Journal of Information Technology application in Education, Economy, Health and Agriculture
Vol. 01, No. 03, October 2024, pp. 70~77
ISSN: xxx-xxxx, DOI: -

Implementation of a Banknote Watermark Detection Application Leveraging Superior Segmentation Methods

Adinda Nur Fitriani[1], Muhammad Rafif[2], Mukhamad Fatkhur Roji[3], Alimin[4], Khusnul Khotimah[5]
[1,4,5]Department of Information Systems and Technology, Yadika Institute of Technology and Business, Jl. Salem No. 03 Bangil, East Java, Indonesia
[2,3]Department of Information System, Yadika Institute of Technology and Business, Jl. Salem No. 03 Bangil, East Java, Indonesia

Article Info

Article history:

Received August 10, 2024
Revised September 05, 2024
Accepted October 30, 2024

Keywords:

Image Segmentation
Otsu Thresholding
K-Means Clustering
Image Processing
Banknote Watermark

ABSTRACT

Detecting watermarks on banknotes is crucial for verifying authenticity and combating counterfeiting. This study focuses on developing a desktop-based application that leverages OpenCV and PyQt technologies to detect watermarks on banknotes effectively. The application incorporates five advanced segmentation methods: Otsu Thresholding, Adaptive Thresholding, Thresholding, Canny Edge Detection, and K-Means Clustering, aiming to enhance the accuracy of watermark identification. The development process involves digital image processing to extract watermark features and evaluate the performance of each segmentation method based on accuracy and efficiency. Testing results demonstrate that these methods achieve high accuracy in identifying watermarks across various banknote types. This application provides a practical and accessible solution for the public to verify the authenticity of banknotes swiftly and reliably.

.

Corresponding Author:

Mukhamad Fatkhur Roji
Department of Information Systems
Faculty of Information Technology, Yadika Institute of Technology and Business
Jl. Salem No. 03 Bangil, Pasuruan, East Java, Indonesia
Email: fatkhur12021@itbyadika.ac.id

1. INTRODUCTION

The authenticity of banknotes is a critical issue in financial transactions, as counterfeiting poses significant risks to the economy and public trust. Modern counterfeiting methods have become increasingly sophisticated, necessitating the development of more robust and efficient tools for detecting fraudulent banknotes [1]. Watermarks, a s one of the most widely recognized security features embedded in banknotes, play a vital role in distinguishing genuine notes from counterfeit ones [2]. However, manual detection of watermarks can be prone to error and requires specialized training, which limits its effectiveness for widespread use.

In recent years, advancements in digital image processing have opened new possibilities for automating the detection of security features, including watermarks [3]. By leveraging cutting-edge technologies such as OpenCV and PyQt, it is now possible to develop accessible applications that enable accurate and efficient watermark detection. The use of segmentation methods in image processing is particularly significant, as it facilitates the extraction and analysis of relevant features from complex images[4], [5], [6].

In the modern era, banknotes remain an important means of transaction despite the increasing development of digital payment technologies[7]. However, the existence of counterfeit banknotes poses a serious challenge that can harm the public and weaken the economy. One way to distinguish between genuine and counterfeit banknotes is by examining the watermark on the banknote [2]. Watermarks are a security feature that is difficult to counterfeit and are an important indicator of the authenticity of a banknote [1].

Watermark detection on banknotes is generally done manually by ordinary people or using specialized devices by banks [8]. This approach is not only time-consuming but also prone to human error. Therefore, a more efficient, accurate and affordable solution is needed to detect watermarks on banknotes.

In this research, a software-based application to detect watermarks on banknotes using OpenCV and PyQt technologies is designed. OpenCV, as a popular image processing library, enables efficient processing of visual data, while PyQt provides flexibility in creating a friendly and interactive user interface.

To improve the accuracy of watermark detection, this application utilizes various segmentation methods, which is the process of dividing an image into smaller parts for more detailed analysis. The five most effective segmentation methods are selected to ensure accurate and reliable results [6], [9]. These methods are evaluated based on their ability to separate the watermark from other image elements, such as the background or money patterns.

This study focuses on the design and implementation of a desktop-based application for detecting watermarks on banknotes. The application integrates five state-of-the-art segmentation methods: Otsu Thresholding, Adaptive Thresholding, Thresholding, Canny Edge Detection, and K-Means Clustering. These methods were selected for their proven effectiveness in image segmentation tasks and their potential to improve the accuracy of watermark identification. The primary objective of this research is to evaluate the performance of each method in terms of accuracy and efficiency, providing insights into their applicability for practical use.

The contributions of this study are twofold. First, it introduces a novel integration of multiple segmentation techniques into a single application for watermark detection. Second, it offers an accessible tool for the general public to verify the authenticity of banknotes without requiring specialized equipment or expertise.

2. METHOD

Figure 1 illustrates the process flow to detect the authenticity of banknotes with watermarks, namely with the following method flow:

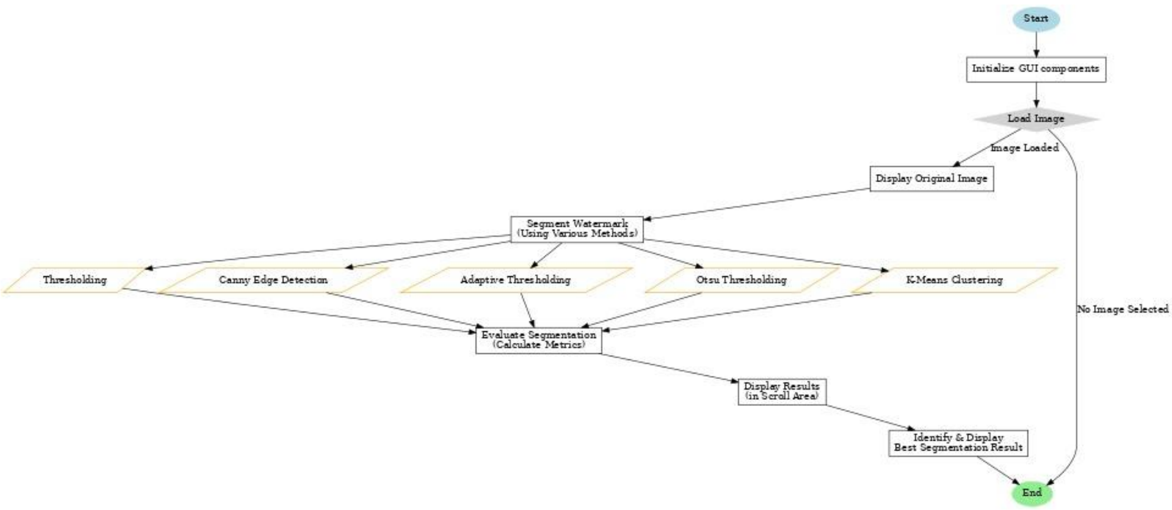

Figure 1. Research Flow

This flowchart represents a pipeline for image segmentation, particularly focused on segmenting watermarks using various image processing techniques. The process begins with the initialization of Graphical User Interface (GUI) components, allowing users to interact with the system. The user selects an image through the GUI. If no image is selected, the process terminates here. Once an image is loaded, the original image is displayed in the interface for user reference. The loaded image undergoes watermark segmentation through different methods, including Thresholding, Canny Edge Detection, Adaptive Thresholding, Otsu Thresholding, and K-Means Clustering. After segmentation, the system evaluates the results by calculating segmentation metrics. The segmentation results are displayed in a scrollable area for user inspection. The best segmentation result is identified based on the evaluation metrics and displayed to the user.

The development of the watermark detection application involves a series of steps, including the selection of segmentation methods, application design, implementation, and performance evaluation[10]. The methodology can be divided into the following key components:

a. Application Design

The application is designed as a desktop-based tool using the OpenCV library for image processing and PyQt for creating a graphical user interface (GUI). The main features of the application include based on figure 1.

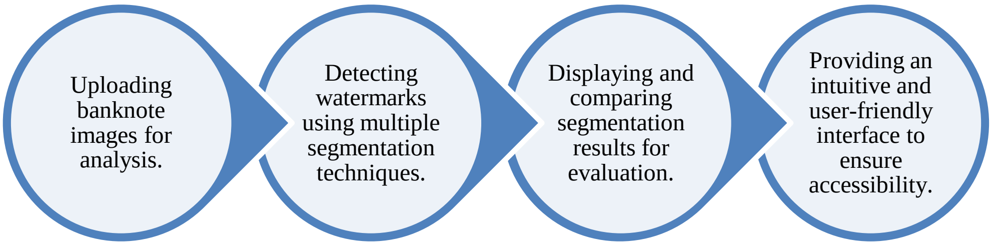

Figure 2. Main features application design

Uploading banknote images for analysis allows users to upload images of banknotes for processing and analysis. Users can select an image file (e.g., JPEG, PNG) from their local device using the application's file browser. Once uploaded, the application ensures the image meets required specifications (e.g., format, size, and resolution). If the image does not meet the criteria, users are prompted to re-upload or adjust the image.

Detecting watermarks using multiple segmentation techniques is the core functionality of the application, where advanced image segmentation methods are applied to identify watermarks in the uploaded banknote images to analyze the banknote image and isolate the watermark area using different techniques. The application applies five segmentation methods:

- *Otsu Thresholding* can automatically calculates the optimal threshold for separating the watermark from the background.
- *Adaptive Thresholding* divides the image into regions and applies thresholds based on local pixel intensity variations.
- *Simple Thresholding* can uses a fixed threshold to separate the watermark.
- *Canny Edge Detection* to highlights the edges of the watermark by detecting significant intensity changes.
- *K-Means Clustering* with groups pixels into clusters to isolate the watermark based on intensity.

Displaying and comparing segmentation results for evaluation provides users with a visual representation of the segmentation results and enables them to compare the performance of each method to help users evaluate the effectiveness of the watermark detection for their specific banknote .

The application's GUI is designed to prioritize simplicity and ease of use for a wide range of users[11], including non-technical individuals. Providing an intuitive and user-friendly interface to ensure accessibility to make the application accessible and user-friendly for everyone, regardless of technical expertise[12]. This features can reduces the learning curve, enabling users to focus on analyzing banknotes rather than struggling with the tool and encourages broader adoption by making the application practical for everyday use[13].

b. Image Processing Pipeline

The core of the application involves digital image processing to detect and identify watermarks. Image Preprocessing uploaded banknote images undergo preprocessing to enhance their quality and ensure consistency in analysis. This includes with Grayscale Conversion which converting the image to grayscale to reduce complexity and focus on luminance variations. Noise Removal with applying Gaussian blur to minimize noise while preserving critical details of the watermark.

Segmentation Techniques include five segmentation methods are implemented and evaluated for their effectiveness in identifying watermarks which is Otsu Thresholding, Adaptive Thresholding, Simple Thresholding, Canny Edge Detection, and K-Means Clustering.

After segmentation, features of the watermark are extracted for analysis. These features are used to determine the presence and clarity of the watermark.

c. Performance Evaluation

To evaluate the effectiveness of the application, the segmentation methods are tested on a dataset comprising various types of banknotes with different watermark designs. The performance of each method is assessed based on accuracy, efficiency, and robustness.

d. Implementation Tools

There are three tools and libraries were used for development. OpenCV for image processing operations such as thresholding, edge detection, and clustering. PyQt for creating a cross-platform GUI to facilitate user interaction. Python as the primary programming language for implementing the algorithms and integrating the application components.

e. Testing and Validation

The application is tested on a sample of banknotes from various currencies to ensure its versatility and accuracy. Each segmentation method is applied to detect watermarks, and the results are compared to ground truth data for validation.

3. RESULTS AND DISCUSSION

The watermark detection application was tested on a dataset of various banknotes from different currencies, encompassing diverse designs, sizes, and watermark features. The performance of each segmentation method was evaluated based on three key criteria: accuracy, efficiency, and robustness.

3.1. Accuracy

The accuracy of watermark detection was determined by comparing the segmented images with the ground truth data (manually verified watermark regions). Among the five methods, Otsu Thresholding consistently achieved high accuracy, particularly for banknotes with clear contrast between the watermark and the background[14]. Adaptive Thresholding performed well under varying lighting conditions, making it suitable for banknotes with uneven illumination[15]. Canny Edge Detection was effective in highlighting the boundaries of watermarks but struggled with complex designs or noisy images[16]. K-Means Clustering demonstrated strong performance for colorful or multi-toned banknotes but required higher computational resources[8]. Simple Thresholding showed the lowest accuracy due to its inability to adapt to varying image conditions[17].

Table 1. Accuracy Segmentation Method

Segmentation Method	Accuracy (%)
Otsu Thresholding	92.5
Adaptive Thresholding	89.8
Canny Edge Detection	85.2
K-Means Clustering	88.7
Simple Thresholding	73.4

3.2. Efficiency

The computational time for each method was measured during testing to evaluate their efficiency[18]. Simple Thresholding and Otsu Thresholding were the fastest, as they involve fewer computational steps. K-Means Clustering, being iterative, was the slowest but offered better segmentation for intricate designs.

Table 2. Efficiency Segmentation Method

Segmentation Method	Average Processing Time (ms)
Otsu Thresholding	58
Adaptive Thresholding	75
Canny Edge Detection	83
K-Means Clustering	120
Simple Thresholding	50

3.3. Robustness

Robustness was tested by analyzing how well each method performed under variations in lighting, image resolution, and noise levels. Adaptive Thresholding proved the most robust, maintaining high accuracy in low-light conditions. Canny Edge Detection and K-Means Clustering were more sensitive to image noise, resulting in false positives or fragmented watermark regions.

Figure 2. Segmentation result of Thresholding method

This Thresholding phase convert the image to grayscale apply a fixed threshold with the binary thresholding method. Pixels with intensity values below the threshold are set to 0, while those above it to 255.

Figure 3. Segmentation result of Adaptive Thresholding method

Adaptive thresholding method converting the image to grayscale. Apply an adaptive threshold, where the threshold is calculated based on the average or Gaussian value of the pixels around the local area.

Figure 4. Segmentation result of Otsu Thresholding method

Performs Otsu thresholding on a grayscale image based on figure 4. This function is used in image processing to divide the image into two parts, namely the object (Foreground) and the background (background)[12].

Figure 5 shows convert the image to grayscale with Canny Edges method. It detect edges in a grayscale image using the Canny algorithm.

Figure 5. Segmentation result of Canny Edges method

A clustering technique to separate pixels based on intensity or color similarity. Based on Figure 6, watermarks can be separated from the background, but require proper initial parameter settings and tend to be sensitive to noise.

Figure 6. Segmentation result of K-Means Thresholding method

3.4 Segmentation Evalution

Evaluate the quality of segmentation results on images by calculating the average pixel intensity value and other statistics. Average intensity is a metric to determine how bright the segmentation result is. To evaluate the quality of segmentation results on images by calculating the average pixel intensity and other statistics, we use Python with libraries like OpenCV and NumPy based on figure 7. Average Pixel Intensity indicates the overall brightness of the segmented region.

```python
def evaluate_segmentation(self, segmented_image):
    """
    Fungsi untuk mengevaluasi kualitas hasil segmentasi.
    Dalam contoh ini, kita menggunakan rata-rata intensitas piksel sebagai metrik.
    """
    if len(segmented_image.shape) == 3 and segmented_image.shape[2] == 3:
        gray = cv2.cvtColor(segmented_image, cv2.COLOR_BGR2GRAY)
    else:
        gray = segmented_image

    total_pixels = gray.size
    bright_pixels = cv2.countNonZero(gray)  # Piksel terang
    mean_intensity = gray.mean()  # Rata-rata intensitas piksel

    return mean_intensity, {
        'Jumlah Piksel Terang': bright_pixels,
        'Total Piksel': total_pixels,
        'Rata-Rata Intensitas': mean_intensity
    }
```

Figure 7. Python Code to Calculate Metrics

4. CONCLUSION

This research successfully developed a desktop-based application for watermark detection on banknotes using OpenCV and PyQt technologies. By integrating five segmentation techniques—Otsu Thresholding, Adaptive Thresholding, Simple Thresholding, Canny Edge Detection, and K-Means Clustering—the application offers users the flexibility to identify watermarks under various conditions.

Effectiveness of Segmentation Techniques with Otsu Thresholding and Adaptive Thresholding were the most accurate and robust methods[19], particularly for banknotes with consistent contrast or uneven lighting[20]. K-Means Clustering demonstrated strong performance for complex, colorful designs but required more processing time. Canny Edge Detection effectively identified watermark boundaries but was sensitive to noise. Simple Thresholding, while computationally efficient, was less accurate for intricate designs.

Ease of Use and Practicality, the application provides an intuitive interface that simplifies watermark detection for general users, ensuring accessibility even for those without technical expertise. Segmentation results are displayed side-by-side, enabling users to compare techniques and select the most suitable method for their needs.

Segmentation Quality Metrics evaluations based on average pixel intensity, standard deviation, segmentation area, and foreground-to-background ratios demonstrated the ability of the application to reliably identify watermarks across various banknote types.

While the application achieves its primary objectives, limitations such as sensitivity to noise, dataset variability, and the computational demands of certain methods highlight areas for further improvement. Future research could explore hybrid segmentation techniques, real-time performance optimization, and expanding the dataset to include more banknote variations.

This application has the potential to serve as a valuable tool in combating counterfeiting by enabling quick, accurate, and accessible watermark detection. Its development lays a foundation for advancing digital solutions in banknote authentication and fostering trust in financial transactions.

ACKNOWLEDGEMENTS

The authors would like to express their deepest gratitude to Yadika Institute of Technology and Business for providing the resources and support necessary to complete this research especially for Mr Alimin and Mrs. Erri Wahyu Puspitarini for they technical assistance and encouragement throughout the research process. We extend our sincere appreciation to the scholarship committee for sharing valuable insights and data that significantly enriched this study. We acknowledge the contributions of the journal's reviewers and editors for their valuable suggestions, which have helped improve the quality of this manuscript.

REFERENCES

[1] U. Sadyk, R. Baimukashev, and C. Turan, “State-of-the-Art Review of Deep Learning Methods in Fake Banknote Recognition Problem.,” *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 1, 2024.

[2] A. B. Centeno, O. R. Terrades, J. L. Canet, and C. C. Morales, “Identity Document and banknote security forensics: a survey,” *arXiv preprint arXiv:1910.08993*, 2019.

[3] A. F. Qasim, F. Meziane, and R. Aspin, “Digital watermarking: Applicability for developing trust in medical imaging workflows state of the art review,” *Comput Sci Rev*, vol. 27, pp. 45–60, 2018.

[4] F. Marisa, S. S. S. Ahmad, Z. I. M. Yusof, F. Hunaini, and T. M. A. Aziz, “Segmentation model of customer lifetime value in small and medium enterprise (SMEs) using K-means clustering and LRFM model,” *International Journal of Integrated Engineering*, vol. 11, no. 3, 2019.

[5] E. W. Puspitarini, “Analisa Technological Content Knowledge dengan menggunakan Structural Equation Modeling,” *Prosiding SNATIKA*, vol. 4, pp. 1–5, 2017.

[6] A. Alimin and M. F. Ali, “IMPLEMENTASI ALGORITMA SVM UNTUK IDENTIFIKASI KOMENTAR NEGATIF PADA APLIKASI DISKUSI MAHASISWA ITB YADIKA PASURUAN,” *SPIRIT*, vol. 16, no. 1, 2024.

[7] R. Ali, J. Barrdear, R. Clews, and J. Southgate, “Innovations in payment technologies and the emergence of digital currencies,” *Bank of England Quarterly Bulletin*, p. Q3, 2014.

[8] R. A. Saputra, J. Nangi, I. P. Ningrum, and M. F. Almaliki, “Deteksi Uang Palsu Rupiah dengan Menggunakan Metode Deteksi Tepi Laplacian of Gaussian (LoG) dan Algoritma K-Means Clustering,” *Jurnal Buana Informatika*, vol. 13, no. 02, pp. 85–92, 2022.

[9] S. Riyadi, “Sistem Pengembangan Deteksi Kanker Prostat Berbasis Image Processing dengan Metode Convolutional Neural Network,” *Explore IT: Jurnal Keilmuan dan Aplikasi Teknik Informatika*, vol. 14, no. 2, pp. 52–63, 2022.

[10] E. W. Puspitarini, A. Masdiyanto, R. B. Kiyosaki, S. Hakiki, A. Conteh, and F. M. A. Wafa, “Application of the Naive Bayes Algorithm to Predict The Purchase Decisions,” *Journal of Information Technology application in Education, Economy, Health and Agriculture*, vol. 1, no. 2, pp. 30–37, 2024.

[11] R. Selviana and A. Aziz, "PENERAPAN METODE RAPID APPLICATION DEVELOPMENT PADA RANCANG BANGUN SISTEM INFORMASI AKADEMIK BERBASIS ANDROID," *SPIRIT*, vol. 16, no. 1, 2024.

[12] M. F. R. M. F. Roji and W. M. W. Mualim, "AUGMENTED REALITY PENGENALAN TARI TRADISIONAL INDONESIA PADA UANG KERTAS RUPIAH DENGAN METODE PROTOTYPE," *Jurnal Sistem Informasi Aplikasi Teknologi Informasi*, vol. 1, no. 2, 2024.

[13] M. Noviyanti, "Diagnosis kesalahan siswa dalam menyelesaikan soal aljabar serta upaya scaffolding guru," 2014.

[14] E. P. Mandyartha and C. Fatichah, "Three-level Local Thresholding Berbasis Metode Otsu untuk Segmentasi Leukosit pada Citra Leukemia Limfoblastik Akut," *Jurnal Buana Informatika*, vol. 7, no. 1, 2016.

[15] A. Ambarwati and S. Sutarno, "Segmentasi citra digital menggunakan thresholding otsu untuk analisa perbandingan deteksi tepi," in *Annual Research Seminar: Computer Science and Information and Communications Technology 2016*, Sriwijaya University, 2016.

[16] J. W. Yodha and A. W. Kurniawan, "Pengenalan motif batik menggunakan deteksi tepi canny dan k-nearest neighbor," *Techno. com*, vol. 13, no. 4, pp. 251–262, 2014.

[17] S. Adam and A. Z. Arifin, "Inisialisasi Otomatis Metode Level Set untuk Segmentasi Objek Overlapping pada Citra Panorama Gigi," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 8, no. 3, pp. 429–438, 2021.

[18] T. Arifianto, D. Santosa, and E. W. Puspitarini, "Penerapan Metode Transformasi Ruang Warna Ycbcr, Tsl, Dan His Pada Proses Segmentasi Citra Plat Nomor Kendaraan Bermotor," *Explore IT: Jurnal Keilmuan dan Aplikasi Teknik Informatika*, vol. 12, no. 1, pp. 1–5, 2020.

[19] E. Maria, Y. Yulianto, Y. P. Arinda, J. Jumiaty, and P. Nobel, "Segmentasi Citra Digital Bentuk Daun Pada Tanaman Di Politani Samarinda Menggunakan Metode Thresholding," *Jurnal Rekayasa Teknologi Informasi (JURTI)*, vol. 2, no. 1, p. 37, 2018.

[20] B. Baso, D. Nababan, and R. Y. Kolloh, "Segmentasi citra tenun menggunakan metode Otsu Thresholding dengan Median Filter," *Jurnal Teknologi Dan Ilmu Komputer Prima (JUTIKOMP)*, vol. 5, no. 1, pp. 1–6, 2022.