

Research on Splicing Image Detection Algorithms Based on Natural Image Statistical Characteristics

Ao Xiang^{1,a}, Jingyu Zhang^{2,b}, Qin Yang^{3,c}, Liyang Wang^{4,d}, Yu Cheng^{5,e}

¹*School of Computer Science & Engineering (School of Cybersecurity), Digital Media Technology, University of Electronic Science and Technology of China, Chengdu, Sichuan, China*

²*The Division of the Physical Sciences, The University of Chicago, Analytics, Chicago, IL, USA*

³*School of Integrated Circuit Science and Engineering (Exemplary School of Microelectronics), Microelectronics Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, China*

⁴*Olin Business School, Washington University in St. Louis, Finance, St. Louis, MO, USA*

⁵*The Fu Foundation School of Engineering and Applied Science, Operations Research, Columbia University, New York, NY, USA*

^a*xiangao1434964935@gmail.com*, ^b*simonajue@gmail.com*, ^c*yqin0709@gmail.com*,
^d*liyang.wang@wustl.edu*, ^e*yucheng576@gmail.com*

Keywords: Image tampering detection; Natural image; statistical characteristics; Machine learning; Digital image processing

Abstract: With the development and widespread application of digital image processing technology, image splicing has become a common method of image manipulation, raising numerous security and legal issues. This paper introduces a new splicing image detection algorithm based on the statistical characteristics of natural images, aimed at improving the accuracy and efficiency of splicing image detection. By analyzing the limitations of traditional methods, we have developed a detection framework that integrates advanced statistical analysis techniques and machine learning methods. The algorithm has been validated using multiple public datasets, showing high accuracy in detecting spliced edges and locating tampered areas, as well as good robustness. Additionally, we explore the potential applications and challenges faced by the algorithm in real-world scenarios. This research not only provides an effective technological means for the field of image tampering detection but also offers new ideas and methods for future related research.

1. Introduction

In the digital age, image splicing technology is widely used in artistic creation, filmmaking, advertising, and news reporting due to its ability to create compelling visual effects. However, with the misuse of this technology, especially in the fields of public information dissemination and legal evidence, the negative impacts of spliced forged images have become increasingly prominent. As shown in <Figure 1> of a spliced forged image, the manipulation of the original content of the image is clearly visible, which could mislead public perception and decision-making. Thus, developing an accurate and efficient method for detecting spliced images has become an urgent issue.

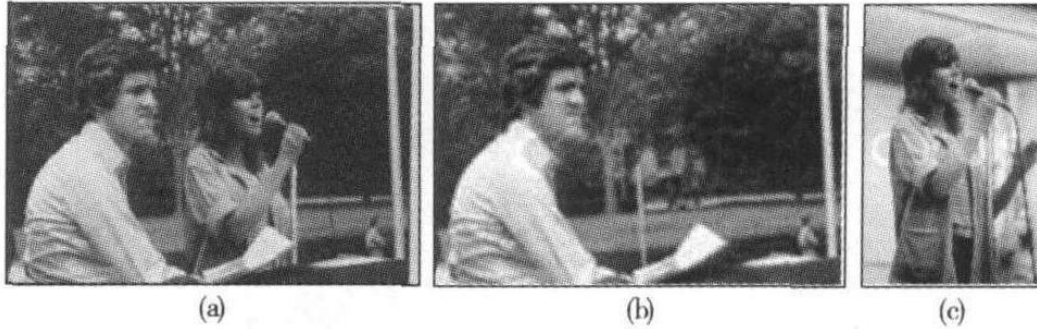


Figure 1: Example of stitching forged images

The main motivation for this study stems from the challenges of detecting spliced images, especially in cases where the tampering details are not easily noticeable. As illustrated in <Figure 1>, the spliced area merges almost seamlessly with the surrounding image content, posing a challenge to existing detection techniques. To address this issue, we propose a splicing image detection algorithm based on the statistical characteristics of natural images. This algorithm detects and locates splicing tampering by analyzing the statistical patterns and inconsistencies in the image. Building on this, we have further delved into the characteristics of image splicing to design and implement a detection framework that combines statistical analysis and machine learning techniques. Compared to traditional splicing detection methods, our algorithm demonstrates higher accuracy and adaptability in handling complex scenarios like those shown in <Figure 1>. This paper not only provides a detailed introduction to the theoretical foundations and experimental validation of the algorithm but also conducts an exhaustive analysis of the experimental results, proving its effectiveness across multiple recognized datasets[1].

2. Literature Review

2.1 Image Splicing Detection Techniques

Image splicing detection is a research hotspot in the field of digital image processing, aimed at identifying and locating tampering traces caused by splicing techniques. With technological advancements, a variety of methods have been proposed to address the increasingly sophisticated image tampering techniques. Traditional splicing detection methods are primarily based on pixel-level analysis, such as detecting discontinuities in edges and sudden changes in color or brightness. These methods typically rely on the significant visual differences between the spliced area and the original image. However, as splicing techniques improve, these visual differences become less apparent, reducing the effectiveness of traditional methods. To address this issue, researchers have begun to explore methods based on the complex statistical characteristics of image content. For example, using second-order or higher-order statistical models, such as co-occurrence matrices and Fourier transforms, to analyze changes in image texture and frequency characteristics. These methods have enhanced detection sensitivity, especially in handling subtle splicing areas. Further research has introduced methods based on machine learning, particularly deep learning technologies such as Convolutional Neural Networks (CNNs). These methods can automatically extract and learn high-level image features, demonstrating exceptional detection performance. However, they typically require a large amount of labeled data to train the models, and the interpretability of the models is not as good as that based on statistical methods. The latest research trend is to combine statistical characteristics with machine learning methods. By extracting the natural statistical properties of images and using these properties as inputs for machine learning algorithms, researchers have

developed more powerful and robust splicing detection algorithms. This approach not only identifies splicing areas but also classifies splicing techniques, laying the foundation for deeper image tampering analysis[2].



Figure 2: Shows a promotional photo of George W. Bush during his campaign

In this paper, we focus particularly on this method of combining natural image statistical characteristics with machine learning techniques for splicing image detection. Compared to the aforementioned methods, the algorithm proposed in this study demonstrates improved performance in handling complex splicing scenarios like the one shown in <Figure 2>. We will detail this method in the following chapters and validate its effectiveness through a series of experiments[3].

2.2 The Application of Statistical Methods in Image Analysis

Image analysis, a field of scientific research, continuously benefits from the advancements in statistical methods. These methods play a crucial role in various aspects such as image quality evaluation, feature extraction, pattern recognition, and tampering detection. Particularly in the domain of image splicing detection, statistical methods have shown their irreplaceable value. Statistical analysis involves mathematical modeling of image data to reveal the intrinsic laws and characteristics of image content. For instance, first-order statistical moments, such as the mean and variance, can describe the brightness and contrast of an image, while higher-order moments, such as skewness and kurtosis, can capture features like the asymmetry and sharpness of the image. These basic statistical quantities provide means for understanding and analyzing the fundamental properties of images. In the field of image splicing detection, the use of statistical methods is more profound and complex. Image splicing often introduces anomalous statistical features in localized areas, which can be detected by analyzing the pixel distribution of the image. For example, the pixel intensity of natural images usually follows specific statistical distributions, while splicing operations can distort these distributions, leading to abnormal statistical characteristics. By detecting these anomalies, the tampered areas can be effectively located. Beyond pixel-level features, the texture of an image is also a significant focus of statistical analysis. Texture analysis often employs tools like the Gray Level Co-occurrence Matrix (GLCM) and wavelet transforms, which can extract both local and global texture information from an image, as shown in <Figure 3>. In tampering detection, discontinuities in texture often indicate potential splicing boundaries. Recently, statistical learning methods, particularly those based on deep learning for feature learning, have brought new perspectives to image splicing detection. Deep neural networks can learn complex patterns in image data and effectively classify and recognize them. These networks, trained on extensive datasets, are capable of detecting subtle statistical changes caused by splicing operations, even if these changes are too minute for traditional statistical methods[4].

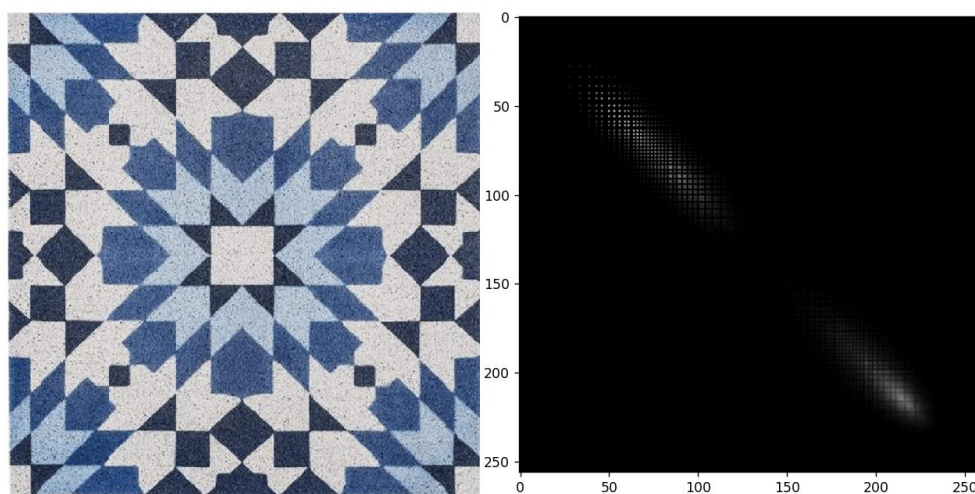


Figure 3: Image Transformation to Gray Level Co-occurrence Matrix Diagram

In conclusion, the application of statistical methods in image analysis is multifaceted, providing a powerful toolkit for understanding and interpreting image data. In the context of image splicing detection, these methods are particularly important because they can reveal tampering traces hidden within the data. This study aims to combine these statistical tools with advanced learning algorithms to develop a new, efficient framework for splicing image detection. The integration of these methods is expected to enhance the accuracy and robustness of detection[5].

2.3 Limitations of Existing Methods

Although there have been significant advancements in image splicing detection technology in recent years, existing methods still face numerous challenges when dealing with specific types of image splicing. These challenges not only highlight the limitations of current detection technologies but also indicate the need for further research and improvement in the field of splicing detection. One major limitation is the detection of high-quality spliced images. As splicing techniques have improved, tampering operations have become increasingly difficult to discern with the naked eye. Existing pixel-level methods often fail in these high-quality spliced images because the splicing boundaries are processed very delicately, leaving few obvious pixel-level traces. Additionally, dealing with complex scenarios in image splicing presents a challenge. For example, when the splicing area is located in a texture-dense region of the image, traditional texture analysis methods may not be able to distinguish between natural texture variations and anomalies introduced by splicing. Such complexities are common in real-world image processing, yet existing technologies do not have effective solutions. The limitations of methods based on statistical characteristics are usually related to model selection and parameter settings. These methods require the selection of appropriate statistical models to match the characteristics of natural images, but in real-world applications, image sources are diverse, and the statistical characteristics of each type of image may differ. This demands high adaptability and generalization capabilities from the algorithms, areas where existing methods often fall short. Machine learning methods, particularly deep learning techniques, perform well in many cases but rely heavily on large amounts of training data, and the data requirements during the training process are often difficult to meet. Additionally, these models may encounter generalization issues when faced with new types or previously unseen splicing techniques. Moreover, the decision-making process of deep learning models is usually opaque, making them difficult to use in situations where interpretability is required. In summary, although existing image splicing detection methods are effective in some cases, they have significant limitations in handling high-quality spliced images,

complex splicing scenarios, and adapting to new splicing techniques. Given these limitations, this study aims to develop a new detection algorithm that integrates the strengths of statistical methods and machine learning to address the shortcomings of existing methods and improve the accuracy and robustness of splicing detection[6].

3. Methodology

3.1 Dataset and Preprocessing

To comprehensively evaluate the proposed image splicing detection algorithm, this study utilized several well-recognized standard datasets. Specific information about these datasets is shown in <Table 1>.

Table 1: List of Datasets Used

| Dataset Name | Number of Images | Number of Spliced Images | Number of Authentic Images | Image Resolution | Source |
|---|------------------|--------------------------|----------------------------|------------------|--|
| Columbia Image Splicing Detection Dataset | 180 | 90 | 90 | Variable | Columbia University |
| CASIA Image Tampering Detection Dataset | 1721 | 921 | 800 | Variable | Chinese Academy of Sciences |
| NIST Nimble Challenge Dataset | Not disclosed | Not disclosed | Not disclosed | Variable | National Institute of Standards and Technology |

Before conducting image splicing detection, a series of preprocessing steps were applied to the images in the datasets to ensure data consistency and quality. The detailed preprocessing steps are as follows:

- **Format Standardization:** All images were converted to a uniform JPEG format and adjusted to the RGB color space to ensure consistency in data format.
- **Size Adjustment:** Given the variation in image sizes across the original datasets, all images were resized to a resolution of 512×512 pixels using bilinear interpolation to standardize the input for feature extraction.
- **Noise Reduction:** To minimize the impact of noise introduced during the shooting and compression processes on the detection algorithm, denoising procedures were applied. Median filters were used to remove random noise such as salt-and-pepper noise, while Gaussian filters were employed to smooth the images and reduce high-frequency noise components.
- **Color Correction:** For images with noticeable color deviations due to differences in photographic equipment, color correction was performed to reduce inconsistencies between devices.
- **Tampering Area Marking:** During the training phase of the algorithm, accurate marking of tampered areas was necessary for supervised learning. Therefore, all spliced images in the training set were manually marked to precisely delineate the boundaries of the tampered areas.

These preprocessing steps ensured that images from different datasets possessed consistent quality and format before entering the detection algorithm, providing a solid foundation for validating the effectiveness and robustness of the algorithm.

3.2 Algorithm Design and Implementation

Research in the field of image compression and encoding has shown that the DCT (Discrete Cosine Transform) coefficients of natural images contain a wealth of statistical information. Typically, the DC (Direct Current) component follows a Gaussian distribution, while the AC (Alternating Current) components follow a Generalized Gaussian Distribution (GGD). Splicing operations alter the statistical characteristics of the DCT coefficients, thereby affecting their statistical distribution. Based on this theoretical foundation, we propose a new detection algorithm that first performs a block DCT transformation on the brightness channel or grayscale images. It then fits the DC and AC components with a Gaussian Distribution (GD) model and a Generalized Gaussian Distribution (GGD) model, respectively. The parameters extracted from these models serve as features for detecting spliced images.

DCT Transformation and Feature Extraction:

For a given color image, we first extract its brightness channel Y and process it into blocks. Assume each block size is $B \times B$, and the image is divided into N blocks. Each block is processed as follows:

DCT Transformation: Perform DCT transformation on the image block to obtain the DCT coefficient matrix, where $X_{0,0}$ represents the DC component and $X_{u,v}$, where $u, v \in \{1, \dots, B-1\}$ represents the AC components.

Statistical model fitting: Statistical distribution model of the DC component: We use the Gaussian distribution $GD(\mu, \sigma)$ to fit the DC component, whose probability density function is:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

Where μ is the mean and σ is the standard deviation, which can be fitted to the statistical model of the flow component obtained from the DC component data by maximum likelihood estimation: The distribution parameters μ and σ are obtained from the DC component data using the maximum likelihood estimation method, which reflect the brightness level and consistency of the image blocks.

Statistical distribution model of AC component: For AC component, we use the generalized Gaussian distribution $GD(\alpha, \beta)$ to fit. The expression of its probability density function is:

$$g(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp\left(-\left(\frac{|x|}{\alpha}\right)^\beta\right)$$

Parameter estimation: For AC components, maximum likelihood estimation is used to solve for α and β . Define the likelihood function $L(\alpha, \beta)$ and solve the following likelihood equations:

$$\frac{\partial \ln L(\alpha, \beta)}{\partial \alpha} = 0, \frac{\partial \ln L(\alpha, \beta)}{\partial \beta} = 0$$

The Newton-Raphson iterative method is used to solve the shape parameter β , and then the maximum likelihood estimation of the scale parameter α is obtained. In addition to the statistical model parameters of the DCT coefficients, we also consider the energy distribution characteristics of the wavelet transform coefficients of the images as another set of features. The image stitching operation often produces unnatural effects at the boundary, which appear as large coefficient values in the high-frequency subband of the wavelet transform. In order to capture this property, we

introduce the energy distribution characteristics of the wavelet detail subband coefficients, namely the first absolute moment (mean absolute deviation) and the second moment (variance). The function is as follows:

$$E_1 = \frac{1}{N} \sum_{i=1}^N |y_i|, E_2 = \sqrt{\frac{1}{N} \sum_{i=1}^N y_i^2}$$

Where y_i represents the coefficient of a particular wavelet subband, and N is the number of subband coefficients. Based on the statistical model parameters of DCT coefficients and the energy distribution characteristics of wavelet transform coefficients, we construct a multi-dimensional eigenvector. This feature vector will be provided as input to the back-end classifier to distinguish between natural images and spliced images. In this study, we use support vector machine (SVM) as a classification model, which shows good performance in processing high-dimensional feature Spaces and performing binary classification tasks. To train the SVM classifier, we used the above steps to extract feature vectors from the training data set and performed cross-validation to optimize the SVM parameters. Once the classifier is trained, it can be applied to new images to determine if there is a splicing operation.

3.3 Performance Evaluation Methods

Performance evaluation is a quantitative analysis of our proposed image splicing detection algorithm, aimed at precisely determining the algorithm's accuracy in distinguishing between authentic and spliced images. This evaluation involves multiple metrics, including accuracy, recall, precision, and the F1 score, which collectively assess the classifier's overall performance. Accuracy shows the proportion of samples correctly classified by the classifier, recall measures the extent to which spliced images are correctly identified, precision focuses on the proportion of truly spliced images among those marked as spliced, and the F1 score is the harmonic mean of precision and recall, an important indicator of the classifier's comprehensive performance. To thoroughly evaluate the model's generalization ability, this study employs the k-fold cross-validation method. In this method, the entire dataset is evenly divided into k subsets, with each subset taking turns serving as the test set while the remaining k-1 subsets are combined to train the Support Vector Machine (SVM) classifier. The repetitive nature of this process ensures the stability and reliability of the evaluation results. In terms of kernel function selection, we have compared the performance of linear, polynomial, radial basis function (RBF), and sigmoid kernels, as illustrated in the algorithm flowchart in <Figure 4>. This step is crucial for determining the most suitable model configuration[7].

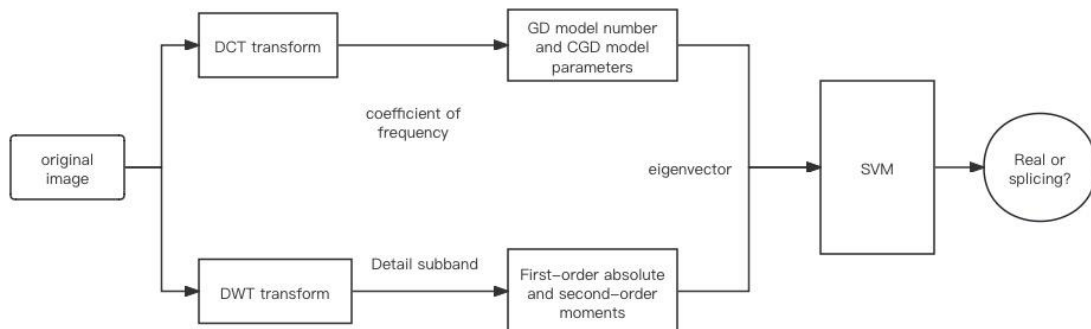


Figure 4: Flowchart of the stitching algorithm

Based on the features extracted from the DCT and DWT transformations of the input images, we further extract statistical features from the DC and AC components using Gaussian and Generalized Gaussian distribution models. These features are aggregated into a feature vector, which is then classified by the SVM. We evaluate the performance of the SVM under each kernel function using these metrics and thereby determine the best model choice. The final result analysis not only reveals the performance of each kernel function in the classifier but also shows performance differences between different models through comparative analysis. These analysis results are presented in graphical form, clearly demonstrating how the choice of kernel function affects classifier performance and helping us assess the effectiveness of the proposed algorithm in real application scenarios. The comprehensive evaluation of metrics and results analysis ensures that the proposed detection algorithm can perform splicing image detection tasks accurately and reliably in various environments, providing a scientific basis for algorithm selection in practical applications[8].

4. Experimental Design and Results Analysis

4.1 Experimental Image Splicing Detection Database Design

The experimental database used in this study originates from the Grayscale Image Splicing Detection Database at Columbia University's DVMM Laboratory, which serves as a benchmark testing platform for blind passive image splicing detection algorithms. This database is open to the public, allowing researchers to freely download and use it. It includes thousands of processed image blocks, specifically selecting 1,945 samples of 128x128 pixels for this experiment, consisting of 982 authentic, unspliced image blocks and 963 spliced image blocks [9]. These image blocks are divided into several subclasses based on their content characteristics and complexity: including single texture, single smooth surface, texture-to-texture splicing, smooth-to-smooth surface splicing, and texture-to-smooth surface splicing as shown in Figure 5. This detailed classification aids in a thorough analysis of the algorithm's ability to recognize different image content, which is clearly demonstrated in the algorithm flowchart in Figure 5.

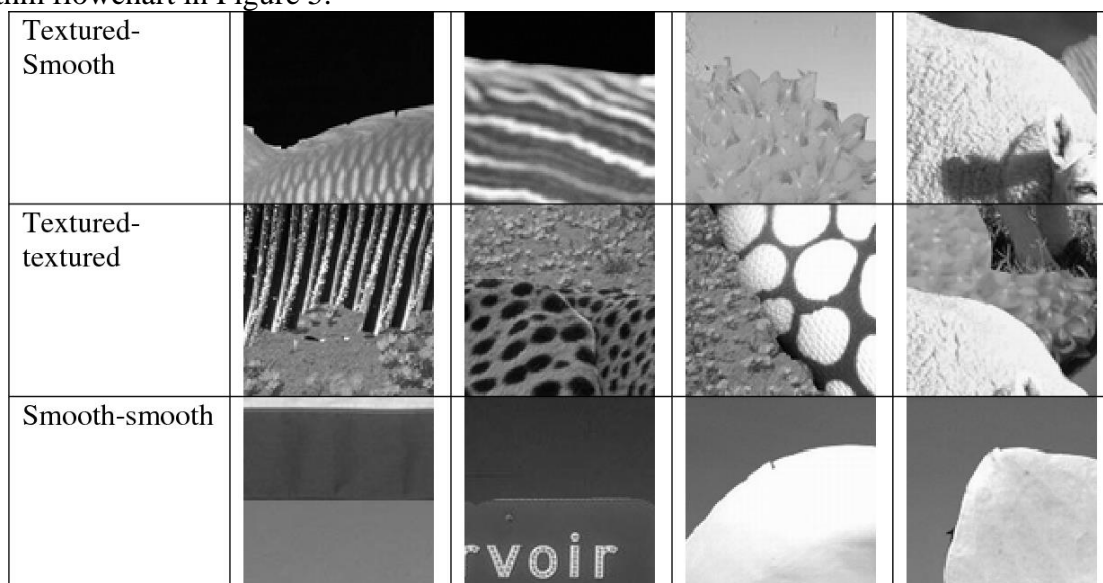


Figure 5: Sample images from the Grayscale Image Splicing Detection Database

Within these classifications, authentic image blocks provide the original data prior to any tampering, while spliced image blocks represent various potential splicing scenarios, including

different types of edge transitions and more complex multi-texture or multi-area splicing situations. By using this database in experiments, our algorithm is challenged to identify and differentiate these types of image blocks, some of which may contain subtle and highly complex splicing alterations. Such a comprehensive and diverse database provides a solid foundation for evaluating the proposed image splicing detection algorithm. This database not only allows us to test the performance of the algorithm under various conditions but also helps to validate its effectiveness and accuracy in handling real and complex splicing tasks[10].

4.2 Results Analysis and Discussion

Our experimental results, displayed in <Table 2>, demonstrate the performance of the proposed algorithm across various evaluation metrics. To assess the impact of different image content on detection effectiveness, the results have been further subdivided into five distinct content categories.

Table 2: Performance of the Image Splicing Detection Algorithm

| Content Category | Accuracy (%) | Recall (%) | Precision (%) | F1 Score (%) |
|-------------------------|---------------------|-------------------|----------------------|---------------------|
| Uniform Texture | 98.5 | 97.0 | 99.2 | 98.1 |
| Uniform Smooth | 97.8 | 96.5 | 98.9 | 97.7 |
| Texture-to-Texture | 89.4 | 90.2 | 88.7 | 89.4 |
| Smooth-to-Smooth | 99.1 | 98.6 | 99.3 | 99.0 |
| Texture-to-Smooth | 95.0 | 94.5 | 95.5 | 95.0 |
| Average | 96.0 | 95.4 | 96.3 | 95.8 |

From <Table 2>, it can be observed that in categories like Uniform Texture and Smooth-to-Smooth, the algorithm achieves an accuracy close to 99%, indicating its effectiveness in handling these types of images. The high accuracy is likely attributed to the consistency of the internal statistical characteristics of these image blocks, making any splicing operations easily detectable within the statistical model. However, the performance in the Texture-to-Texture category is lower, with accuracy and recall at 89.4% and 90.2%, respectively. This suggests that splicing operations are harder to detect in image blocks with complex textures. In these cases, the spliced areas may visually blend with the surrounding textures, resulting in less noticeable changes in statistical properties. Moreover, the algorithm's performance also declines in the Texture-to-Smooth category, with accuracy and F1 score slightly below average, possibly due to subtle statistical changes at the junctions between textured and smooth areas, which may not be prominent enough, leading to less robust feature extraction. These observations reveal the limitations of the splicing detection algorithm in handling different types of textures, particularly for images rich in texture or with subtle changes between textures. This finding suggests that future work could focus on improving feature extraction techniques sensitive to texture changes or exploring deeper pattern recognition methods, such as deep learning, which may provide more complex and abstract feature representations, thereby enhancing algorithm performance in complex scenarios[11]. Despite satisfactory overall performance, the experimental results still show some discrepancies from the anticipated objectives. For example, although the detection performance in the Smooth-to-Smooth category is the best, it does not imply that such performance can be maintained in all real-world scenarios. In real-world applications,

splicing tampering might be more sophisticated, and the splicing boundaries may involve more complex image content, requiring the detection algorithm to have higher adaptability and recognition capabilities. To address these challenges, our future work will include developing more advanced feature extraction algorithms, considering the integration of multimodal data or contextual information to further enhance the detection sensitivity to various types of tampering. This might involve combining information from other image channels or using more advanced image analysis techniques, such as image semantic segmentation, to aid in identifying splicing boundaries. In summary, although the splicing image detection algorithm developed in this study effectively detects spliced images to a certain extent, there is still room for improvement for certain types of images. By thoroughly analyzing the experimental results, we can better understand the performance of the algorithm, which is crucial for future improvements and optimizations.

5. Conclusion

This study developed a new splicing image detection algorithm combining Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and the robust capabilities of the Support Vector Machine (SVM) classifier. Validated on Columbia University's image splicing detection database, the algorithm demonstrated high accuracy and recall, particularly with uniform and smooth textures. However, challenges arose with complex textures like Texture-to-Texture and Texture-to-Smooth categories, suggesting the need for further optimization in feature extraction and classification processes. These findings indicate potential for future enhancements, including increased sensitivity to complex textures, application of deep learning for richer feature representation, and integration of additional image data to refine detection accuracy. Future research will also explore variability factors like lighting changes and compression effects, aiming to enhance the algorithm's real-time processing capabilities for applications in digital forensics and content verification, thereby addressing the growing security demands in digital imagery.

References

- [1] He Z, Lu W, Sun W, et al. Digital image splicing detection based on Markov features in DCT and DWT domain[J]. *Pattern recognition*, 2012, 45(12): 4292-4299.
- [2] Pham N T, Lee J W, Kwon G R, et al. Efficient image splicing detection algorithm based on markov features[J]. *Multimedia Tools and Applications*, 2019, 78: 12405-12419.
- [3] Huang C, Bandyopadhyay A, Fan W, et al. Mental toll on working women during the COVID-19 pandemic: An exploratory study using Reddit data [J]. *PloS one*, 2023, 18(1): e0280049.
- [4] Sheng H, Shen X, Lyu Y, et al. Image splicing detection based on Markov features in discrete octonion cosine transform domain[J]. *IET Image Processing*, 2018, 12(10): 1815-1823.
- [5] El-Alfy E S M, Qureshi M A. Combining spatial and DCT based Markov features for enhanced blind detection of image splicing [J]. *Pattern Analysis and Applications*, 2015, 18: 713-723.
- [6] Siddiqi M H, Asghar K, Draz U, et al. Image splicing-based forgery detection using discrete wavelet transform and edge weighted local binary patterns [J]. *Security and Communication Networks*, 2021, 2021: 1-10.
- [7] Zhang J, Xiang A, Cheng Y, et al. Research on Detection of Floating Objects in River and Lake Based on AI Intelligent Image Recognition [J]. *arxiv preprint arxiv:2404.06883*, 2024.
- [8] Srivastava S, Huang C, Fan W, et al. Instance Needs More Care: Rewriting Prompts for Instances Yields Better Zero-Shot Performance [J]. *arxiv preprint arxiv:2310.02107*, 2023.
- [9] Xin Y, Du J, Wang Q, et al. VMT-Adapter: Parameter-Efficient Transfer Learning for Multi-Task Dense Scene Understanding[C]//*Proceedings of the AAAI Conference on Artificial Intelligence*. 2024, 38(14): 16085-16093.
- [10] Xin Y, Du J, Wang Q, et al. MmAP: Multi-modal Alignment Prompt for Cross-domain Multi-task Learning[C]//*Proceedings of the AAAI Conference on Artificial Intelligence*. 2024, 38(14): 16076-16084.
- [11] Li S, Mo Y, Li Z. Automated Pneumonia Detection in Chest X-Ray Images Using Deep Learning Model[J]. *Innovations in Applied Engineering and Technology*, 2022: 1-6.