International Journal Software Engineering and Computer Science (IJSECS)
5 (3), 2025, 938-948
Published Online December 2025 in IJSECS (http://www.journal.lembagakita.org/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v5i3.5183.

RESEARCH ARTICLE Open Access

Optimization of Tesseract OCR for Automatic Text Extraction on Indonesian ID Cards (KTP) Through Image Quality Enhancement Using Preprocessing Techniques

Gilang Ramadhan *
Informatics Engineering Study Program, Faculty of Computer Science, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, East Jakarta City, Special Capital Region of Jakarta, Indonesia.
Corresponding Email: initial.gr.101@gmail.com.

Dadang Iskandar Mulyana
Informatics Engineering Study Program, Faculty of Computer Science, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, East Jakarta City, Special Capital Region of Jakarta, Indonesia.
Email: mahvin2012@gmail.com.

Sopan Adrianto
Informatics Engineering Study Program, Faculty of Computer Science, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, East Jakarta City, Special Capital Region of Jakarta, Indonesia.

Received: July 26, 2025; Accepted: September 5, 2025; Published: December 1, 2025.

Abstract: Tesseract OCR ranks among the most widely adopted open-source tools for text extraction. Nevertheless, processing documents with degraded image quality—including blurry e-KTPs, low-contrast specimens, or those affected by uneven lighting—presents substantial challenges. We conducted experimental research to generate empirical data supporting the development of text detection systems for e-KTPs operating under non-ideal conditions. Our methodology involved testing 10 e-KTP images, each containing 15 text attributes, yielding 150 evaluated data points. Image preprocessing proceeded sequentially through grayscale conversion, denoising, contrast enhancement (CLAHE), and thresholding to improve image clarity prior to Tesseract OCR processing. We evaluated accuracy using confusion matrix analysis, emphasizing True Positive (TP), False Positive (FP), and False Negative (FN) metrics. Results demonstrate that preprocessing stages substantially improved text readability. Baseline OCR accuracy of 39.55% increased incrementally: +22.68% following grayscale conversion, +47.70% after denoising, +60.99% post-CLAHE application, and +19.62% after thresholding, culminating in 64.97% accuracy upon completing all preprocessing stages. Average TP values rose from 4 to 8 out of 15 attributes per image, while precision remained stable at 100% (FP = 0). Despite variable CLAHE performance across samples, preprocessing stages proved essential for OCR systems operating under degraded image conditions. Our work introduces a novel preprocessing pipeline tailored specifically to Indonesian e-KTP characteristics, providing quantitative benchmarks and systematic analysis that can inform the development of more adaptive digitalization and verification systems for population documents under real-world field conditions.

Keywords: Optical Character Recognition; Tesseract; Pre-Processing; Image Enhancement; Confusion Matrix.

1. Introduction

OCR accuracy on e-KTP has not been optimal, with some attributes having detection rates as low as 75%. The problems mostly come from blurry images, misplacement or mis-sizing of the elements, and unstructured layouts. Although initial tests resulted in an accuracy of 98.09%, the results varied substantially across attributes and therefore indicated the need for improved image preprocessing and structured visual design to enhance system performance [1]. Other approaches have had mixed outcomes. Faster R-CNN combined with Template Matching (Oriented FAST and Rotated BRIEF with KNN-BFM) for matching e-KTP images only reached an accuracy of 43.46%, which is much lower than the accuracies achieved by OCR-based methods even though it attained a 94% mean Average Precision (mAP) in object detection. Feature-based image matching methods are still not sufficient for reliable verification and validation of e-KTP even if they are faster. Dataset quality improvement and feature extraction from important parts of e-KTP may increase accuracy [2].

The manual entry of KTP data by typing is still an inefficient and error-prone process. Automated data entry systems using OCR are available but choosing the right OCR methods is difficult because lighting conditions and camera types significantly influence extraction results. Tests showed that Pytesseract has better accuracy (98.33%) when compared to template matching (67.33%), thus making it suitable for developing an OCR-based automatic KTP data entry system [3]. An OCR and Template Matching Correlation desktop-based system detected text in 15 fields of e-KTP but got only 79% average accuracy, which means there is still some work needed before such systems can be used reliably in population data archiving digitalization [3]. Earlier studies have established that Tesseract OCR has limitations in text extraction from images under different quality conditions such as blur e-KTP images, high noise levels, uneven lighting, and low resolution [1][2]. Thresholding, denoising, and contrast enhancement techniques have been effective in preprocessing image input to OCR. Optimization effectiveness measurement is generally done using a confusion matrix with special emphasis on key attributes NIK, Name, and Date of Birth since these are frequently misread [3].

Research on the Reichsanzeiger-GT dataset shows that high-quality, well-annotated ground truth data and adherence to standardized transcription guidelines can significantly improve OCR performance over complex layouts and difficult character sets. This supports the need for structured dataset preparation in improving OCR accuracy for identity documents. A desktop-based system using OCR and Template Matching Correlation algorithms was able to detect text across 15 e-KTP fields but averaged only 79% accuracy. This indicates an urgent need to improve the system's accuracy and reliability in supporting the digitalization of population data archiving processes. The same challenges have been found with non-Latin scripts; preprocessing steps such as rotation correction, stroke enhancement, and noise removal have been very helpful in improving OCR performance. One study that used Tesseract OCR for Javanese characters applied a hybrid bounding box annotation method which attained up to 97.50% accuracy. This proves that careful dataset preparation and preprocessing are critical for maximizing recognition accuracy over various script types. Existing studies do not yet provide systematic analysis of sequential preprocessing effects on e-KTP text extraction under real-world degraded conditions. Most of the previous research has been focused on either detection models or direct application of OCR without a comprehensive evaluation of preprocessing pipelines that are specific to the characteristics of Indonesian e-KTP. This work fills that gap by implementing and evaluating a sequential preprocessing pipeline—grayscale conversion, denoising, CLAHE, and thresholding—specifically designed for e-KTP images. Quantitative benchmarks are provided using confusion matrix analysis over 150 attributes from 10 e-KTP samples in order to supply empirical data that can be used toward developing more robust and adaptive digitalization systems for population documents under field conditions.

2. Related Work

2.1 OCR-Based Approaches for e-KTP Text Extraction

Several studies have explored Optical Character Recognition (OCR) technology for Indonesian e-KTP data extraction, addressing manual data entry inefficiencies in population services. Toha and Triayudi (2022) developed a web-based OCR system achieving 100% attribute detection accuracy, demonstrating OCR's potential for enhancing digital public services [1]. However, this high accuracy was achieved under controlled conditions and may not reflect real-world performance with degraded image quality. Octaviani *et al.* (2023) compared Pytesseract and Template Matching methods under various lighting conditions, confirming Pytesseract's superior accuracy (98.33%) over Template Matching (67.33%) [7]. Putra *et al.* (2023) utilized Google Cloud Vision OCR integrated with web services, complemented by regex and fuzzy matching for post-processing [9], while Rusli *et al.* (2020) proposed an OCR and NLP-based pipeline addressing common OCR errors through regex correction and confidence scoring [11]. Sugiarta *et al.* (2021) introduced a CNN-based OCR approach specifically for e-KTP text extraction, achieving faster processing times and reasonable accuracy on small datasets [5]. Despite these advances, most studies focus on overall system accuracy without detailed per-attribute analysis, limiting understanding of which specific fields require targeted improvement.

2.2 Hybrid Detection and Matching Methods

Alternative approaches have combined object detection with feature-based matching to address OCR limitations in handling low-quality images. Hudaya *et al.* (2021) proposed Faster R-CNN for e-KTP area detection combined with ORB and KNN-BFM for image matching [2][6]. While their detection accuracy reached 94% mAP, matching accuracy remained relatively low at 43.46%, highlighting the inadequacy of visual-based verification compared to text-based OCR for reliable e-KTP authentication. Haris *et al.* (2023) focused on Template Matching for e-KTP archiving in rural areas, achieving only 79% accuracy and recognizing the critical need for better image preprocessing [8]. These findings suggest that feature-based methods, despite faster processing times, cannot yet replace OCR-based text extraction for verification tasks requiring high precision across multiple attributes.

2.3 OCR Applications Beyond Identity Documents

OCR technology has been successfully applied across diverse domains, demonstrating its adaptability to various text recognition challenges. In transportation, Ibnutama and Suryanata (2020) applied Tesseract OCR for vehicle license plate extraction, achieving 96.39% average accuracy for parking security systems [3]. Angela *et al.* (2024) developed a Flutter-based mobile OCR application for general text extraction, demonstrating 88% user acceptance and strong accuracy under optimal conditions [12]. In healthcare, Putri *et al.* (2024) combined OCR with CNN for classifying medicine placement in pharmacies, achieving 83.78% accuracy [15]. Other implementations include student card data extraction (Reswan *et al.*, 2024) [16], guest registration systems with face recognition integration (Darajat *et al.*, 2023) [17], OCR-based e-KTP detection using EasyOCR and TensorFlow SSD models (Iskandar *et al.*, 2022) [18], Indonesian currency recognition for visually impaired users (Bahar *et al.*, 2023) [14], and nutritional label extraction from food packaging (Suhairi *et al.*, 2025) [15]. While these applications demonstrate OCR's versatility, they also reveal that domain-specific preprocessing and adaptation remain essential for achieving consistent accuracy across varying image conditions.

2.4 Research Gaps and Contributions

Despite demonstrated potential, existing e-KTP OCR systems suffer from inconsistent accuracy across attributes, particularly under suboptimal conditions such as blur, uneven lighting, and low resolution [1][2][7][8]. Most approaches focus either on detection models (*e.g.*, Faster R-CNN) [2][6]or direct OCR methods (Pytesseract, Google Cloud Vision) [7][9], but neglect comprehensive preprocessing pipelines that can significantly improve recognition performance. Feature-based image matching methods, despite achieving high object detection mAP scores, show lower accuracy in actual text extraction [2], indicating limited suitability for reliable e-KTP verification. Furthermore, previous studies primarily evaluate overall detection accuracy without systematic per-attribute analysis, obscuring which specific fields—such as NIK, Name, or Date of Birth—require targeted improvement. This research addresses these limitations by integrating advanced image preprocessing techniques—grayscale conversion, denoising, CLAHE, and adaptive thresholding—with structured dataset preparation to improve OCR accuracy across multiple critical attributes. Building upon insights from studies on complex datasets like Reichsanzeiger-GT [27] and non-Latin scripts such as Javanese characters [29], our approach emphasizes high-quality annotated datasets and robust preprocessing to handle diverse visual challenges. Unlike previous works, this study employs per-attribute evaluation using confusion matrix analysis across 150 attributes from 10 e-KTP samples, enabling detailed understanding of OCR performance for individual fields. By systematically evaluating sequential preprocessing effects under real-world degraded conditions, we provide empirical benchmarks that can inform the development of more robust and generalizable OCR systems for e-KTP automation, bridging the gap between controlled laboratory results and practical deployment requirements.

3. Research Method

3.1 Research Design and Overview

This research employs an experimental method to systematically evaluate the effect of preprocessing techniques on text extraction accuracy from e-KTP images using the Tesseract OCR engine. Distinct from previous studies that primarily focus on overall detection accuracy or apply preprocessing arbitrarily, this work emphasizes systematic optimization of OCR preprocessing under varying image quality conditions, aiming to contribute to more robust data extraction systems applicable in real-world administrative scenarios. The experimental design tests each preprocessing stage both sequentially and independently to isolate and quantify the contribution of individual techniques to overall OCR performance improvement. The research targets attributes particularly prone to recognition errors due to poor image quality, including NIK (National Identity Number), Name, and Date of Birth. The dataset comprises e-KTP images collected from the Roboflow platform, representing diverse quality levels commonly encountered in field conditions—including blur, uneven lighting, low resolution, and noise artifacts. Each image contains 15 text attributes serving as evaluation points, with ground truth prepared manually to ensure evaluation validity. The preprocessing pipeline is implemented using Python with OpenCV and pytesseract libraries, while the Tesseract OCR engine is configured for Indonesian language recognition (lang='ind'). Evaluation employs multiple metrics including accuracy, precision, recall, F1-score, character-level recognition rate per attribute, and confusion matrix analysis to comprehensively assess classification errors at each preprocessing stage.

3.2 Research Workflow

Figure 1 illustrates the complete experimental workflow, which begins with code initialization in Google Colab and e-KTP image upload. The process follows two parallel paths for comparative analysis: the Original Process path, where raw, unaltered images are sent directly to the OCR Text Extraction engine to establish baseline performance, and the Preprocessing Process path, where images undergo individual preprocessing techniques—Grayscale Conversion, Denoising, Contrast Enhancement (CLAHE), and Thresholding—each tested independently to isolate their effects. After OCR text extraction from both paths, results enter a comparison stage where ground truth data is formulated for each process. All OCR outputs and corresponding ground truth data are compiled into an Excel file for systematic analysis. Performance metrics are calculated by comparing OCR results against ground truth, enabling quantification of accuracy improvements or degradations attributable to each preprocessing method. The final stage produces comprehensive reports summarizing whether and to what extent each preprocessing technique optimizes OCR performance, providing empirical evidence for preprocessing strategy selection in e-KTP digitalization systems.

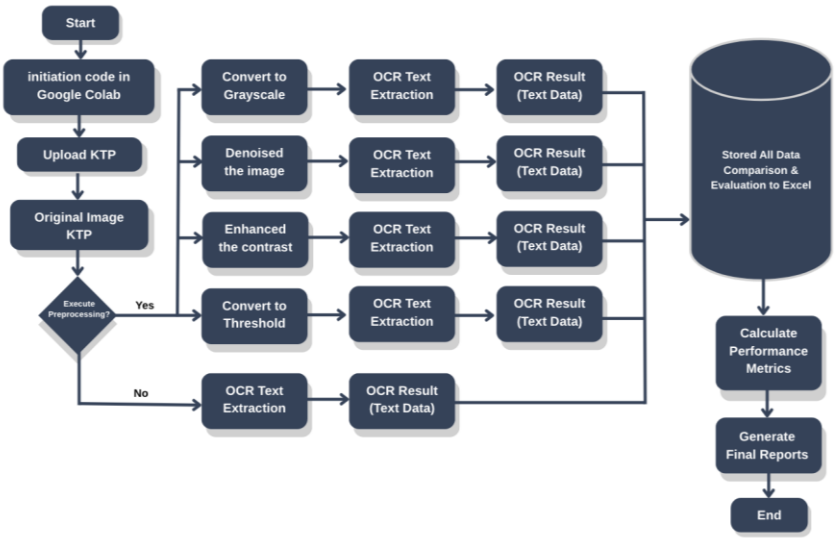

Figure 1. Research Workflow Flowchart

3.3 Preprocessing Pipeline

The preprocessing pipeline consists of four sequential stages, each designed with specific technical parameters fine-tuned through preliminary experiments to achieve optimal performance on the e-KTP dataset. While the stages are applied sequentially in the complete pipeline, each is also tested independently to assess its individual contribution to OCR accuracy improvement.

3.3.1 Grayscale Conversion

1) Purpose: Converts color (RGB) images into grayscale format, simplifying visual information from three color channels into a single intensity channel, thereby reducing processing complexity while retaining essential luminance information for text recognition.
2) Implementation:
   - Copy
   - gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
3) Technical Parameters: Color space conversion from BGR to GRAY using OpenCV's standard conversion formula.

4) Effect: Eliminates irrelevant color information, facilitates subsequent text segmentation, and reduces computational load for downstream processing stages without sacrificing text distinguishability.

3.3.2 Noise Removal (Denoising)

1) Purpose: Removes random noise artifacts such as speckles, sensor noise, and compression artifacts that may interfere with character recognition, while preserving critical text edge information.
2) Implementation:
   - Copy
   - denoised_img = cv2.fastNlMeansDenoising(gray_img, None, h=14,
   - templateWindowSize=7,
   - searchWindowSize=21)
3) Technical Parameters:
   - h = 14: Filter strength controlling the degree of noise reduction; higher values remove more noise but risk over-smoothing text details.
   - templateWindowSize = 7: Size of the pixel patch used for similarity comparison; determines local feature preservation.
4) searchWindowSize = 21: Size of the area searched for similar patches; larger values improve denoising quality at the cost of computational time.
5) Effect: Reduces noise while retaining critical details such as letter edges and stroke boundaries, making text more distinguishable from background artifacts and improving OCR segmentation accuracy.

3.3.3 Contrast Enhancement (CLAHE)

1) Purpose: Improves local contrast to enhance visibility of faint text or text appearing under uneven lighting conditions, addressing common e-KTP image quality issues encountered in field capture scenarios.
2) Implementation:
   - Copy
   - clahe = cv2.createCLAHE(clipLimit=1.8, tileGridSize=(21,21))
   - enhanced_img = clahe.apply(denoised_img)
3) Technical Parameters:
   - clipLimit = 1.8: Limits contrast amplification to prevent over-enhancement and noise amplification in uniform regions; balances contrast improvement with artifact suppression.
   - tileGridSize = (21,21): Block size for adaptive histogram calculation; determines the locality of contrast enhancement, with larger tiles providing smoother transitions.
4) Effect: Enhances intensity differences between text and background regions, enabling OCR to distinguish character boundaries more accurately, particularly for attributes printed with low contrast or affected by shadows and uneven illumination.

3.3.4 Thresholding (Binarization)

1) Purpose: Converts grayscale images into binary (black-and-white) format, creating clear separation between text (foreground) and background, which is optimal for Tesseract OCR's character segmentation algorithms.
2) Implementation:
   - Copy
   - _, binary_img = cv2.threshold(enhanced_img, 100, 255, cv2.THRESH_BINARY)
3) Technical Parameters:
   - threshold_value = 100: Pixels with intensity values >100 are classified as background (white), while values ≤100 are classified as text (black); this value was empirically determined through preliminary testing.
   - max_value = 255: Maximum intensity assigned to background pixels, ensuring full contrast in the binary image.
4) Effect: Produces images with sharp black text on white background, facilitating character segmentation by Tesseract OCR and eliminating grayscale ambiguity that can cause misclassification of edge pixels.

3.4 Evaluation Metrics

System performance was evaluated using confusion matrix analysis and derived classification metrics. Evaluation was conducted at the attribute level, comparing OCR extraction results against manually prepared ground truth for each of the 15 e-KTP attributes across all test images.

3.4.1 Confusion Matrix Components

The confusion matrix for this study is defined as follows:

1) True Positive (TP): Text attributes correctly detected by OCR with exact match to ground truth, indicating successful recognition.
2) False Positive (FP): Text attributes predicted as correct by OCR but actually incorrect (not an exact match to ground truth), representing over-detection or misrecognition.
3) False Negative (FN): Text attributes that should have been correctly detected but failed OCR recognition (result did not match ground truth), representing under-detection or miss-detection.
4) True Negative (TN): Cases where non-attributes or empty areas were correctly identified as absent. In this per-attribute evaluation context, TN values are generally zero because each attribute is assumed to have valid ground truth.

3.4.2 Performance Metrics

From the confusion matrix, the following metrics were calculated:

Accuracy: Overall correctness of OCR predictions across all attributes.

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

Precision: Proportion of correctly recognized attributes among all attributes predicted as correct.

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

Recall (Sensitivity): Proportion of correctly recognized attributes among all attributes that should have been recognized.

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

F1-Score: Harmonic mean of precision and recall, providing a balanced measure of OCR performance.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Additionally, character-level recognition rate per attribute was calculated to assess partial matching performance, providing insight into near-miss cases where OCR output was close but not identical to ground truth. Confusion matrix analysis was conducted to identify systematic error patterns, including specific attributes prone to miss-detection (FN) and the preprocessing stages most effective at reducing such errors.

4. Result and Discussion

4.1 Results

At this stage, the data obtained from e-KTP samples are processed by applying image preprocessing techniques to improve the quality of images that initially have low accuracy when detected by OCR. This processing aims to improve the accuracy of text extraction from e-KTP images, especially for attributes that are often problematic, such as NIK, Name, Place and Date of Birth, and Address. The data used in this study come from various samples of e-KTP images collected under different conditions, such as varying lighting, image quality, and orientation. After undergoing the preprocessing stage, the OCR process is carried out to evaluate the accuracy improvement before and after the application of preprocessing techniques. The results of this analysis are presented in the following tables to provide a clearer picture of the increase in recognition accuracy, especially on images categorized as poor quality. In this stage, the data are processed and classified according to the detection outcomes, and the accuracy metrics are calculated using True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) indicators. Based on the data obtained from the experiments, improvements in detection accuracy before preprocessing can be seen, as presented in Table 1 below:

Table 1. Detection Before Preprocessing

No	KTP	TP	FP	FN	TN	Accuracy	Recall	Precision	F1-Score	Detection
1	KTP 1	9	6	0	0	60%	60.00%	100.00%	75.00%	76.55%
2	KTP 2	6	9	0	0	40%	40.00%	100.00%	57.14%	49.36%
3	KTP 3	6	9	0	0	40%	40.00%	100.00%	57.14%	49.74%
4	KTP 4	1	14	0	0	7%	6.67%	100.00%	12.50%	36.11%
5	KTP 5	1	14	0	0	7%	6.67%	100.00%	12.50%	11.83%
6	KTP 6	0	15	0	0	0%	0.00%	0.00%	0.00%	8.25%
7	KTP 7	0	15	0	0	0%	0.00%	0.00%	0.00%	0.00%
8	KTP 8	4	11	0	0	27%	26.67%	100.00%	42.11%	50.38%
9	KTP 9	3	12	0	0	20%	20.00%	100.00%	33.33%	72.05%
10	KTP 10	2	13	0	0	13%	13.33%	100.00%	23.53%	41.27%

Testing was conducted by comparing the results of OCR text extraction on e-KTP images with ground truth data. The following is a summary of the test results after image preprocessing:

Table 2. OCR Results After Preprocessing

No	KTP	Preprocessing				
		Original	Grayscale	Denoise	CLAHE	Threshold
1	KTP 1	76.55%	96.53%	96.53%	96.00%	94.29%
2	KTP 2	49.36%	82.83%	70.72%	13.36%	87.80%
3	KTP 3	49.74%	95.00%	96.86%	82.59%	95.80%
4	KTP 4	36.11%	83.13%	62.10%	24.70%	59.27%
5	KTP 5	11.83%	78.18%	96.22%	90.77%	95.48%
6	KTP 6	8.25%	29.81%	62.90%	64.90%	91.53%
7	KTP 7	0.00%	80.94%	77.57%	75.68%	89.92%
8	KTP 8	50.38%	72.18%	98.27%	30.03%	93.46%
9	KTP 9	72.05%	86.33%	86.16%	73.42%	92.02%
10	KTP 10	41.27%	64.81%	66.51%	80.45%	77.07%

Table 3. Confusion Matrix Results After Preprocessing

No	KTP	Preprocessing				
		Original	Grayscale	Denoise	CLAHE	Threshold
		TP / FN	TP / FN	TP / FN	TP / FN	TP / FN
1	KTP 1	9 / 6	13 / 2	13 / 2	12 / 3	8 / 7
2	KTP 2	6 / 9	12 / 3	8 / 7	1 / 14	8 / 7
3	KTP 3	6 / 9	14 / 1	14 / 1	9 / 6	13 / 2
4	KTP 4	1 / 14	10 / 5	6 / 9	2 / 13	5 / 10
5	KTP 5	1 / 14	11 / 4	11 / 4	11 / 4	10 / 5
6	KTP 6	0 / 15	3 / 12	7 / 8	7 / 8	9 / 6
7	KTP 7	0 / 15	8 / 7	9 / 6	8 / 7	8 / 7
8	KTP 8	4 / 11	8 / 7	12 / 3	3 / 12	8 / 7
9	KTP 9	3 / 12	10 / 5	9 / 6	7 / 8	10 / 5
10	KTP 10	2 / 13	7 / 8	7 / 8	7 / 8	6 / 9

A comprehensive analysis of both tables convincingly shows that preprocessing is an essential step to optimize OCR performance on e-KTP images. Table 2 provides an overview of accuracy percentages, while Table 3 offers concrete evidence in the form of the number of correctly detected attributes (True Positives, TP) and incorrectly detected ones (False Negatives, FN). The baseline OCR performance without preprocessing shows consistently low results. KTP 7 had an accuracy of 0.00%, and KTP 6 only reached 8.25%. Both images recorded 0 True Positives (TPs), confirming the OCR engine's inability to detect attributes on raw images. Grayscale conversion provided a dramatic increase in accuracy. KTP 7 jumped from 0.00% to 80.94%, and KTP 4 increased from 36.11% to 83.13%. The TP count for KTP 7 rose from 0 to 8, while KTP 2 jumped from 6 to 12 TPs. Removing color information allows the OCR engine to focus on luminance patterns that represent text characters more clearly. Denoising was very effective, often delivering the highest accuracy. KTP 5 jumped from 11.83% to 96.22%, and KTP 8 reached 98.27%, indicating these images were affected by significant noise. The TP count for KTP 8 increased from 4 to 12, and KTP 5 from 1 to 11. Noise reduction proves necessary for images captured under poor conditions or with lower-quality cameras. CLAHE showed inconsistent results. While it successfully reached 96.00% on KTP 1, it dropped to only 13.36% on KTP 2 and 30.03% on KTP 8. CLAHE only detected 1 TP for KTP 2, far below the 12 TPs achieved by grayscale. The varied performance suggests CLAHE's effectiveness depends heavily on the initial contrast characteristics of each image. Thresholding is the most reliable method, consistently producing high accuracy percentages. It achieved 95.80% on KTP 3 and 91.53% on KTP 6. KTP 6, which had no TPs in the original image, successfully detected 9 TPs after thresholding. Converting images into binary format creates a stark black-and-white contrast that clearly separates text from background, making character recognition easier for the OCR engine.

4.2 Discussion

The experimental results clearly show that image preprocessing significantly improves the accuracy of text extraction from e-KTP images using the Tesseract OCR engine. This finding supports the widely held view in OCR research that image quality is crucial for performance, especially with real-world documents that have inherent quality variations.

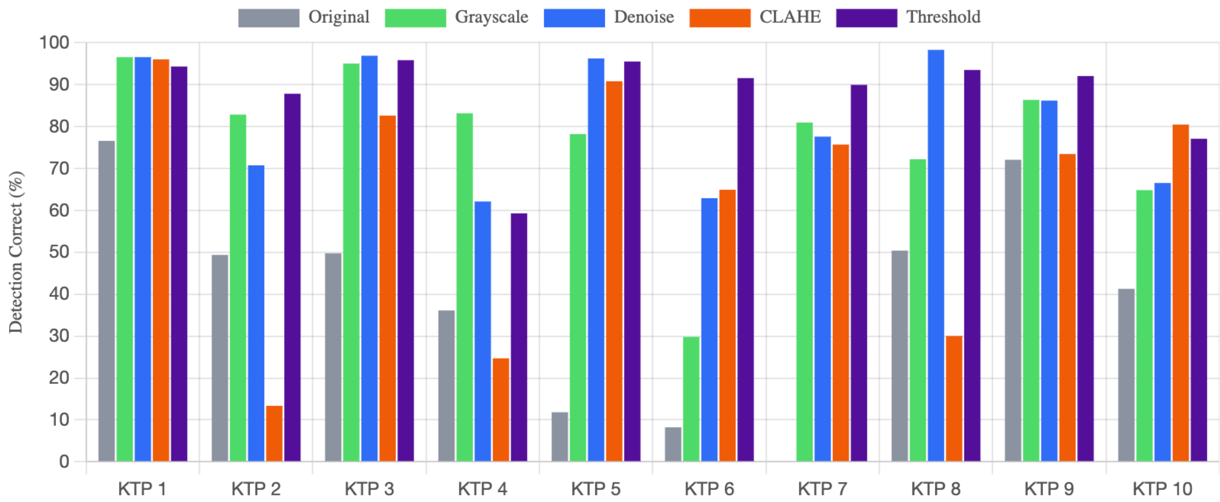

Figure 2. Final Report

Applying sequential preprocessing stages, from grayscale to thresholding, substantially enhanced the quality of e-KTP images. On average, the percentage of correctly extracted characters more than doubled, increasing from 30.82% for original images to 64.97% after full preprocessing. The average True Positive (TP) count rose from 4.1 TPs per KTP to 8.7 TPs out of 15 total attributes. These results show that preprocessing effectively doubles character detection capabilities, confirming its necessary role in optimizing OCR performance. The initial preprocessing stages of grayscale and denoise proved to be fundamental. Both consistently increased the number of TPs and significantly reduced False Negatives (FNs), demonstrating their effectiveness in cleaning images by removing noise and making characters clearer for the OCR system. By eliminating color information and visual disturbances, the OCR engine can focus on textual features that matter most for accurate recognition. The Contrast Limited Adaptive Histogram Equalization (CLAHE) stage showed inconsistent results. While it improved some images, it caused a drastic decrease in TPs for others, specifically KTP 2, 4, and 8. Overly aggressive contrast enhancement can distort or remove character details, hindering detection. Enhancement techniques don't work universally well, and their parameters may require careful adjustment or dynamic selection based on initial image properties to avoid negative effects. Despite the drawbacks of CLAHE, applying thresholding as the final preprocessing step was vital. Thresholding played a significant role in recovering performance that declined during the CLAHE stage and consistently boosted the number of correctly detected characters to an optimal level. Binarization effectively separates text from the background, a technique widely adopted in many OCR applications. The preprocessing pipeline, especially the combination of grayscale, denoise, and thresholding, significantly enhances the Tesseract OCR engine's ability to extract text from challenging e-KTP images. While CLAHE requires careful consideration and potential optimization, the overall approach provides a robust method for improving OCR accuracy in real-world scenarios, supporting more reliable digitalization processes.

5. Conclusion and Future Work

This study successfully demonstrated the significant impact of a sequential image preprocessing pipeline—consisting of grayscale conversion, denoising, CLAHE, and thresholding—on enhancing the performance of Tesseract OCR for text extraction from Indonesian e-KTP images. The experimental results unequivocally show a substantial improvement in text recognition accuracy, with the average correct characters drastically increasing from 30.82% in the original images to 64.97% after applying all preprocessing stages. This more than twofold increase was also reflected in the average True Positive (TP) count, which rose from 4 TPs per e-KTP to 8 TPs per e-KTP, out of 15 total attributes. Furthermore, the system maintained a consistent 100% precision (FP=0) across all tests, indicating its high reliability in avoiding false detections. The analysis also revealed the varied impact of the CLAHE stage, which, in some instances (*e.g.*, KTP 2 and KTP 4), temporarily decreased accuracy before being rectified by subsequent thresholding. This underscores that not all enhancement techniques are universally beneficial and may require careful parameter tuning based on specific image characteristics. Despite this challenge, the overall preprocessing approach provides a robust method for improving OCR accuracy in challenging real-world scenarios, contributing to more reliable digitalization processes for population documents. The findings confirm that systematic preprocessing is essential for optimizing OCR performance on documents with inherent quality variations, particularly for critical attributes such as NIK, Name, and Date of Birth that are prone to recognition errors.

To further enhance the robustness and practical application of this OCR system, several recommendations are proposed for implementation, addressing current limitations, and guiding future research directions. For practical implementation in institutions or companies, it is recommended to integrate this optimized preprocessing pipeline into the document digitalization workflow. The initial step should include using grayscale and denoising, as both consistently provide significant improvements across various image quality conditions. This OCR system can be integrated with existing resident data verification systems to accelerate the e-KTP data entry process, where the text output from the OCR can be directly used as input for the system, reducing the need for manual entry and minimizing human errors. Given that initial image quality is crucial for optimal OCR performance, brief training for staff responsible for scanning or photographing e-KTPs is necessary. Guidelines should be provided to ensure optimal image capture, such as maintaining sufficient lighting, correct positioning, and appropriate distance, to maximize accuracy even before preprocessing is applied. This research has several limitations that should be acknowledged. First, the study only uses 10 sample e-KTP images collected from a limited source (the Roboflow website), which means that the results may not fully reflect the much wider variation of e-KTP image conditions encountered in the field, such as variations in cameras, resolution, shooting angles, and types of physical damage to the cards. Second, the scope of testing was focused on the individual and sequential evaluation of preprocessing stages but did not extensively test different combinations or orders of these techniques. While the stage-by-stage evaluation provides valuable insights into the contribution of each preprocessing method, there may be more optimal configurations or alternative sequences that have not yet been explored and could potentially yield better results. Future research should incorporate a significantly larger and more diverse e-KTP dataset to ensure broader system validation and generalizability. This dataset should encompass a wider range of lighting conditions, resolutions, camera angles, and various levels of image degradation to better represent real-world scenarios. Additionally, future research should focus on optimizing CLAHE parameters (*e.g.*, clip limit and grid size) through more adaptive approaches or machine learning methods. This could involve dynamically adjusting parameters based on initial image properties to avoid detrimental effects and improve performance consistency across various e-KTP qualities. Investigating the combination of this preprocessing pipeline with deep learning-based text detection and recognition models could yield even better results. Furthermore, exploring the fine-tuning of the Tesseract model itself with custom e-KTP data could potentially mitigate persistent miss-detection issues, especially for unique character styles or challenging attributes like NIK, Name, and Date of Birth. Such advancements would contribute to developing a more robust and adaptive OCR system capable of handling the full spectrum of e-KTP image quality variations encountered in practical applications.

References

[1] Benaissa, A., Bahri, A., El Allaoui, A., & Salahddine, M. A. (2023). Build a trained data of Tesseract OCR engine for Tifinagh script recognition. *Data and Metadata*, *2*, 185. https://doi.org/10.56294/dm2023185

[2] Handayani, L. F., *et al*. (2021). Sistem pengenalan dan pencocokan citra KTP elektronik Indonesia menggunakan OCR dan CNN. *Jurnal RESTI*, *5*. http://dx.doi.org/10.31544/jtera.v6.i1.2021.1-6

[3] Toha, M. R., & Triayudi, A. (2023). Penerapan membaca tulisan di dalam gambar menggunakan metode OCR berbasis website pada e-KTP. *Jurnal Ilmu dan Teknologi Pendidikan (JST Undiksha)*, *11*(1). https://doi.org/10.23887/jstundiksha.v11i1.42279

[4] Hudaya, M. M., Sa'adah, S., & Irawan, H. (2021). Implementasi verifikasi dan pencocokan gambar e-KTP menggunakan Faster R-CNN, ORB, dan KNN-BFM. *Jurnal RESTI*, *5*(4). https://doi.org/10.29207/resti.v5i4.3175

[5] Ibnutama, K., & Suryanata, M. G. (2020). Ekstraksi karakter citra menggunakan optical character recognition untuk pencetakan nomor kendaraan pada struk parkir. *Jurnal Media Informatika Budidarma*, *4*(4), 1119–1125. https://doi.org/10.30865/mib.v4i4.2432

[6] Memon, J., Sami, M., Khan, R. A., & Uddin, M. (2020). Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). *IEEE Access*, *8*, 142642–142668. https://doi.org/10.1109/ACCESS.2020.3012542

[7] Sugiarta, G., Andini, D. P., & Hidayatullah, S. (2021). Ekstraksi informasi/data e-KTP menggunakan optical character recognition convolutional neural network. *JTERA*, *6*(1), 1–6. https://doi.org/10.31544/jtera.v6.i1.2021.1-6

[8] Octaviani, T., Setiawan, H., & Kelana, O. H. (2023). Perbandingan pytesseract dan template matching untuk otomatisasi input data KTP. *Jurnal Buana Informatika*, *14*(2), 170–179. https://doi.org/10.24002/jbi.v14i02.7612

[9] Haris, M., Suryanata, M. G., & Yetri, M. (2023). Implementasi OCR menggunakan algoritma template matching correlation pada pengarsipan e-KTP. *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD*, *6*(2), 357–364. https://doi.org/10.53513/jsk.v6i2.8134

[10] Putra, N. R., Rachmadi, R. F., & Nugroho, S. M. S. (2023). Web service untuk ekstraksi informasi KTP menggunakan Google Cloud Vision. *Jurnal Teknik ITS*, *12*(3), A218–A223. https://doi.org/10.12962/j23373539.v12i3.122400

[11] Taqwa, F. S., *et al*. (2024). Perancangan sistem informasi pada Jambi sistem informasi pelatihan pertanian menggunakan optical character recognition. *Jurnal Sifo Mikroskil (JSM)*, *25*(1), 45–54. https://doi.org/10.55601/jsm.v25i1.1139

[12] Rusli, F. M., Adhiguna, K. A., & Irawan, H. (2021). Indonesian ID card extractor using optical character recognition and natural language post-processing. *arXiv*. https://arxiv.org/abs/2101.05214

[13] Angela, S. M., Eviyanti, A., & Mauliana, M. I. (2024). Pengembangan teknologi optical character recognition di Flutter berupa deteksi teks pada gambar. *Jurnal TEKINKOM*, *7*(1), 64–72. https://doi.org/10.37600/tekinkom.v7i1.1167

[14] Bahar, R. D. Y., & Raban, R. A. (2023). Model pendeteksi nominal uang kertas rupiah menggunakan teknologi optical character recognition. *Jurnal TICOM*, *12*(1), 17–24. https://doi.org/10.70309/ticom.v12i1.101

[15] Suhairi, M., Rahmi, E., & Kurniawaty, E. (2025). Penerapan teknologi OCR pada aplikasi pemindaian nutrisi di label kemasan makanan. *Jurnal Ilmiah Teknik Informatika dan Komunikasi*, *5*(1), 1–9. https://doi.org/10.55606/juitik.v5i1.1205

[16] Putri, W. P., *et al*. (2024). Implementasi OCR untuk klasifikasi penempatan obat berdasarkan kelas terapi di apotek. *Jurnal Elkolind*, *11*(2), 245–254. https://doi.org/10.33795/elkolind.v11i2.5341

[17] Reswan, Y., Raffles, R., *et al*. (2024). Penerapan algoritma OCR untuk ekstraksi informasi dari citra KTM. *JATI*, *8*(5), 2339–2346. https://doi.org/10.36040/jati.v8i5.11006

[18] Darajat, T. A., Pratondo, A., & Prasetyanto, F. (2023). Penerapan fitur face recognition dan ekstraksi data dari citra KTP. *e-Proceeding of Applied Science*, *9*(5), 3316–3325.

[19] Iskandar, R., & Kesuma, M. E. K. (2022). Designing a real-time-based OCR to detect ID cards. *International Journal of Electronics and Communications System*, *2*(1), 43–52. https://doi.org/10.24042/ijecs.v2i1.13108

[20] Anakpluek, *et al*. (2025). Improved Tesseract OCR performance on Thai document datasets. *Big Data Research*, *39*, 100508. https://doi.org/10.1016/j.bdr.2025.100508

[21] Nugroho, I. A., *et al.* (2024). The design of a C1 document data extraction application using Tesseract-OCR engine. *Jurnal RESTI*, *8*(1), 152–159. https://doi.org/10.29207/resti.v8i1.5151

[22] Zacharias, E., Teuchler, M., & Bernier, B. (2020). Image processing based scene-text detection and recognition with Tesseract. *arXiv*. https://doi.org/10.48550/arXiv.2004.08079

[23] Munawaroh, A., & Jamzuri, E. R. (2023). Automatic optical inspection for detecting keycaps misplacement using Tesseract OCR. *International Journal of Electrical and Computer Engineering*, *13*(5), 5147–5155. https://doi.org/10.11591/ijece.v13i5.pp5147-5155

[24] Kotwal, N., *et al.* (2021). Optical character recognition using Tesseract engine. *International Journal of Engineering Research & Technology*, *10*(9), 508–512. https://doi.org/10.17577/IJERTV10IS090157

[25] Boruah, P. K., & Haloi, P. (2025). Development and implementation of a custom license plate detection and recognition system using YOLOv10 and Tesseract OCR. *International Journal of Innovative Technology and Exploring Engineering*, *14*(6), 1–8. https://doi.org/10.35940/ijitee.E1083.14060525

[26] Aji, M. I. S., Mulyana, D. I., & Akbar, Y. (2023). Penerapan IoT dengan algoritma fuzzy dalam monitoring kesehatan mata berbasis Android. *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD*, *6*(1), 135–144. https://doi.org/10.53513/jsk.v6i1.7346

[27] Schmidt, T., Kamlah, J., & Weil, S. (2024). Reichsanzeiger-GT: An OCR ground truth dataset based on the historical newspaper Deutscher Reichsanzeiger und Preußischer Staatsanzeiger (1819–1945). *Data in Brief*, *54*, 110274. https://doi.org/10.1016/j.dib.2024.110274

[28] Ponnuru, M., Ponmalar, S., Likhitha, A., Sree, T. B., & Chaitanya, G. G. (2024). Image-based extraction of prescription information using OCR-Tesseract. *Procedia Computer Science*, *235*, 1077–1086. https://doi.org/10.1016/j.procs.2024.04.102

[29] Robby, A. G., Tandra, A., Susanto, I., Harefa, J., & Chowanda, A. (2019). Implementation of optical character recognition using Tesseract with the Javanese script target in Android application. *Procedia Computer Science*, *157*, 499–505. https://doi.org/10.1016/j.procs.2019.09.006