Enhanced OCR Recognition for Madurese Text Documents: A Genetic Algorithm Approach with Tesseract 5.5

Mohammad Nazir Arifin[1*], Muhammad Umar Mansyur[2], Ali Rahman[3], Nindian Puspa Dewi[4], Fauzan Prasetyo Eka Putra[5]

[1,2,3,4,5]*Universitas Madura, Indonesia*

*corr-author: nazir@unira.ac.id

Abstract—Character Recognition (OCR) for the Madurese language using Genetic Algorithms (GA). The study addresses the challenges in processing Madurese text documents by implementing a nine-step image preprocessing workflow optimized through GA. Our methodology combines rescaling, grayscale conversion, adaptive thresholding, deskewing, median blur, Otsu thresholding, border removal, contrast enhancement, and noise reduction, with the sequence determined by GA optimization. The system utilizes Tesseract 5.5 OCR engine configured with Vietnamese language model parameters to accommodate Maderese writing characteristics. Experiments conducted on a dataset of 500 images demonstrated significant improvements in recognition accuracy. The GA-optimized preprocessing sequence achieved a 24.32% Word Error Rate (WER) and 7.47% Character Error Rate (CER), marking substantial improvements over the baseline Tesseract implementation. Further optimization through language model selection, particularly using the Occitan (OCI) model, yielded 100% accuracy in specific test cases. The research also explored various fitness function configurations, with a 0.7:0.3 WER-to-CER ratio proving most effective. These results demonstrate the potential of GA optimization in enhancing OCR performance for regional languages with unique characteristics, contributing to the broader field of document digitization and language preservation.

Keywords: Image Preprocessing, Optical Character Recognition, Genetic Algorithm Optimization, Madurese Language Processing, Tesseract OCR

I. INTRODUCTION

Optical Character Recognition (OCR) has become an essential part of various image processing and document digitization applications. With the advancement of technology, OCR is increasingly used to process documents in various languages, including those with unique or uncommon characteristics [1,2]. One example is the Madurese language. Madurese has a unique writing system and distinctive letter characteristics, such as bhisat, kapeng, and other special symbols. As a result, handling Optical Character Recognition (OCR) for language requires a specialized approach in the character recognition process [3].

Text processing in the Madurese language through OCR still faces various challenges, such as variations in writing styles, inconsistencies in image quality, and significant noise interference in the images. Additional challenges arise at the preprocessing stage, where the number of preprocessing steps applied to an image can result in over 2,000 combinations. This complexity necessitates optimization due to the variations in the images. Various approaches have been developed to enhance OCR performance, including edge detection methods, noise reduction techniques, and contrast enhancement [4,5,6,7].

This study proposes a novel approach to image processing for Madurese language OCR by utilizing nine optimized image processing steps through a Genetic Algorithm (GA) [8]. These steps include image rescaling, grayscale conversion, adaptive thresholding, deskewing, median blurring, Otsu thresholding, border removal using contour detection, contrast enhancement with CLAHE, and noise reduction. Optimizing the sequence of these steps aims to significantly improve image quality, which in turn can enhance the accuracy of the OCR system in recognizing Madurese text.

One of the key innovations of this study is the application of a Genetic Algorithm (GA) to determine the optimal sequence of image processing rules that can reduce text recognition errors, measured by metrics such as Word Error Rate (WER), Character Error Rate (CER), and Levenshtein Distance. By leveraging GA, the study not only optimizes the sequence of image processing steps but also minimizes the computational cost associated with evaluating all possible rule combinations.

This research is expected to make a significant contribution to the development of OCR systems for the

Madurese language, particularly in supporting document digitization and automatic translation within a broader social and cultural context. The lack of digitization efforts for the Madurese language can negatively impact its preservation and dissemination. Additionally, this gap in digitalization has resulted in limited research on the Madurese language, leaving it less known and less studied. This situation hinders efforts to preserve and develop the language for future generations.

II. METHODS

This study comprises six main stages, as illustrated in Fig. 1. The first stage involves data collection. Subsequently, the collected data undergoes an optimization phase using the Genetic Algorithm (GA) to determine the best rules for image processing. These rules encompass nine steps, including rescaling to 300 DPI, grayscale conversion, adaptive Gaussian thresholding, deskewing using the Hough Transform, median blurring, Otsu thresholding, border removal with contour detection, contrast enhancement using CLAHE, and noise reduction with FastNlMeansDenoisingColored.

Once the optimal rules are identified by GA, the images are processed according to these rules. The next stage involves Optical Character Recognition (OCR) using Tesseract 5.5. The results of OCR are then evaluated using metrics such as Word Error Rate (WER), Character Error Rate (CER), and Levenshtein Distance to measure the accuracy of text extraction. The final stage is result analysis, aimed at assessing system performance and providing insights into the effectiveness of the methods used in this study.

The first step in this study is the data collection process, which involves gathering images containing text in the Madurese language. These image data are sourced from various Madurese reading materials, such as books, magazines, and other scanned documents. This data collection aims to obtain a diverse and representative set of Madurese text from different types of written media.

Once the images are collected, the next step is manual transcription of the text present in the images to create ground truth data. This transcription is carefully performed to ensure an accurate representation of the text. Each transcription is then verified by two assessors who are proficient in the Madurese language, to ensure accuracy and consistency in the ground truth writing. The verification process involves comparing the transcription results from both assessors and reaching a consensus on any discrepancies found. The outcome of this verification is used as a reference to ensure the validity of the ground truth data that will be used in subsequent research. Fig. 2 shows a sample of the collected dataset.

A. Data Collection

The first step in this study is the data collection process, which involves gathering images containing text in the Madurese language. These image data are sourced from various Madurese reading materials, such as books, magazines, and other scanned documents. This data collection aims to obtain a diverse and representative set of Madurese text from different types of written media.

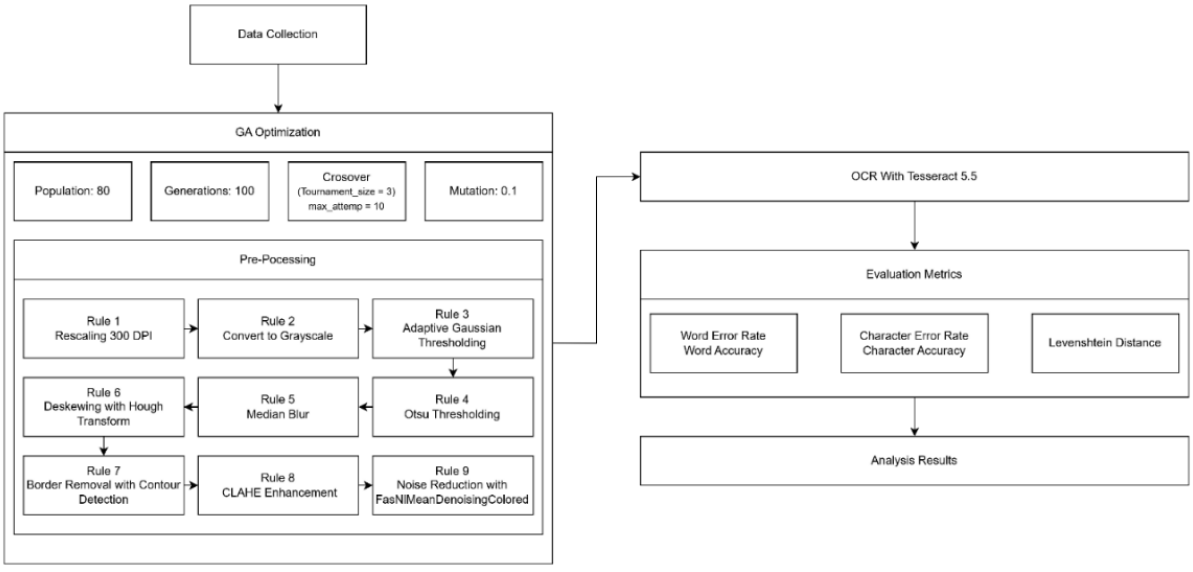
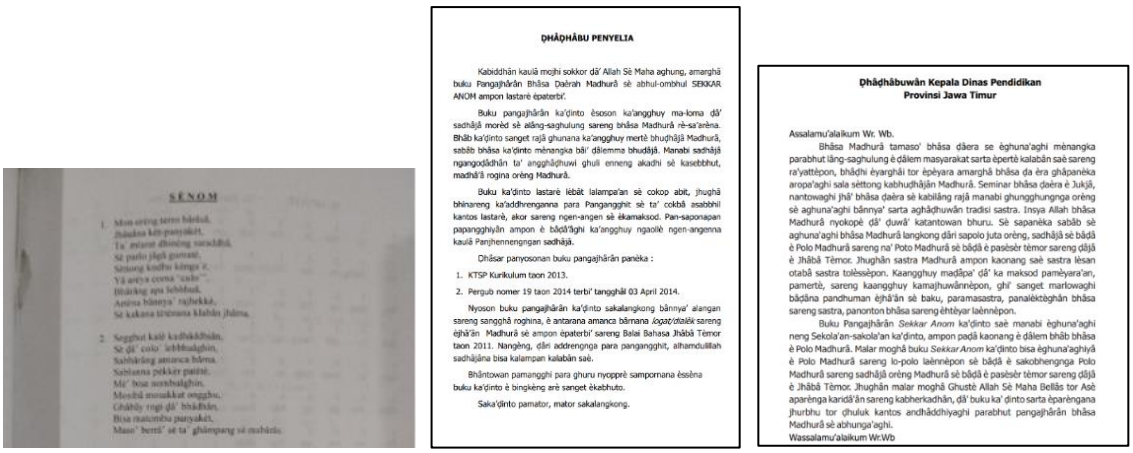


Fig. 1 Research plan



(a) (b) (c)

Fig. 2 Sample dataset

Once the images are collected, the next step is manual transcription of the text present in the images to create ground truth data. This transcription is carefully performed to ensure an accurate representation of the text. Each transcription is then verified by two assessors who are proficient in the Madurese language, to ensure accuracy and consistency in the ground truth writing. The verification process involves comparing the transcription results from both assessors and reaching a consensus on any discrepancies found. The outcome of this verification is used as a reference to ensure the validity of the ground truth data that will be used in subsequent research. Fig. 2 shows a sample of the collected dataset.

B. GA Optimization

After the dataset is collected and manual transcription (ground truth) is verified, the next step is optimization using the Genetic Algorithm (GA). The GA is applied to the nine rules defined according to Fig. 1 to find the optimal sequence of rules. Given the number of rules, the possible combinations that can be formed are 9! (9 factorial), which amounts to 362,880 different combinations. Testing all these combinations on the dataset would be highly inefficient in terms of time and computational cost, especially considering the large number of combinations and the size of the dataset that needs to be processed. Based on literature review and previous studies, the Genetic Algorithm has proven to be effective and widely used in optimizing large datasets with relatively efficient computational costs [9 -16].

The advantages of GA in this case include the ability to efficiently explore a vast search space without having to test every possible combination. The natural selection mechanism allows for convergence toward an optimal solution. GA's flexibility in representing and optimizing the sequence of rules through genetic operators enables effective search. Additionally, GA can avoid local optima through mutation processes and has good scalability for large datasets [17,18,19]. The GA implementation for rule optimization will involve several key components, such as a suitable chromosome representation for the rule sequence, a fitness function that measures the effectiveness of a rule combination, genetic operators (crossover and mutation) designed specifically for permutation problems, and GA parameters including population size, crossover probability, mutation probability, and termination criteria.

The GA implementation in this study is based on several configuration values shown in Table I. These parameters are chosen based on literature review and preliminary experiments to ensure a balance between exploring the search space and the algorithm's convergence speed.

TABLE I
GA CONFIGURATION

Parameter	Nilai
Maximum Generations	100
Tournament Size	3
Max Attempt	10
Mutation Rate	0.1
Sequence Length	9 Rules
Selection	Tournament Selection
Crossover	Single-point binary
Elitism	1 Best Individual
Fitness Function	$\alpha(1 - WER)$ $+ \beta(1 - CER)$

The selection of these parameters takes into account the characteristics of the rule optimization problem at hand, where a population size of 80 individuals provides sufficient genetic diversity [20], Meanwhile, a maximum of 100 generations provides ample opportunity for the solution to converge [21,22,23]. The fitness function that combines Word Error Rate (WER) and Character Error Rate (CER) with weights. α and β enable balanced optimization between word-level and character-level accuracy, with the constraint that $\alpha + \beta = 1$.

1) Rule 1 – Rescaling: The target DPI is set to 300, as it is the recommended standard for the Tesseract OCR engine[24,25]. Linear interpolation is chosen as the default method in OpenCV because it provides good results for most cases without requiring heavy computation. The scaling factor is calculated using (1).

$$\text{Scaling Factor} = \frac{Target\ DPI}{Original\ DPI} \quad (1)$$

Where
Target DPI : 300
Original DPI : 96

2) Rule 2 – Convert to Grayscale: The parameter color space BGR2GRAY is used because OpenCV reads images by default in BGR format (Blue, Green, Red). The conversion results in a single-channel image, where each pixel is represented by a single intensity value, unlike RGB images which have three channels. An 8-bit depth is chosen because it provides a sufficient range of values (0-255) to represent grayscale intensity variations in most document image processing cases [26]. A range of 0 represents absolute black, while 255 represents absolute white.

3) Rule 3 - Adaptive Gaussian Thresholding: This creates a binary image with a max value of 255 for the brightest pixels. A block size of 11 balances local detail and noise, while a constant C of 2 adjusts the threshold for robustness. ADAPTIVE_THRESH_GAUSSIAN_C uses Gaussian weighting for smoother results, and THRESH_BINARY sets pixels above the threshold to 255 and below to 0.

4) Rule 4 - Otsu Thresholding: The threshold starts at 0 and is automatically set by the Otsu algorithm based on the image's histogram. With THRESH_BINARY + THRESH_OTSU, pixels above the threshold become 255, and those below become 0, using a max value of 255.

5) Rule 5 - Median Blur: A 3x3 kernel removes salt-and-pepper noise while preserving details. The odd-sized kernel ensures a central pixel, balancing noise reduction and detail retention.

6) Rule 6 - Deskewing with Hough Transform: An angle range of ±30° covers most document tilts. Min line length (width/4.0) ensures significant lines are detected, max line gap (height/4.0) accommodates text spacing, and a threshold of 30 balances line detection and noise resistance.

7) Rule 7 - Border Removal with Contour Detection: RETR_EXTERNAL retrieves only outer contours efficiently, and CHAIN_APPROX_SIMPLE saves resources by storing contour endpoints. A threshold of 128 separates content from background effectively.

8) Rule 8 - CLAHE Enhancement: A clip limit of 2.0 prevents excessive noise amplification, and an (8,8) tile grid size balances local and global contrast without artifacts.

9) Rule 9 - Noise Reduction with FastNlMeansDenoisingColored: Parameters h luminance and h color are set to 10 for filter strength. A 7x7 template window and 21x21 search window balance denoising quality and computation time.

C. OCR With Tesseract

Text recognition is performed using Tesseract OCR version 5.5, which is well-known for its superior performance in handling documents with complex characteristics. To achieve optimal recognition results, Tesseract is configured with Page Segmentation Mode (PSM) 6, which assumes the input document consists of structured and uniform text blocks. This mode is suitable for documents with consistent text layouts, such as book pages or official documents. The engine mode used is a combination of Legacy + LSTM, a classical approach combined with Long Short-Term Memory (LSTM) deep learning. This approach allows Tesseract to leverage the strengths of both methods, resulting in more precise text recognition, particularly for documents with unique characteristics. During the model training phase to determine the golden rule, no parameters are provided to Tesseract. This ensures that Tesseract does not influence the rule configuration. Parameter configuration will be set after the golden rule is found.

D. Evaluation Metric

To evaluate OCR performance in recognizing Madurese text, WER and CER are used as fitness functions. WER assesses word-level accuracy, while CER measures character-level errors. Additional metrics include Accuracy (overall correctness), Precision (correct recognitions among detected text), Recall

(ability to capture correct text), and F1-Score (balance between precision and recall).

E. Analysis Result

This stage is the final part of the research, where the results are analyzed based on evaluation metrics to measure the OCR system's ability to recognize Madurese text. A high accuracy and precision indicate strong performance, while a good F1-Score shows a balanced relationship between precision and recall. However, a high WER or CER suggests frequent recognition errors. Low recall or accuracy indicates the system struggles to recognize text correctly. Based on these findings, researchers can determine improvements, such as model modifications, better character detection, or enhanced training data.

III. RESULTS AND DISCUSSION

In this section, the results of the model training process will be explained, leading to the testing phase and comparisons between using genetic algorithms for optimizing OCR Tesseract and OCR Tesseract without any optimization. The testing results will be presented in the form of evaluation metrics, including WER (Word Error Rate), CER (Character Error Rate), Accuracy, Precision, Recall, and F1-Score.

A. GA Model Training Test Results

The training process uses 500 images and corresponding ground truth data, with an 80-20 split for training and testing. The fitness function is tested in three configurations: 0.7 × WER + 0.3 × CER, 0.3 × WER + 0.7 × CER, and 0.5 × WER + 0.5 × CER. WER and CER values are obtained from Tesseract OCR without parameter adjustments. Results from 15 GA model trials (Table II) show that fitness function weight variations significantly impact WER and CER.

From the 15 experiments conducted, there is significant variation in the Word Error Rate (WER) and Character Error Rate (CER) across three different weight configurations (0.3:0.7, 0.7:0.3, and 0.5:0.5). Experiments 1-5 use the weight configuration 0.3:0.7, where WER values range from 41.08% to 55.68% and CER values range from 12.92% to 15.15%. In experiments 6-10 with the weight configuration 0.7:0.3, there is a noticeable performance improvement, with the lowest WER reaching 24.32% in Experiment 7, which is the best result across all experiments. The implementation of the nine preprocessing rules optimized by the Genetic Algorithm (GA) plays a key role in achieving these results. As shown in Table II, the optimal sequence from Experiment 7 (Rule 7: Border Removal, Rule 8: CLAHE Enhancement, Rule 1: Rescaling, Rule 5: Median Blur, Rule 9: Noise Reduction, Rule 4: Otsu Thresholding, Rule 6: Deskewing, Rule 2: Grayscale Conversion) effectively enhances image quality for Madurese text recognition. Border removal (Rule 7), as the initial step, eliminates non-text elements from the document edges, as observed in Fig. 3(b), where the colorful background is removed. CLAHE enhancement (Rule 8) further improves text contrast, enabling Tesseract to better recognize characters, which contributes to a reduction in CER to 7.47%. Rescaling to 300 DPI (Rule 1) aligns the image resolution with Tesseract's standards, while Median Blur (Rule 5) reduces noise without blurring character details. Rule 9 (Noise Reduction) and Rule 4 (Otsu Thresholding) smooth the image and create a sharper binary separation

TABLE II
EXPERIMENTAL RESULT OF TRAINING MODEL

Experiment	WER (%)	CER (%)	Weight (WER:CER)	Golden Rule Sequence
1	55,68	13,71	0.3 : 0.7	2,8,5,1,9,7,6,4
2	55,14	15,15	0.3 : 0.7	9,1,5,7,6,4,8,2
3	54,59	13,21	0.3 : 0.7	5,8,6,2,9,1,7,4
4	54,05	14,29	0.3 : 0.7	4,6,9,1,8,5,7,2
5	41,08	12,92	0.3 : 0.7	8,9,7,6,1,5,2,4
6	40,54	12,78	0.7 : 0.3	1,5,2,9,4,6,7,8
7	24,32	7,47	0.7 : 0.3	7,8,1,5,9,4,6,2
8	37,84	12,2	0.7 : 0.3	6,2,8,5,1,7,4,9
9	37,3	12,56	0.7 : 0.3	8,7,1,5,4,9,6,2
10	37,3	12,2	0.7 : 0.3	7,8,5,1,9,4,6,2
11	36,76	11,92	0.5 : 0.5	7,8,1,5,9,4,2,6
12	35,68	10,91	0.5 : 0.5	8,7,1,5,9,4,6,2
13	27,57	9,19	0.5 : 0.5	7,1,8,5,9,4,6,2
14	37,84	12,2	0.5 : 0.5	7,8,1,5,4,9,6,2
15	56,22	13,78	0.5 : 0.5	7,8,1,9,5,4,6,2

between text and background. Deskewing (Rule 6) corrects document tilt, which proves critical for images like Fig. 3(a), and Grayscale Conversion (Rule 2) simplifies pixel data for further processing. This sequence, optimized by GA from 362,880 possible combinations, significantly reduces WER from 55.68% (Experiment 1) to 24.32% (Experiment 7), demonstrating that the selection and ordering of rules greatly influence OCR performance.

Further analysis of Fig. 4(c) indicates that Rule 6 (Deskewing) causes a temporary spike in WER, likely due to the tilt correction revealing previously hidden text, which is subsequently refined by rules such as Rule 9 (Noise Reduction). This highlights the importance of Deskewing for tilted documents, followed by refinement steps to ensure optimal final results.

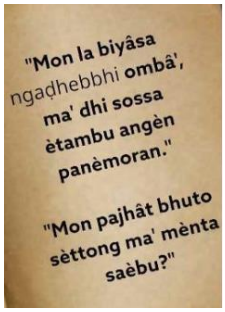
B. Testing Results and Evaluation Metrics

In the testing process, training data with a pixel dimension of 256 x 356, as shown in Fig. 3(a), is used as a sample. Fig. 3(a) is chosen due to its slightly tilted position and the presence of a colorful background. The testing results can be seen in Table III.

Based on the testing results shown in the Table III, there is a significant difference between the use of Tesseract OCR with default settings or applying various rules (RULE_1 to RULE_9) compared to the implementation of Tesseract optimized using Genetic Algorithm (GA). In all tests with default settings and applied rules (RULE_1 to RULE_9), the system produces a value of 0.00 for all evaluation metrics, including WER (Word Error Rate), CER (Character Error Rate), Precision, Accuracy, Recall, and F1-Score. This result indicates that the system fails to recognize text under those conditions. The tested image condition is image (a) from Fig. 3.

TABLE III
TESTING RESULT FOR TESSERACT

Categories	WER	CER	Precision	Accuracy	Recall	F1-Score
Tesseract Default	0.00	0.00	0.00	0.00	0.00	0.00
Tesseract + RULE_1	0.00	0.00	0.00	0.00	0.00	0.00
Tesseract + RULE_2	0.00	0.00	0.00	0.00	0.00	0.00
Tesseract + RULE_3	0.00	0.00	0.00	0.00	0.00	0.00
Tesseract + RULE_4	0.00	0.00	0.00	0.00	0.00	0.00
Tesseract + RULE_5	0.00	0.00	0.00	0.00	0.00	0.00
Tesseract + RULE_7	0.00	0.00	0.00	0.00	0.00	0.00
Tesseract + RULE_8	0.00	0.00	0.00	0.00	0.00	0.00
Tesseract + RULE_9	0.00	0.00	0.00	0.00	0.00	0.00
Tesseract + GA	22.22	3.81	75.00	75.00	70.59	72.73



(a)

(b)

Output Tesseract + GA

(c)

Fig. 3 Testing image before and after prepocessing

When Tesseract is optimized using the Genetic Algorithm (GA), the system shows significant performance improvement with a WER of 22.22% and a CER of 3.81%. At this stage, the image undergoes preprocessing, resulting in data as shown in Fig. 3(b). The system also achieves a good level of precision and accuracy, at 75.00%, with a recall value of 70.59%, indicating the system's ability to recognize relevant text. Additionally, the F1-Score reaching 72.73% demonstrates a good balance between precision and recall. This result confirms that optimization using the Genetic Algorithm provides significant improvements in OCR system performance compared to using Tesseract without optimization or with predefined rules.

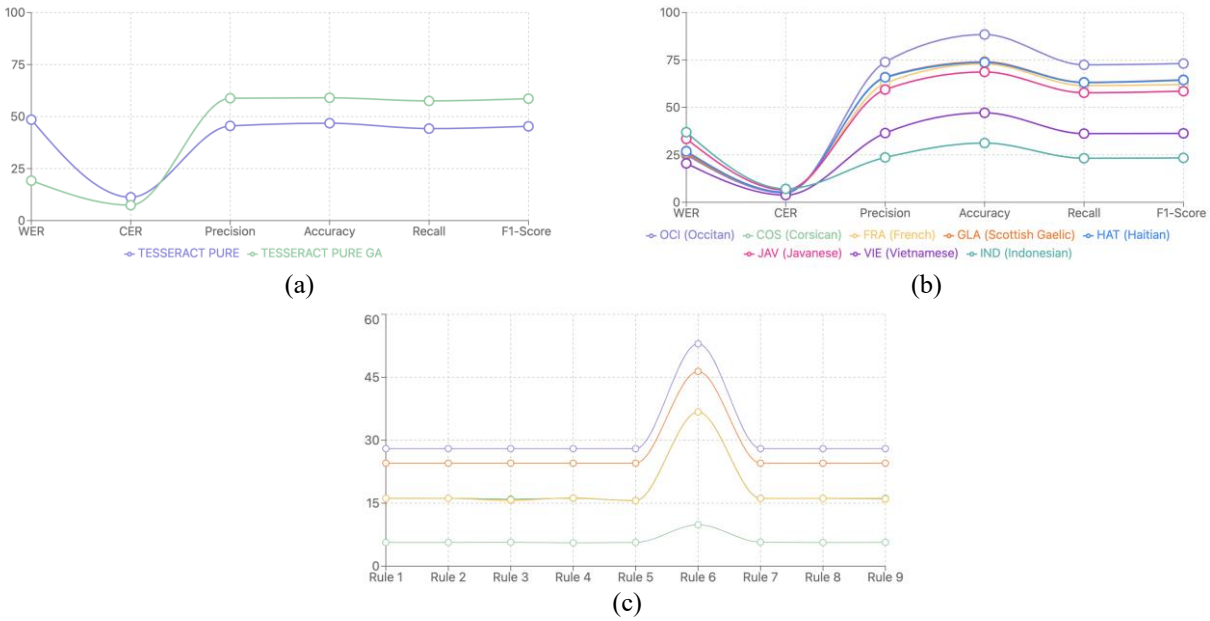
Next, Table IV shows the testing results with Tesseract configured using various language models. Given that Tesseract language models cover more than 100 languages, this study only displays 5 data points with the most optimal WER and CER values. Table IV presents the testing results with the combination of GA optimization applied to Tesseract that is already configured.

Based on the evaluation metric values shown in Table IV, there is a significant increase in accuracy. From the initial result of 70.59%, accuracy improved to 100% with the OCI (Occitan) language model. Other languages, such as COS (Corsican), FRA (French), GLA (Scottish Gaelic), and HAT (Haitian Creole), also demonstrate excellent performance, with WER (Word Error Rate) ranging from 16.67% to 2.86%, and accuracy reaching up to 81.25%. In this case, optimizing performance in terms of WER and CER can be enhanced through selecting the appropriate language model.

Additionally, researchers used test data on 19 images, resulting in an average performance as shown in the graph.

TABLE IV
TESTING RESULTS WITH GA OPTIMIZATION AND TESSERACT CONFIGURATION

Language Models	WER	CER	Precision	Accuracy	Recall	F1-Score
OCI	0.00	0.00	93.75	100.00	88.24	90.91
COS	16.67	2.86	81.25	81.25	76.47	78.79
FRA	16.67	2.86	81.25	81.25	76.47	78.79
GLA	16.67	2.86	81.25	81.25	76.47	78.79
HAT	16.67	2.86	81.25	81.25	76.47	78.79



(a) (b)

(c)

Fig. 4 Training results of 19 training data

Based on the graph in Fig. 4(a), the performance comparison between TESSERACT PURE and TESSERACT PURE GA reveals some interesting patterns that highlight the significant impact of Genetic Algorithm (GA) optimization. In terms of the Word Error Rate (WER) metric, TESSERACT PURE has a value of around 48%, indicating that nearly half of the recognized words experience errors. In contrast, TESSERACT PURE GA shows a significant improvement, with WER dropping to approximately 20%, indicating a drastic reduction in word recognition errors. However, for the Character Error Rate (CER) metric, both models perform relatively similarly, at around 10%. This suggests that GA optimization is more effective in word pattern recognition than in character recognition, where CER is more influenced by the quality of the input images.

A striking difference is seen in metrics such as Precision, Accuracy, Recall, and F1-Score. The Precision of TESSERACT PURE GA reaches about 60%, which is much higher than TESSERACT PURE's 45%. This shows that TESSERACT PURE GA is better at producing accurate text compared to the total text detected. Similar patterns are observed in Accuracy, where TESSERACT PURE GA achieves around 60%, while TESSERACT PURE only reaches 47%. For Recall and F1-Score, TESSERACT PURE GA records higher values, around 58% for both, compared to TESSERACT PURE's 45%. Overall, GA optimization consistently improves performance across various important metrics, especially in word recognition and error reduction.

Based on the graph in Fig. 4(b), language model comparison reveals that OCI (Occitan) performs the best with an accuracy of around 75% and a precision of about 75%. Other European languages like COS (Corsican), FRA (French), GLA (Scottish Gaelic), and HAT (Haitian) show relatively consistent performance, with accuracy and precision ranging from 65-70%. For Southeast Asian languages, JAV (Javanese) shows good performance with precision and accuracy around 60%. On the other hand, VIE (Vietnamese) performs lower, with accuracy around 47% and precision around 36%. The IND (Indonesian) model has the poorest performance, with an accuracy of around 31% and precision of 23%. In terms of error rate, all models show WER values between 20-36% and relatively low CERs below 10%. Interestingly, while VIE has the lowest WER (around 20%) and CER, its performance in other metrics like precision, accuracy, recall, and F1-Score is below the average of European language models. This suggests that low error rates do not always correlate with higher performance in other evaluation metrics.

According to the graph in Fig. 4(c), an interesting pattern emerges when comparing different rules. The values tend to remain stable from Rule 1 to Rule 5, with the top blue line around 28%, the orange line around 25%, the yellow line around 15%, and the green line the lowest at 5%. Dramatic changes occur at Rule 6, where all lines show significant increases. The blue line peaks at around 55%, followed by the orange line at 45%, the yellow line at 35%, and the green line reaches around 10%. After Rule 6, the values start to decline in Rule 7 and stabilize again up to Rule 9, returning to levels similar to the initial condition. The blue and orange lines return to around 28% and 25%, the yellow line to 15%, and the green line to 5%.

IV. CONCLUSION

Our research demonstrates the effectiveness of genetic algorithm optimization in improving OCR accuracy for Maderese language documents. The implementation of GA-optimized preprocessing sequences, combined with appropriate language model selection, significantly enhanced recognition performance compared to standard Tesseract OCR implementations. The study achieved notable improvements in key metrics, with WER reduction to 24.32% and CER to 7.47% using optimal GA configurations. The success of the Occitan language model in achieving 100% accuracy in specific cases suggests promising directions for future research in cross-lingual OCR adaptation. These findings contribute valuable insights to the field of OCR optimization for regional languages and provide a framework for future developments in document digitization technologies. The methodology developed in this study can be adapted for other regional languages with similar characteristics, potentially advancing the preservation and accessibility of cultural and linguistic heritage through digital means.

ACKNOWLEDGMENT

This research was fully funded by the Institute for Research and Community Service (LPPM) of Universitas Madura under the 2024-2025 Research Grant scheme. The authors would like to express their sincere gratitude to LPPM Universitas Madura for their generous financial support and continued commitment to advancing technological research in regional language preservation. We also appreciate the technical support and facilities provided by the university throughout the research period. Special thanks to all the research assistants and language experts who contributed to the development and validation of our dataset.

REFERENCES

[1] K. Thammarak, P. Kongkla, Y. Sirisathitkul, and S. Intakosum, "Comparative analysis of Tesseract and Google Cloud Vision for Thai vehicle registration certificate," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 2, pp. 1849–1858, 2022, doi: 10.11591/ijece.v12i2.pp1849-1858.

[2] M. Aviles, L. M. Sánchez-Reyes, R. Q. Fuentes-Aguilar, D. C. Toledo-Pérez, and J. Rodríguez-Reséndiz, "A Novel Methodology for Classifying EMG Movements Based on SVM and Genetic Algorithms," *Micromachines (Basel)*, vol. 13, no. 12, 2022, doi: 10.3390/mi13122108.

[3] A. Nuzulia, "Peningkatan Kemampuan Berbahasa Madura Yang Baik dan Benar Pada Masyarakat Dusun Banlanjang Tlonto Raja Kecamatan Pasean di Masjid Al Muttaqin," *Angewandte Chemie International Edition, 6(11), 951–952.*, vol. 1, no. 1, pp. 5–24, 2019.

[4] T. Hegghammer, "OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment," *J Comput Soc Sci*, vol. 5, no. 1, pp. 861–882, 2022, doi: 10.1007/s42001-021-00149-1.

[5] I. N. T. Lestari and D. I. Mulyana, "Implementation of Ocr (Optical Character Recognition) Using Tesseract in Detecting Character in Quotes Text Images," *Journal of Applied Engineering and Technological Science*, vol. 4, no. 1, pp. 58–63, 2022, doi: 10.37385/jaets.v4i1.905.

[6] V. E. Bugayong, J. Flores Villaverde, and N. B. Linsangan, "Google Tesseract: Optical Character Recognition (OCR) on HDD / SSD Labels Using Machine Vision," *2022 14th International Conference on Computer and Automation Engineering, ICCAE 2022*, pp. 56–60, 2022, doi: 10.1109/ICCAE55086.2022.9762440.

[7] A. Shanthakumari, R. Kalpana, J. Jayashankari, B. Umamaheswari, and M. Sirija, "Mask RCNN and Tesseract OCR for vehicle plate character recognition," *AIP Conf Proc*, vol. 2393, 2022, doi: 10.1063/5.0074442.

[8] R. Widianti, S. Surono, and K. I. Ibraheem, "Handling Noise Data with PCA Method and Optimization Using Hybrid Fuzzy C-Means and Genetic Algorithm," *JUITA : Jurnal Informatika*, vol. 12, no. 2, pp. 141–147, 2024.

[9] Z. H. Ahmed, A. S. Hameed, and M. L. Mutar, "Hybrid Genetic Algorithms for the Asymmetric Distance-Constrained Vehicle Routing Problem," *Math Probl Eng*, vol. 2022, 2022, doi: 10.1155/2022/2435002.

[10] A. Anwaar, A. Ashraf, W. H. K. Bangyal, and M. Iqbal, "Genetic Algorithms: Brief review on Genetic Algorithms for Global Optimization Problems," *Proceedings - 2022 International Conference on Human-Centered Cognitive Systems, HCCS 2022*, 2022, doi: 10.1109/HCCS55241.2022.10090327.

[11] M. Zeinali, G. Rahimi, and S. Hosseini, "Optimizing buckling load of sandwich plates with cutouts using artificial neural networks and genetic algorithms," *Mechanics Based Design of Structures and Machines*, vol. 52, no. 9, pp. 6173–6190, 2024, doi: 10.1080/15397734.2023.2272679.

[12] H. Naseri, A. Fani, and A. Golroo, "Toward equity in large-scale network-level pavement maintenance and rehabilitation scheduling using water cycle and genetic algorithms," *International Journal of Pavement Engineering*, pp. 1–13, 2020, doi: 10.1080/10298436.2020.1790558.

[13] V. Skorpil and V. Oujezsky, "Parallel Genetic Algorithms' Implementation Using a Scalable Concurrent Operation in Python†," *Sensors*, vol. 22, no. 6, 2022, doi: 10.3390/s22062389.

[14] R. Peña-García, R. D. Velázquez-Sánchez, C. Gómez-Daza-Argumedo, J. O. Escobedo-Alva, R. Tapia-Herrera, and J. A. Meda-Campaña, "Physics-Based Aircraft Dynamics Identification Using Genetic Algorithms," *Aerospace*, vol. 11, no. 2, 2024, doi: 10.3390/aerospace11020142.

[15] Z. Guo, Y. Wang, S. Zhao, T. Zhao, and M. Ni, "Modeling and optimization of micro heat pipe cooling battery thermal management system via deep learning and multi-objective genetic algorithms," *Int J Heat Mass Transf*, vol. 207, 2023, doi: 10.1016/j.ijheatmasstransfer.2023.124024.

[16] B. M. Achmad, S. Sa, and I. Kurniawan, "LSTM Algorithm in Predicting Chronic Kidney Disease Optimized Using Genetic Algorithm," *JUITA : Jurnal Informatika*, vol. 12, no. 2, pp. 243–253, 2024.

[17] G. P. Salachoris, G. Standoli, M. Betti, G. Milani, and F. Clementi, "Evolutionary numerical model for cultural heritage structures via genetic algorithms: a case study in central Italy," *Bulletin of Earthquake Engineering*, vol. 22, no. 7, pp. 3591–3625, 2024, doi: 10.1007/s10518-023-01615-z.

[18] V. Singh, R. Mehra, K. B. Ramesh, P. Srivastava, and A. Mishra, "Treatment of carpet and textile industry effluents using Diplosphaera mucosa VSPA: A multiple input optimisation study using artificial neural network-genetic algorithms," *Bioresour Technol*, vol. 387, 2023, doi: 10.1016/j.biortech.2023.129619.

[19] D. Carreres-Prieto, J. Ybarra-Moreno, J. T. García, and J. F. Cerdán-Cartagena, "A Comparative analysis of neural networks and genetic algorithms to characterize wastewater from led spectrophotometry," *J Environ Chem Eng*, vol. 11, no. 3, 2023, doi: 10.1016/j.jece.2023.110219.

[20] M. Elyasi, M. E. Simitcioğlu, A. Saydemir, A. Ekici, O. Ö. Özener, and H. Sözer, "Genetic algorithms and heuristics hybridized for software architecture recovery," *Automated Software Engineering*, vol. 30, no. 2, 2023, doi: 10.1007/s10515-023-00384-y.

[21] F. Ye, C. Doerr, H. Wang, and T. Bäck, "Automated Configuration of Genetic Algorithms by Tuning for Anytime Performance: Hot-off-the-Press Track at GECCCO 2022," *GECCO 2022 Companion - Proceedings of the 2022 Genetic and Evolutionary Computation Conference*, pp. 51–52, 2022, doi: 10.1145/3520304.3534075.

[22] F. García-Gutierrez *et al.*, "GA-MADRID: design and validation of a machine learning tool for the diagnosis of Alzheimer's disease and frontotemporal dementia using genetic algorithms," *Med Biol Eng Comput*, vol. 60, no. 9, pp. 2737–2756, 2022, doi: 10.1007/s11517-022-02630-z.

[23] Z. Zou, B. Wang, X. Hu, Y. Deng, H. Wan, and H. Jin, "Enhancing requirements-to-code traceability with GA-XWCoDe: Integrating XGBoost, Node2Vec, and genetic algorithms for improving model performance and stability," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 8, 2024, doi: 10.1016/j.jksuci.2024.102197.

[24] B. Wang, Q. Xu, Z. Bian, and Y. You, "Tesseract: Parallelize the Tensor Parallelism Efficiently," *ACM International Conference Proceeding Series*, 2022, doi: 10.1145/3545008.3545087.

[25] P. Lertsawatwicha, P. Phathong, N. Tantasanee, K. Sarawutthinun, and T. Siriborvornratanakul, "A novel stock counting system for detecting lot numbers using Tesseract OCR," *International Journal of Information Technology (Singapore)*, vol. 15, no. 1, pp. 393–398, 2023, doi: 10.1007/s41870-022-01107-4.

[26] I. H. Al amin and A. Aprilino, "Implementasi Algoritma Yolo Dan Tesseract Ocr Pada Sistem Deteksi Plat Nomor Otomatis," *Jurnal Teknoinfo*, vol. 16, no. 1, p. 54, 2022, doi: 10.33365/jti.v16i1.1522.