Instal : Jurnal Komputer

E-ISSN: 2808-683X
Edisi : Volume 16 Nomor 05 | December 2024

Available online at https://journalinstal.cattleyadf.org/index.php/Instal/index

Image Processing Application for Detecting Computer Answer Sheets Using the Canny Method

Ilka Zufria[1], Heri Santoso[2], Muhafiz Aditio Prabowo[3]
[1,2,3]Universitas Islam Negeri Sumatera Utara

ABSTRACT

The high costs associated with specialized scanning devices for computer-based answer sheets (LJK) present significant barriers for educational institutions, especially those in underfunded or rural areas. This research introduces an application utilizing the Canny edge detection algorithm to process and evaluate LJK responses efficiently and accurately. The proposed system employs advanced image processing techniques to detect marked answers and automate scoring based on a pre-configured answer key, significantly reducing manual errors. Python programming was used to develop the application, leveraging its robust libraries and flexibility. The system demonstrated reliable edge detection and efficient response evaluation in various testing scenarios. This approach provides a scalable and cost-effective alternative to manual grading systems or expensive scanning devices, potentially transforming how assessments are handled in educational institutions.

Keywords:

Image Processing
Canny Edge Detection
Computer Answer Sheet Detection
Python

Corresponding Author:
Muhafiz Aditio Prabowo
Universitas Islam Negeri Sumatera Utara
Email: muhafizaditio@gmail.com

INTRODUCTION

The assessment of student performance often involves exams using Computer Answer Sheets (LJK). These tools facilitate the processing of multiple-choice tests but require specialized scanning devices, which are costly and inaccessible for many schools. According to Rifandi [1], evaluating learning outcomes through such tools can significantly enhance the efficiency of the assessment process. However, the manual correction of LJK is still prevalent in many institutions, often leading to errors and inefficiencies.

With the advent of digital image processing, algorithms like Canny Edge Detection offer an affordable solution for analyzing scanned LJK images. Canny's algorithm, introduced in 1986, is recognized as one of the most effective edge detection techniques, offering minimal error rates, precise localization, and efficient performance [2]. This makes it particularly suitable for processing complex patterns, such as those found on LJK.

This research seeks to leverage the Canny algorithm to automate the grading process of LJK, reducing costs while improving accuracy and efficiency. By converting scanned LJK images into edge-detected outputs, the system identifies marked answers and compares them against a pre-configured answer key. The approach not only eliminates the need for costly scanners but also ensures scalability across different educational settings.
This paper aims to design and implement an application capable of detecting student responses on LJK using standard scanning devices and the Canny algorithm. Such a solution provides a streamlined and cost-effective alternative for educational institutions, especially those with limited resources. The implementation aligns with prior studies, such as those by Munantri et al. [3] and Triyana [4], which demonstrated the feasibility of image processing techniques in educational assessments.

METHODS

This study adopts a waterfall methodology, which is a systematic approach characterized by a sequential progression through distinct phases: planning, design, implementation, testing, and evaluation. The methodology ensures each stage is completed before the next begins, facilitating a clear and organized development process. This structured approach aligns well with the needs of this research, which involves both software development and iterative testing to ensure accuracy in LJK processing.

Tools and Materials

The tools and materials employed in this study are categorized as follows:
Hardware: The primary hardware includes an Acer Aspire E14 laptop equipped with an Intel processor and 4GB RAM, sufficient for running Python-based image processing tasks, and an HP Scanjet N6350 scanner for acquiring high-resolution digital images of the LJK forms.
Software: Key software tools include the Python programming environment, leveraging libraries such as OpenCV for image processing, NumPy for numerical computations, and Matplotlib for visualization. The system operates on Windows 10, providing compatibility with the hardware and software tools.
Materials: Standardized LJK forms, pre-configured answer keys, and writing utensils were utilized to simulate real-world test scenarios. These materials were essential for ensuring the system's accuracy and relevance in practical applications.

Workflow

The development of the application followed a series of well-defined steps, as outlined below:
Image Acquisition: LJK forms were scanned using the HP Scanjet N6350 scanner, ensuring consistent resolution and alignment for all samples. The scanned images were stored in JPEG format to maintain quality while minimizing file size.

1. Preprocessing: To enhance processing accuracy, the RGB images were converted to grayscale, reducing computational complexity by eliminating color

information. Additionally, Gaussian filtering was applied to minimize noise and smooth the images before edge detection.

2. Edge Detection: The Canny algorithm was implemented to detect edges on the grayscale images. This process involved calculating gradients, applying non-maximum suppression to refine edge localization, and using double thresholding and hysteresis to produce a binary edge map.
3. Answer Comparison: Detected responses were compared to the pre-configured answer key. This step involved mapping the edge-detected marks to predefined answer regions on the LJK form, ensuring accurate detection of student responses.
4. Scoring: The system calculated scores by tallying correct responses based on the answer key. This process was automated to eliminate manual errors and provide immediate results. Scoring outputs were displayed as individual scores per student and aggregate results for all test participants.

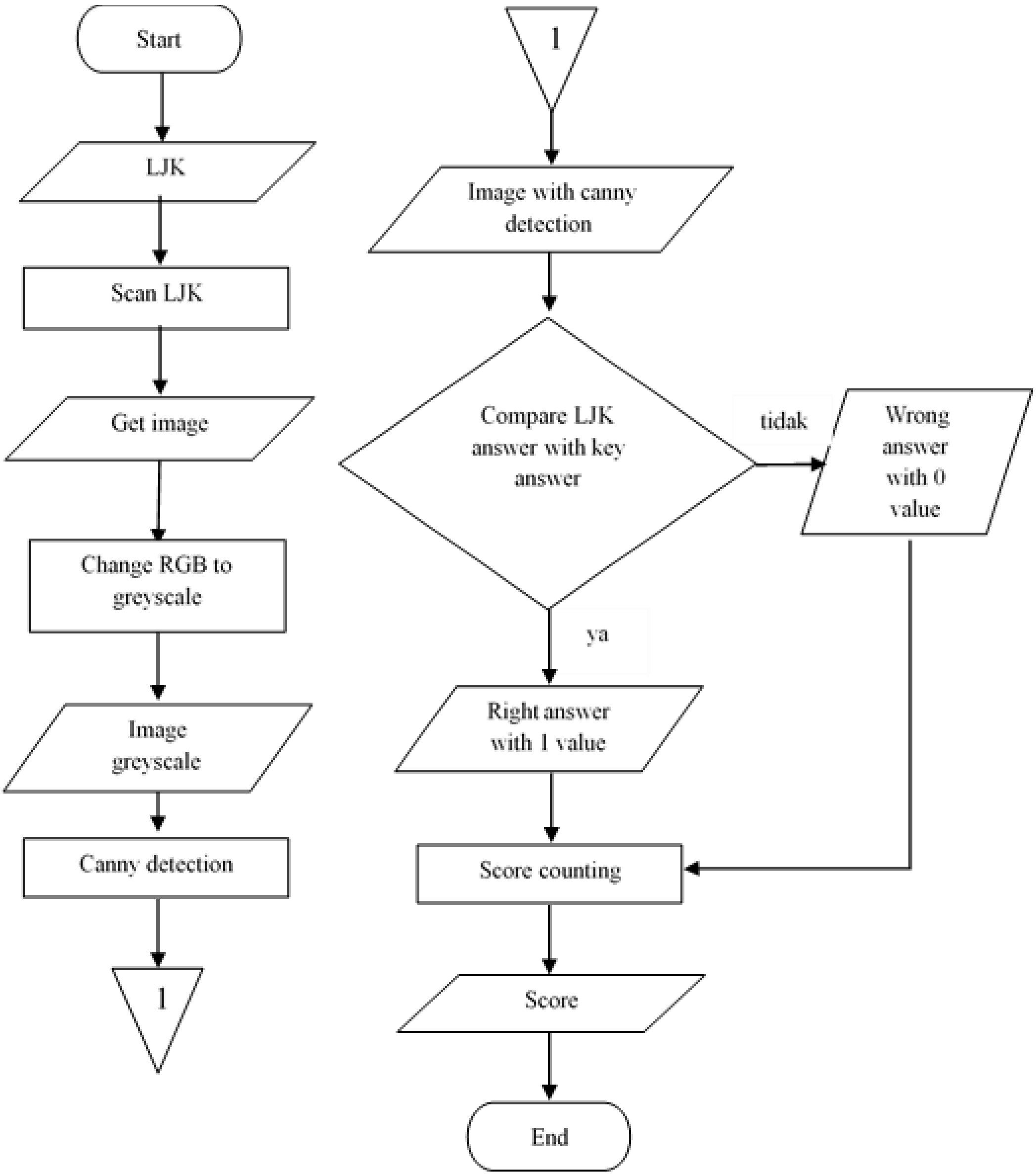

Figure 1. Flowchart Canny Process

RESULTS AND DISCUSSION

This study successfully developed a digital image processing application that uses the Canny Algorithm to detect answers on Computer Answer Sheets (Lembar Jawaban Komputer, LJK). The implementation process, results, and key findings are presented as follows:

1. Process of Edge Detection Using the Canny Algorithm. The edge detection process comprised several sequential steps to ensure high accuracy in detecting the edges of LJK responses.
2. Input RGB Image. The first step involved taking a digital RGB image of an LJK as the input. The sample image used was 8x8 pixels, with color intensities ranging between 0 and 255. The transformation from RGB to grayscale was carried out to simplify the computational process. Grayscale conversion reduced the image depth while maintaining the essential intensity information needed for edge detection.

81	84	92	88	83	83
87	85	88	83	80	85
93	94	87	87	82	81
96	94	84	91	93	85
93	94	89	87	93	93
87	92	97	89	78	87

Figure 2. Sample Matrix

3. Gaussian Filtering. To minimize noise and achieve a smoother image, Gaussian filtering was applied. This step involved convolution between the input image and a Gaussian mask, which helps blur unnecessary details while preserving significant structural edges. The resulting smoothed image prepared the data for accurate gradient calculations. The filtered image retained the intensity values relevant for subsequent processing steps.

1	2	1
2	3	2
1	2	1

Figure 3. Gaussian Filter

Gaussian Mask applied for smoothing and noise reduction. Gradient Calculation. This step is crucial in identifying potential edge pixels based on intensity changes in the image.

-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1

Figure 4. Gradient Calculation

Matrices representing and gradients from Sobel convolution. Hysteresis Thresholding. To refine the edge detection process, hysteresis was applied. Two thresholds—upper and lower ensured that only significant edges were preserved while weak edges were discarded. This step also connected broken edges, producing a more cohesive edge map. Final edge detection result showing clearly identified response areas. Output after applying hysteresis:

255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

Figure 5. Edge Detection Output

4. Scoring and Results. The detected edges were matched with the predefined answer key to score the responses. Correct responses were scored as 1, while incorrect or missing responses were scored as 0. The application calculated the total score for each LJK and automatically generated detailed reports.

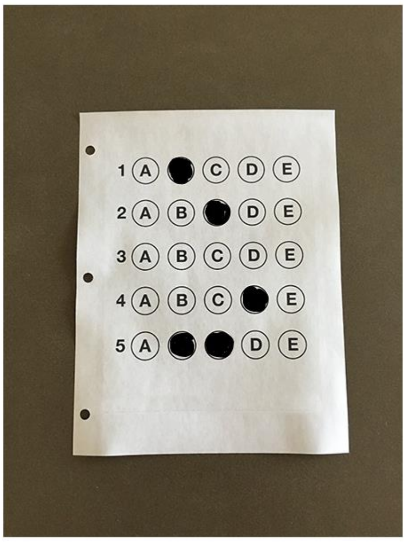

Figure 6. LJK Example

5. Key Observations from Scoring Process: The application processed multiple LJKs with a consistent level of accuracy. Scores and corresponding reports were generated automatically, minimizing manual intervention. Sample scanned LJK output:

Figure 7. Processed LJK Output

The study successfully demonstrated the practical implementation of the Canny Algorithm for detecting answers on LJK sheets. Several key points and observations are discussed below:

1. Accuracy of the Canny Algorithm. The Canny Algorithm proved to be highly reliable in detecting edges with minimal error. Its ability to localize edges accurately and handle noise effectively was evident from the smooth and uninterrupted edges in the final processed images.
2. Comparison with Other Methods. Compared to other edge detection methods such as Sobel, Prewitt, or Robert, the Canny Algorithm provided several

advantages: Noise Reduction: The Gaussian filtering step significantly reduced image noise before edge detection, enhancing accuracy. Edge Localization: The algorithm's multi-step approach, including non-maximum suppression and hysteresis, ensured sharper and more defined edges. Error Minimization: The algorithm avoided detecting false edges, a common limitation in simpler methods like Sobel or Prewitt.

3. Challenges Encountered. While the results were promising, some challenges were observed: Dependence on Input Quality: Poorly scanned LJKs with faint or incomplete markings occasionally led to misdetections. This highlights the importance of using high-quality scanners or improving preprocessing steps. Threshold Sensitivity: The selection of hysteresis thresholds and required careful tuning to balance between detecting valid edges and minimizing noise.
4. Improvements and Future Directions. To enhance the application further, the following improvements are recommended: Advanced Preprocessing: Implementing techniques such as adaptive histogram equalization or morphological operations could improve the detection of faint markings. Robust Thresholding: Using adaptive or machine-learning-based thresholding could optimize the hysteresis step for varied input conditions. Scalability: Expanding the application to process larger batches of LJKs simultaneously could increase its utility in large-scale assessments.
5. Practical Implications. This study offers several practical benefits: Cost-Effectiveness: By using standard scanning devices and Python-based image processing, the application eliminates the need for expensive LJK scanning hardware. Automation: The application automates the grading process, reducing the time and effort required by educators. Scalability: The developed system can be scaled to process LJKs for large institutions, making it suitable for broader educational use.
6. Broader Impact. The application has the potential to transform how educational institutions conduct assessments by making grading faster, more accurate, and less resource-intensive. This is especially beneficial for schools with limited budgets, as they can now achieve the same level of efficiency as institutions with specialized equipment.

CONCLUSION

In conclusion, the application designed for processing digital images using the Canny method effectively detects and inspects Optical Mark Recognition (OMR) answer sheets. The Canny method excels in performing edge detection on OMR patterns, ensuring accurate identification of the selected answers. The detection process highlights only the correct answer choices, while any discrepancies, such as multiple selections or incomplete marks, are indicated by crossed-out choices or over-inking. Furthermore, the application incorporates a database menu to store both student responses and answer keys, providing efficient data management. Experimental tests demonstrate that this system is reliable and proficient in detecting answers and correcting OMR sheets, making it a viable tool for automated answer sheet analysis.

REFERENCES

[1] F. M. Rozaqi and W. Wahyono, "Sistem pengukuran ketinggian air sungai berbasis deteksi tepi Sobel, "Jurnal Teknologi dan Sistem Komputer, vol. 10, no. 1, pp. 38-45, 2022.

[2] W. Supriyatin, "Perbandingan Metode Sobel, Prewitt, Robert dan Canny pada Deteksi Tepi Objek Bergerak," ILKOM Jurnal Ilmiah, vol. 12, no. 2, pp. 112–120, 2020.

[3] Saluky and Y. Marine, "Penerapan Algoritma Deteksi Tepi Canny Menggunakan Python dan OpenCV," Smart Techno (Smart Technology, Informatics and Technopreneurship), vol. 5, pp. 1–7, 2023.

[4] G. Q. O. Pratamasunu, R. E. Pawening, and U. N. Wulandari, "Deteksi Pilihan Jawaban Otomatis pada Lembar Jawaban Komputer Menggunakan Metode Image Thresholding dan Contour Sorting," COREAI: Jurnal Kecerdasan Buatan, Komputasi dan Teknologi Informasi, vol. 1, no. 1, pp. 1–9, Nov. 2020.

[5] L. P. R. Noviana, I. P. E. Indrawan, and G. I. Setiawan, "Analysis of Canny Edge Detection Method for Facial Recognition in Digital Image Processing," Jurnal Manajemen dan Teknologi Informasi (JMTI), vol. 15, no. 2, pp. 15–22, 2024.

[6] A. Maulana, F. Auliatunnajah, N. Rosidin, M. R. Rizki Darmawan, and P. Rosyani, "Implementasi OpenCV dengan Metode Image Thresholding pada Gambar," Jurnal AI dan SPK: Jurnal Artificial Inteligent dan Sistem Penunjang Keputusan, vol. 2, no. 1, pp. 27–32, Jun. 2024.

[7] R. Novria, B. Kurniawan, and Suryanto, "Aplikasi Pemesanan Makanan di Bebek dan Ayam Tekaeng Menggunakan PHP dan MySQL," Jurnal Informatika dan Komputer (JIK), vol. 13, no. 1, pp. 15–26, Jun. 2022.

[8] S. Supiyandi, T. Panggabean, N. Ramadhan, S. R. Dewi, and S. Yusra, "Deteksi Tepi Sederhana pada Citra Menggunakan Operator Sobel," Repeater, vol. 2, no. 3, pp. 43–56, Jun. 2024.

[9] I. Zahari, Z. Pratama, W. Mahmud, and D. A. Wibowo, "Sistem Pengecekan Lembar Jawaban Komputer Dengan Optical Mark Recognition (OMR) Berbasis Open Computer Vision Python," in Seminar Nasional Inovasi dan Pengembangan Teknologi Terapan (SENOVTEK), Cilacap, 20 Oktober 2022.

[10] Sugiyono, Metode Penelitian Kuantitatif, Kualitatif dan R&D. Bandung: Alfabeta, 2020, ISBN: 978-602-289-533-6

[11] A. Arnita, F. Marpaung, F. Aulia, N. Suryani, R. C. Nabila, and A. B. Surya, Computer Vision dan Pengolahan Citra Digital, Surabaya, Jawa Timur, Indonesia: Pustaka Aksara, 2022, ISBN: 978-623-8230-27-3.

[12] S. Saluky and Y. Marine, "Penerapan Algoritma Canny untuk Deteksi Tepi Menggunakan Python dan OpenCV," SMART TECHNO (Smart Technology, Informatic, and Technopreneurship), vol. 5, no. 1, pp. 1–7, Feb. 2023.

[13] W. Supriyatin, "Perbandingan Metode Sobel, Prewitt, Robert dan Canny pada Deteksi Tepi Objek Bergerak," ILKOM Jurnal Ilmiah, vol. 12, pp. 112–120, 2020.

[14] L. Maximillian, Y. F. Riti, M. Anugraha, and Y. J. Palis, "Perbandingan Algoritma Sobel dan Canny untuk Deteksi Tepi Citra Daun Lidah Buaya," KOMPUTA: Jurnal Ilmiah Komputer dan Informatika, vol. 12, no. 2, pp. 69, Oct. 2023.

[15] I. Ummah and N. Yannuansa, "Analisis Pendeteksian Tepi Objek pada Pengolahan Citra," in Seminar Nasional SAINSTEKNOPAK Ke-4, LPPM UNHASY TEBUIRENG, Jombang, 2020.