimates the noise level. Blind image splicing detection using noise level function has some limitations. For example, it may not be effective for detecting carefully manipulated images.

A similar approach based on error level analysis ELA and the "local binary pattern" (LBP) for image splicing detection was proposed by [12]. The LBP is a texture analysis method that extracts features from the image by analyzing the local binary patterns of the pixels. This method operated by first creating an ELA image using the original and suspected images. Then, LBP features were extracted from the ELA images. Finally, a classifier is used to determine whether the image is authenticated or spliced. In conclusion, ELA-based image-splicing detection is not always accurate because some image-splicing methods created to bypass this technique.

The study in [13] proposed a "two-dimensional discrete cosine transform" (DCT) with a "discrete wavelet transform" (DWT) feature-based model to detect image tampering. The DCT and DWT transforms can extract different features from images, such as statistical moments, texture descriptors, and wavelet coefficients. These features are used to train SVM to recognize altered images. Overall, low-dimensional DCT and DWT feature-based models are suitable approaches to detecting image forgery. It is crucial to remember that these models might not be robust to more sophisticated tampering methods.

Fractional calculus methods for image feature extraction have been used recently for image forgery detection. One of the challenges in detecting image splicing is that the spliced regions can be visually like the surrounding regions, making it difficult to distinguish between them. One approach to detecting image splicing is to use conformable focus measures [14], which compute scores indicating the level of focus in different regions of an image. In this approach, focus measures apply to different regions of an image to detect spliced areas. However, using a single focus measure may not be sufficient to detect image splicing. To solve this problem, researchers have proposed combining multiple conformable focus measures to improve detection accuracy. Additionally, redundant discrete wavelet transforms coefficients with conformable focus measures provide more robust features for detecting splicing and improve the accuracy of splicing detection. The experimental results have shown that this method achieves better accuracy in detecting image splicing concerning existing methods.

Moreover, the study by Jalab et al. [2] proposed a new approach for detecting digital image splicing forgery using a modified fractional entropy texture descriptor. The proposed method in this paper uses a modified version of fractional entropy, which measures the texture complexity in an image. The modification involves applying a threshold to the fractional entropy values to create a binary texture descriptor. This descriptor is used to classify image blocks as either spliced or authentic. The results showed that the modified fractional entropy descriptor outperformed other texture descriptors in terms of detection accuracy and robustness to noise. Overall, the study offers a promising method for splicing forgeries of digital images using texture analysis, which could be helpful in a variety of applications like digital forensics and content authentication. Furthermore, the traditional approach can be computationally intensive, especially when analyzing large images or datasets.

2.2 Deep Learning Methods of Feature Extraction

Deep learning (DL) is a part of machine learning that uses artificial neural networks [15]. DL is used widely in image classification, recognition, and security computing. Detecting splicing image forgery is a challenging problem that requires specialized techniques. Deep learning approaches usually use a "convolutional neural network" (CNN). In several image processing tasks, deep learning has demonstrated promising results, including image forgery detection, in recent years. Wang et al. [16] proposed an image splicing detecting method using a CNN with a weight combination strategy. This study involved training CNN on a large dataset of authentic and manipulated images. The weight combination strategy involved combining the outputs of multiple CNNs with different weights to improve the detection accuracy. The study evaluated using several datasets of authentic and manipulated images, and it achieved high detection rates with low false-positive rates. Variety of splicing manipulations, such as copy-move, splicing, and retouching, could be detected using the method. The study by Liu et al. [17] proposed a method for detecting splicing forgery in images based on a neural network. Overall, this study presented a promising approach for detecting splicing forgery in realistic scenes using deep learning techniques. Another method used "generative adversarial network" (GAN) for detecting splicing image forgery [18]. The GAN consists of two neural networks: a generator creates fake images, while the discriminator differentiated between authentic and forged images. In the case of splicing detection, the generator trained to create fake images, that resemble spliced images, while the discriminator trained to distinguish between spliced and authentic images. To further improve the accuracy of splicing detection, reality transforms adversarial generators used to transform the input images into a different domain, such as grayscale or frequency domain, before generating the fake images. This approach provided a powerful tool for detecting and localizing image-splicing forgeries. The study by Hosny et al. [19] proposed a new method to detect splicing image forgery using CNN. The proposed method involves training CNN to classify image patches into authentic or spliced. The CNN architecture consists of several convolutional layers followed by fully connected layers. The input to the CNN is a patch of the image, and the output is a binary classification indicating whether the patch is authentic or spliced. The CNN trained using a binary cross-entropy loss function. In conclusion, the proposed deep learning-based approaches have shown promising results in splicing forgery detection. It is crucial to remember that these algorithms are not error-free and are still subject to error. Therefore, it is crucial to employ a variety of techniques and be aware of each one's drawbacks. Table 1 illustrates summary of current research on image splicing detection.

Table 1: A review of recent studies on image splicing detection

References	Method	Accuracy (%)	Limitations
	Traditional methods of feature extraction		
[11]	Using the relationship between noise variance and image block sharpness	83.75	The spliced region cannot be located precisely using a large block size.
[12]	Based on error level analysis ELA and LBP	97.30	This method was computationally intensive because it used a series of post-processing operations to locate the tampered area.
[13]	DCT and DWT	98.5	This approach applies DCT and DWT are computationally expensive, especially for large images or when dealing with many images.
[14]	Redundant discrete wavelet transforms coefficients with conformable focus measures	97.60	The single focus measure may not be sufficient to detect image splicing.
	Deep Learning methods of feature extraction		
[17]	CNN with a weight combination strategy	99.32	CNNs can be computationally demanding when working with high-resolution images or large datasets. Furthermore, low image quality may degrade CNN performance.
[18]	Generative adversarial network (GAN)	88.88	Different hyperparameters, such as learning rates, network architectures, and regularization techniques, are used in the adversarial generators. The selection and tuning of these hyperparameters may affect the generator's performance. Finding the best configuration may necessitate extensive experimentation and domain knowledge.
[19]	Deep learning CNN	99.3	For CNNs to learn effectively, a lot of labeled training data is typically needed. Furthermore, manually labeling spliced regions in images can be subjective and prone to human errors, affecting CNN's performance.

3 Materials and Methods

The study's main objective is to propose a new technique for detecting image splicing using Sonine functions convex and the deep features model (SFC-DFs). Fig. 2 shows the steps of the proposed image splicing detection model, which includes preprocessing, feature extraction using SFC-DFs, dimensionality reduction, and classification.

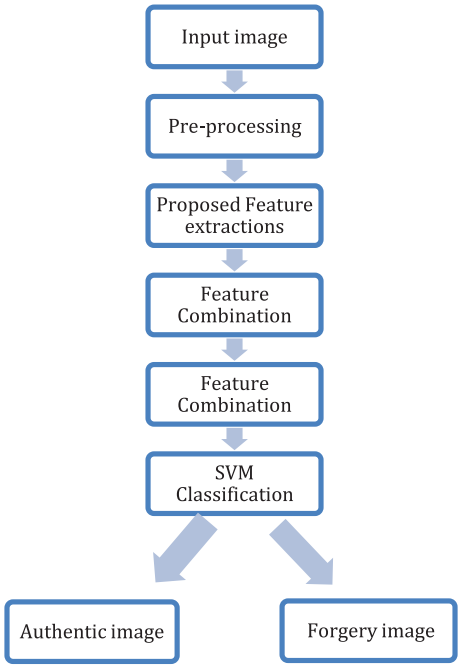

Figure 2: Image splicing detection model proposed block diagram

3.1 Preprocessing

The preprocessing step is used to improve feature extraction, which improves the algorithm's accuracy. The color input image transformed from RGB (Red, Green, and Blue) into the YCbCr color space (Y is luma, Cb is Blue Chroma, Cr is Red Chroma). The Y component represents the image brightness, while the Cb and Cr components represent the color information. The YCbCr color space is often used in image and video processing applications because it separates the luminance and chrominance information. Fig. 3 depicts the RGB and YCbCr versions of an image. The most well-known and frequently utilized color space in digital imaging is RGB. The RGB color spaces, however, are inappropriate for use in image splicing detection due to the close relationship between red, green, and blue.

3.2 Proposed Feature Extraction

In this sub-section, new feature extraction using functions convex and deep features model (SFC-DFs).

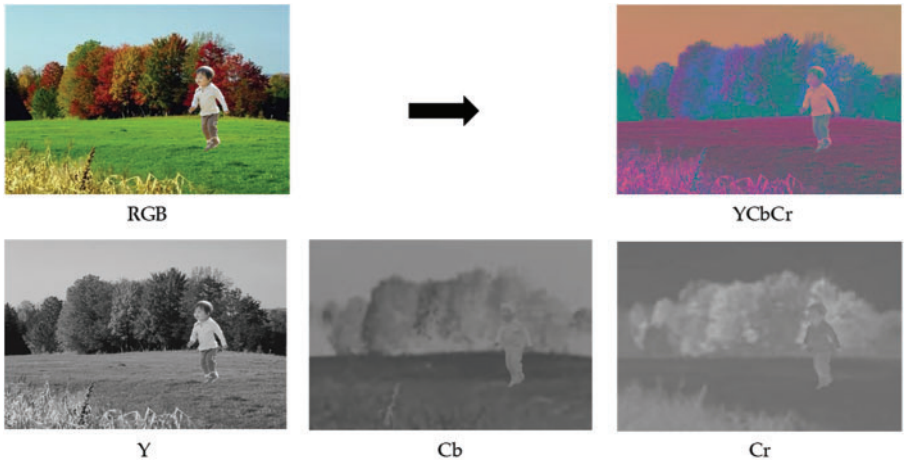

Figure 3: Preprocessing of the input image

3.2.1 Proposed Sonine Functions Convex Feature Extraction (SFC)

The textural aspects of the image are directly impacted by the image forgery, as was already mentioned. Authentic from fake images can be distinguished quantitatively using the distribution of textural qualities, which includes patterns and structures like pixel placement, texture, and color. In this study Sonine functions convex-based image feature extraction model is proposed as a feature extraction approach for image splicing detection. The Sonine functions are a family of special functions that arise in the study of problems in mathematical physics, such as the theory of gases and fluids [20]. The Sonine functions are defined as a sequence of orthogonal polynomials, which are solutions to a particular differential equation known as the Sonine equation. The Sonine functions (SFs) provide increasingly accurate approximations of the distribution as the order of the function increases. The properties of the Sonine functions are extensively studied, and they have been found to have a wide range of applications in physics and other areas of science and engineering. There are two different formulas for SFs, as follows:

$$\Phi^{\beta}{}_{1}(\chi) = \sum_{n=0}^{\infty} \left(\frac{\chi^{n-\beta}}{\Gamma(n+1)\,\Gamma(n+1-\beta)} \right) \tag{1}$$

and

$$\Phi^{\beta}{}_{2}(\chi) = \sum_{n=0}^{\infty} \left(\frac{\chi^{n+\beta-1}}{\Gamma(n+1)\,\Gamma(n+\beta)} \right) \tag{2}$$

We proceed to generalize SFs by using the k-symbol fractional calculus which is verbalized by Diaz et al. [21], as follows:

Definition 1

The stimulate gamma function, identified as the k-symbol gamma function, is supposed by the prescription:

$$\Gamma_{k}(x) = \lim_{n\to\infty} \frac{n!\,k^{n}\,(nk)^{\frac{x}{k}-1}}{(x)_{n,k}}, \tag{3}$$

where

$$(x)_{n,k} := x\,(x+k)\,(x+2k)\cdots(x+(n-1)\,k,$$

$$(x)_{n,k} = \frac{\Gamma_k\,(x+nk)}{\Gamma_k\,(x)}. \tag{4}$$

Note that $\Gamma_k\,(x) \to \Gamma\,(x)$ when $k \to 1$, and $\Gamma_k\,(x+k) = x\Gamma_k\,(x)$, $\Gamma_k\,(k) = 1$.

Founded by the k-symbol description, k-symbol SFs can be presented, as follows:

$$\left[\Phi^{\beta}{}_1\,(\chi)\right]_{\mathrm{k}} = \sum\nolimits_{\mathrm{n}=0}^{\infty}\left(\frac{\chi^{\mathrm{n}-\beta}}{\Gamma_{\mathrm{k}}\,(\mathrm{n}+1)\,\Gamma_{\mathrm{k}}\,(\mathrm{n}+1-\beta)}\right) \tag{5}$$

and

$$\left[\Phi^{\beta}{}_2\,(\chi)\right]_{k} = \sum\nolimits_{\mathrm{n}=0}^{\infty}\left(\frac{\chi^{\mathrm{n}+\beta-1}}{\Gamma_{\mathrm{k}}\,(\mathrm{n}+1)\,\Gamma_{\mathrm{k}}\,(\mathrm{n}+\beta)}\right) \tag{6}$$

Note that when $k = 1$, the classic SF is attained.

Now combining (5) and (6), we get

$$\left[\Phi^{\beta}\,(\chi)\right]_{\mathrm{k}} = \alpha\ \left[\ \Phi^{\beta}{}_1\,(\chi)\right]_{\mathrm{k}} + (1-\alpha)\left[\ \Phi^{\beta}{}_2\,(\chi)\right]_{\mathrm{k}} \tag{7}$$

Experimentally, the fractional parameters α, β, are fixed to 0.5, while k is the k-symbol ε N, and χ is the pixel probability of the image. Eq. (7) indicates the convex situation between SFs. The extraction of image features using convexity-based algorithms is a common technique for image processing. Convexity-based algorithms can precisely identify the boundaries of objects in images with complex shapes. The proposed SFs-based model includes the following steps:

i. The RGB input image is converted to YCbCr color space and divided into equal-sized blocks.
ii. Features are extracted from each block using Eqs. (5)–(7) after experimentally fixing the k-symbol and the fractional parameters, α and β.
iii. Images are divided into two groups using the SVM classifier: authenticated or spliced forged images.

The distribution of textural qualities can provide a quantitative basis for identifying the authenticity of forged images. Textural quality refers to the patterns and structures within an image, such as the arrangement of pixels, texture, and color. These qualities are unique to each image and can be used to detect manipulated images. In terms of texture feature extraction, Sonine functions have several advantages over other widely used transforms like DCT and Wavelet Transform. Sonine functions are computationally efficient, making them suitable for image processing applications. However, the DCTs and wavelet transforms may not be as effective as Sonine functions in capturing the complex spectral and temporal characteristics of spliced images, which can lead to increased computational complexity and reduced classification accuracy. The reduction of feature dimensionality is used to ensure that the algorithm operates at the most efficient and optimal settings. The current study employs the "Mean", "Variance", "Skewness", and "Kurtosis" to reduce the dimension of the features in each image.

The "Mean" for features M of scalar observations F, is described as

$$M = \frac{1}{M}\sum\nolimits_{I=1}^{M} F_i \tag{8}$$

The "Variance" is defined as

$$V = \frac{1}{M-1}\sum_{i=1}^{NM} |F_i - \mu|^2 \tag{9}$$

where μ is the "Mean" of F_i

The "Skewness" refers to measurements of the asymmetries of feature data around the feature mean.

$$S = \frac{V\,(x-\mu)^3}{\sigma^3} \tag{10}$$

where σ stands for standard deviation, and V stands for the quantity estimated value.

The 'Kurtosis' is characterized as

$$K = \frac{V\,(x-\mu)^4}{\sigma^4} \tag{11}$$

The Standard Deviation described as follows:

$$Sd = \sqrt{\frac{1}{M-1}}\sum_{i=1}^{M} |F_i - \mu|^2 \tag{12}$$

3.2.2 CNN-Based Deep Features Model (DFs)

A CNN is a type of deep learning model that is primarily used for image and video recognition and processing tasks. Convolutional layers are a type of neural network used by CNNs to extract features from images. Learning a set of filters that can be convolved with the input data to produce feature maps is the fundamental principle behind a CNN. The filters are learned during training by backpropagation, and they are used to capture different patterns in the input data, such as edges, curves, and textures. The feature maps are then passed through a series of non-linear activation functions, pooling layers, and fully connected layers, which combine the features to make predictions about the input data. The output of the convolution and pooling layers is then passed through one or more fully connected layers, which perform classification based on the extracted features. The fully connected layers use an activation function such as Softmax to assign probabilities to each possible class. CNNs can be trained on large datasets to learn the features that are most important for a given task. Once trained, they can be used to classify new images with high accuracy. CNNs have been used successfully in a wide range of applications, including image and video recognition, natural language processing, and speech recognition. There are two crucial components of the CNN model: the feature learning (convolutional and pooling layers) and the classification component (fully connected layers). The convolutional layer composes several convolutional filters. After each convolutional layer, the dimensions of the input image are decreased due to the stride process. Therefore, to retrieve the original spatial dimensions of the input volume, zero-padding is used to pad the input volume with zeros. Then, apply an element-wise nonlinear activation function on the obtained feature map through the "rectified linear unit" (ReLU) layer. Subsequently, the rectified feature map is passed through the pooling layer for dimensionality. Moreover, the maximum number in each sub-region of the feature maps is determined by the max-pooling function. A batch normalization layer is used to regulate and speed the training process of CNN by normalizing the produced feature maps. The Adam optimizer is used to decrease the error function of CNN and produced an extremely improved weight when the learning rate was set to 0.001. Preprocessing is the initial phase. The image is reduced in size to $227 \times 227 \times 3$ at this stage so that it can be seamlessly inserted into the following stage. The second

stage is the feature extraction. At this stage, each convolutional layer is followed by the following: a batch normalization layer, a Relu Layer, and a max-pooling layer.

The model architecture of the proposed CNN-based feature extraction involves the following elements: 16 feature maps, (5, 5) filter, and an activation function (RELU) are present in the first convolutional layer. With a pool size of (2, 2) in the first max-pooling layer (2, 2). The second convolutional layer has an activation function (RELU), a filter size of (5, 5), and 32 feature maps. The pool size of the second max-pooling layer is (2, 2). With the same filter size of (5, 5) and RELU activation function, the third convolutional layer has 64 feature maps. In addition, the pool size of the third max-pooling layer is (2, 2). With the same filter size of (5, 5) and RELU activation function. The fourth convolutional layer has 128 feature maps. In addition, the pool size of the fourth max-pooling layer is (2, 2). Finally, the fifth convolutional layer has 256 feature maps with the same filter size and same activation function (RELU), and the max-pooling layer has a pool size of (2, 2). CNN-based feature extraction is a powerful tool to detect splicing image forgery. For feature extraction, the last fully connected layer of the CNN network is removed, and the resulting feature map is flattened to obtain a feature vector for each input image. The CNN network's hyperparameters were fixed to enable the convergence of the loss function during training. The categorical cross-entropy loss and the Adam Optimizer are used in the training model with a learning rate of 0.0001 for 60 iterations, 30 validation frequencies, and 32 batch sizes. Deep learning feature extraction models can learn invariant representations directly from the data which shows promise in handling geometric transformations effectively. Due to their ability to accurately capture complex patterns and spatial information in images, CNN-based features are frequently used for image splicing detection. The benefits of CNN-based features make them ideal for tasks requiring the detection of image splicing.

3.3 Classification

In this study, MATLAB R2021b [22] on Windows 10 with processor Intel(R) Core i7-7700HQ CPU @ 2.80 GHz 16 GB was used to produce the test results. The SVM classifier was used to classify images into authenticated and spliced images. The SVM is a popular machine-learning algorithm used for classification. The SVM classifier works by finding a hyperplane that separates the data into different classes with the maximum possible margin. SVMs have several advantages over other classifiers due to the following factors:

- Simplicity: SVMs are easier to comprehend and use. Compared to CNNs, they consume less computational power.
- Avoid Overfitting: SVMs have solid theoretical bases to prevent overfitting in scenarios where the data may not be separable.
- The choice between SVMs and CNNs for classifying images largely depends on the problem's nature, the dataset's size, and the dataset's composition.

3.4 Image Dataset

A publicly accessible "CASIA V2.0" dataset is utilized in this study [23]. There are 12613 images in the dataset, 7408 of which are authenticated images and 5122 of which are forgery. This dataset is regarded as a standard in image splicing detection and is used in the literature. Examples of the CASIA V2.0 image dataset are shown in Fig. 4.

Figure 4: CASIA V2.0 dataset samples. (A), and (B) contain authentic images, whereas (C) contains spliced images

3.5 Evaluation Metrics

Depending on the task and requirements, different evaluation metrics are used to assess the effectiveness of the image forgery detection model. A confusion matrix has two axes: predicted values and actual values. The elements of a confusion matrix are TP-Forged images detected as forged. FN-Forged images detected as authentic. FP-Non-Forged images detected as forged. TN-Authentic images

detected as authentic. The most widely used evaluation metrics for identifying image forgery are listed:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{15}$$

4 Results

This study employs 5-fold cross-validation. Five subsets of the dataset were created, and the main procedure was repeated five times, with each iteration using 30% of the images for testing and 70% for training. The process of 5-fold cross-validation is used to assess the machine learning algorithm performance on a dataset by dividing the data into five equal-sized folds, four of which are used for training and one for testing. Each fold was tested once, and the remaining four folds were used for training. The nested 5-fold cross-validation procedure reduces the possibility of overfitting the training data. The performance results are illustrated in Table 2. Highest detection accuracy for proposed SFC features and DFs achieved by SVM with a quadratic kernel.

Table 2: The results of the proposed model over CASIA V2.0

	Accuracy (%)	Precision	Recall
SFC features	95.60	94.21	94.65
CNN features	94.90	93.13	94.76
SFC and CNN features	98.93	97.21	97.95

Table 3 displays the classification outcomes from the three classifiers: long short-term memory (LSTM), CNN, and SVM. The LSTM is used for sequential data, but because it can recognize image features over time by connecting memory blocks through its layers. According to Table 3, the SVM classifier outperformed the other two classifiers. The rates generated by the three classifiers were nevertheless remarkably similar. The SVM classifier has several advantages in splicing detection. They are computationally efficient, making them suitable for image classification applications. Additionally, they are less likely to overfit, which happens when a model is overly complex and fits the training data too closely.

Table 3: Results of various classifiers using proposed SFC and DFs features

Classifiers	Accuracy (%)	Precision	Recall
LSTM	90.72	90.52	90.165
CNN	92.49	91.32	91.87
SVM	98.93	97.21	97.95

The high detection accuracy measures the overall accuracy of the detection method by calculating the percentage of correctly classified spliced and authentic images. Accuracy is an important metric

to evaluate the performance of the detection algorithm. The proposed approach has a high detection capability on the CASIA V2.0 dataset, which is a challenging database.

5 Comparison with Existing Methods

This section compares the proposed method to various techniques. The performance is evaluated in terms of Recall, Precision, and detection accuracy. Each method is described as follows:

• Alahmadi et al. [24] presented a method for detecting image forgery using a combination of DCT and LBP features. The proposed method first applies DCT to the image to obtain frequency coefficients. Then, LBP is applied to the resulting DCT coefficients to extract texture information. Frequency and texture domain inconsistencies are frequently introduced by image forgeries. The CASIA V2.0 dataset was used in the study's evaluation.

• El-Latif et al. [25] proposed a method for detecting image splicing in two main parts: deep learning and the Haar wavelet transform. The deep learning part based on the VGG-16 architecture as their CNN model. While the Haar wavelet transform coefficients used to extract the final features. The proposed method evaluated using two different datasets (CASIA v1.0 and CASIA v2.0).

• Ding et al. [26] proposed "DCU-Net" for detecting splicing image forgery. The network uses two separate inputs: one for the original image and another for the potentially spliced image to improve the accuracy and robustness of the network by allowing it to capture both low-level and high-level features.

• Shen et al. [27] presented a method for detecting image splicing forgery using textural features based on grey-level co-occurrence matrices (GLCMs). The proposed method first divides the image into blocks and computes the GLCMs for each block. Then, it extracts four textural features from each GLCM: contrast, correlation, energy, and homogeneity. These features capture different aspects of image texture.

• Nath et al. [28] proposed a method for detecting image splicing using deep CNN-learned features. This method based on VGG-16 network and evaluated on CASIA V2.0 dataset.

As can be observed from Table 4, the proposed method achieved a detection accuracy of 98.93% on CASIA V2.0. Alahmadi et al. [24] recorded the highest Precision value of 98.45%. In addition, the highest Recall belongs to the El-Latif et al. [25] method with 99.03%. The proposed method presents a promising approach for detecting image splicing using proposed texture features of Sonine functions convex combined with CNN-based features.

Table 4: Comparison with different methods of detecting image splicing forgery using the CASIA V2.0 dataset

Method	Recall (%)	Precision (%)	Accuracy (%)	Feature extraction	Classifier
Alahmadi et al. [24]	96.84	98.45	97.50	Hand-crafted features (LBP and DCT)	SVM
El-Latif et al. [25]	99.03	97.14	96.36	Deep learning and hand-crafted features (Haar Wavelet Transform)	SVM

(Continued)

Table 4 (continued)

Method	Recall (%)	Precision (%)	Accuracy (%)	Feature extraction	Classifier
Ding et al. [26]	88.93	87.72	97.93	Deep learning features (dual-channel U-Net)	U-Net
Shen et al. [27]	97.73	97.72	97.30	Hand-crafted features (grey level co-occurrence matrices)	SVM
Nath et al. [28]	94.15	96.69	96.45	Deep learning CNN features.	ANN
Proposed	97.95	97.21	98.93	Deep learning CNN features and hand-crafted features (Sonine functions convex)	SVM

6 Conclusions

Combining and merging images to produce a composite fabricated image is known as image splicing. In this study, a new method for detecting image-splicing forgery was proposed. The proposed method uses a new feature extraction model based on deep features with the proposed Sonine functions convex features. The proposed CNN was used to automatically generate the deep features from the color image, while the proposed Sonine functions convex used to extract the texture features from the input images. Finally, the SVM was utilized for classification. The method is reliable for identifying image splicing forgery, as demonstrated by the results on the CASIA V2.0 dataset. The study is limited in its ability to locate the areas or places where image tampering has occurred. Future work will need to investigate more advanced forensics and image analysis methods that can provide a higher level of accuracy in detecting and localizing tampering.

Acknowledgement: The authors would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charge (APC) of this publication and their support.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Study conception and design: A.R.A.; data collection: A.R.A.; analysis and interpretation of results: R.W.I., A.R.A.; draft manuscript preparation: R.W.I. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The dataset analyzed during this study is a standard image forgery dataset and is available from [23].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. P. Jaiprakash, M. B. Desai, C. S. Prakash, V. H. Mistry and K. L. Radadiya, "Low dimensional DCT and DWT feature based model for detection of image splicing and copy-move forgery," *Multimedia Tools and Applications*, vol. 79, no. 1, pp. 29977–30005, 2020.

[2] H. A. Jalab, T. Subramaniam, R. W. Ibrahim, H. Kahtan and N. F. M. Noor, "New texture descriptor based on modified fractional entropy for digital image splicing forgery detection," *Entropy*, vol. 21, no. 4, pp. 371–381, 2019.

[3] M. A. Elaskily, M. M. Dessouky, O. S. Faragallah and A. Sedik, "A survey on traditional and deep learning copy move forgery detection (CMFD) techniques," *Multimedia Tools and Applications*, vol. 1, no. 1, pp. 1–27, 2023.

[4] H. Farid, "Image forgery detection," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16–25, 2009.

[5] M. Bilal, H. Habib, Z. Mehmood, R. Yousaf, T. Saba *et al.,* "A robust technique for copy-move forgery detection from small and extremely smooth tampered regions based on the DHE-SURF features and mDBSCAN clustering," *Australian Journal of Forensic Sciences*, vol. 53, no. 4, pp. 459–482, 2020.

[6] Z. N. Khudhair, F. Mohamed, A. Rehman, T. Saba and S. A. Bahaj, "Detection of copy-move forgery in digital images using singular value decomposition," *Computers, Materials & Continua*, vol. 74, no. 2, pp. 4135–4147, 2023.

[7] S. Dadkhah, M. Koppen, S. Sadeghi, K. Yoshida, H. A. Jalab *et al.,* "An efficient ward-based copy-move forgery detection method for digital image forensic," in *Proc. of IVCNZ*, Christchurch, New Zealand, pp. 1–6, 2017.

[8] A. Baumy, A. D. Algarni, M. Abdalla, W. El-Shafai, F. E. Abd El-Samie *et al.,* "Efficient forgery detection approaches for digital color images," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3257–3276, 2022.

[9] S. Sadeghi, S. Dadkhah, H. A. Jalab, G. Mazzola and D. Uliyan, "State of the art in passive digital image forgery detection: Copy-move image forgery," *Pattern Analysis and Applications*, vol. 21, no. 2, pp. 291–306, 2018.

[10] D. M. Uliyan, H. A. Jalab and A. W. Wahab, "Copy move image forgery detection using Hessian and center symmetric local binary pattern," in *Proc. of ICOS*, Malacca, Malaysia, pp. 7–11, 2015.

[11] N. Zhu and Z. Li, "Blind image splicing detection via noise level function," *Signal Processing: Image Communication*, vol. 68, no. 2, pp. 181–192, 2018.

[12] Y. Zhang, T. Shi and Z. M. Lu, "Image splicing detection scheme based on error level analysis and local binary pattern," *Journal of Network Intelligence*, vol. 6, no. 2, pp. 303–312, 2021.

[13] S. P. Jaiprakash, M. B. Desai, C. S. Prakash, V. H. Mistry and K. L. Radadiya, "Low dimensional DCT and DWT feature based model for detection of image splicing and copy-move forgery," *Multimedia Tools and Applications*, vol. 79, no. 39, pp. 29977–30005, 2020.

[14] T. Subramaniam, H. A. Jalab, R. W. Ibrahim and F. M. Noor, "Improved image splicing forgery detection by combination of conformable focus measures and focus measure operators applied on obtained redundant discrete wavelet transform coefficients," *Symmetry*, vol. 11, no. 11, pp. 1392–1402, 2019.

[15] C. S. Yadav, J. Singh, A. Yadav, H. S. Pattanayak, R. Kumar *et al.,* "Malware analysis in IoT & Android systems with defensive mechanism," *Electronics*, vol. 11, no. 15, pp. 1–20, 2022.

[16] J. Wang, Q. Ni, G. Liu, X. Luo and S. K. Jha, "Image splicing detection based on convolutional neural network with weight combination strategy," *Journal of Information Security and Applications*, vol. 54, no. 1, pp. 1–8, 2020.

[17] B. Liu and C. M. Pun, "Exposing splicing forgery in realistic scenes using deep fusion network," *Information Sciences*, vol. 526, no. 1, pp. 133–150, 2020.

[18] X. Bi, Z. Zhang and B. Xiao, "Reality transform adversarial generators for image splicing forgery detection and localization," in *Proc. of IEEE/CVF*, Montreal, Canada, pp. 14294–14303, 2021.

[19] 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, pp. 1272–1282, 2023.

[20] K. Stempak, "A new proof of Sonine's formula," in *Proc. of AMC*, Rhode Island, USA, pp. 453–457, 1988.

[21] R. Diaz and E. Pariguan, "On hypergeometric functions and Pochhammer *k*-symbol," *Divulgaciones Matemáticas*, vol. 15, no. 2, pp. 179–192, 2007.
[22] Matlab tools. The Mathworks Inc., Natick, Massachusetts, USA, 2021.
[23] CASIA tampered image detection evaluation database (CASIA TIDE v2.0). [Online]. Available: http://forensics.idealtest.org:8080/index_v2.html (accessed on 10/03/2023)
[24] A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis *et al.,* "Passive detection of image forgery using DCT and local binary pattern," *Signal, Image and Video Processing*, vol. 11, no. 1, pp. 81–88, 2017.
[25] A. El-Latif, I. Eman, A. Taha and H. H. Zayed, "A passive approach for detecting image splicing using deep learning and Haar wavelet transform," *International Journal of Computer Network & Information Security*, vol. 11, no. 5, pp. 28–35, 2019.
[26] H. Ding, L. Chen, Q. Tao, Z. Fu, L. Dong *et al.,* "DCU-Net: A dual-channel U-shaped network for image splicing forgery detection," *Neural Computing and Applications*, vol. 35, no. 7, pp. 5015–5031, 2023.
[27] X. Shen, Z. Shi and H. Chen, "Splicing image forgery detection using textural features based on the grey level co-occurrence matrices," *IET Image Processing*, vol. 11, no. 1, pp. 44–53, 2017.
[28] S. Nath and R. Naskar, "Automated image splicing detection using deep CNN-learned features and ANN-based classifier," *Signal, Image and Video Processing*, vol. 15, no. 1, pp. 1601–1608, 2021.