WSR-LA: An Improved Method for Post-OCR Text Correction in Video

Mukhamad Nurkamid[1], Nur Rokhman[2,*], Khabib Mustofa[3]

[1]*Department of Informatics Engineering, Universitas Muria Kudus, Jl. Lingkar Utara UMK, Gondangmanis, Bae, Kudus and 59327, Indonesia*

[1,2,3]*Department of Computer Science and Electronics, Universitas Gadjah Mada, Bulaksumur Caturtunggal, Sleman, Yogyakarta and 55281, Indonesia*

(Received: May 10, 2026; Revised: July 15, 2026; Accepted: September 27, 2026; Available online: October 29, 2026)

Abstract

Extracting textual information from video recordings remains challenging due to variations in the size, layout, and visual orientation of each image frame, which can lead to character recognition and word segmentation errors in Optical Character Recognition (OCR). This study aims to develop a post-processing framework for improving the quality of OCR-generated text by addressing over-segmentation and minor character recognition errors. The main contribution is the Word Segmentation Recovery Lexicon Approach (WSR-LA), which uses an Indonesian lexicon to reconstruct over-segmented words prior to character-level correction. The reconstructed text is subsequently corrected using similarity-based matching to identify the most appropriate lexicon candidates. An ablation study was conducted on 45 video frames to compare Levenshtein Distance (LD), Jaro (Jr), Jaro-Winkler (Jrw), Jaccard (Jcd), and Damerau-Levenshtein (DL) similarity methods at different correction thresholds. The LD method with a 70% threshold achieved the best performance, reducing the average Character Error Rate (CER) to 0.95% and Word Error Rate (WER) to 3.89%. The proposed WSR-LA framework was then evaluated on 25 additional video frames and achieved an average CER of 0.63% and WER of 1.96%, representing the lowest error rate among the four evaluated rule-based and lexicon-based baseline methods. These results demonstrate that the proposed framework effectively addresses minor OCR character errors, including insertions, deletions, and substitutions, as well as word over-segmentation. The framework can therefore support more reliable extraction of textual information from video frames for downstream video analysis and retrieval applications.

Keywords: OCR, post-OCR correction, lexicon, word segmentation, video.

1. Introduction

Optical Character Recognition (OCR) technology has become a necessity for converting text images into digital data. However, when dealing with documents with high visual complexity, OCR output often contains character recognition errors [1], [2] or over-segmentation [3]. OCR reading failures are generally triggered by various factors, such as noise, lighting variations, blurring effects [4], overlapping objects [5], or font irregularities. Furthermore, some non-text elements combined with text, such as images, may cause inconsistent spacing [6], [7]. These errors result in invalid OCR output and may affect the quality of information retrieval, particularly for dynamic video images. Post-processing with text correction is a crucial step after the OCR process to normalize and improve text quality.

Text data are post-processed to correct any errors that might have occurred during the OCR process. The post-processing techniques used in this research consist of normalization and tokenization. The results of normalization and tokenization are then improved by comparing the OCR text data with the Indonesian lexicon using five similarity measurement algorithms, namely Levenshtein Distance, Damerau-Levenshtein, Jaccard, Jaro, and Jaro-Winkler. This is known as string similarity matching for comparing data. OCR text that has undergone post-processing generally contains errors caused by character substitution, deletion, and insertion [8]. These errors result in invalid tokens. There are two types of errors: errors that cause the token to be absent from the dictionary (non-word errors) and errors that result in a valid token present in the dictionary (real-word errors) but whose usage is contextually inappropriate when applied in a sentence [9]. In the OCR process, these errors are often referred to as character recognition errors. A video can be viewed as a sequence of frames in chronological order, and individual frames can be extracted for analysis [10]. Each frame in a video has different visual characteristics. For example, a single frame can have text occluded by a

***Corresponding author: Nur Rokhman (nurrokhman@ugm.ac.id)**
DOI: https://doi.org/10.47738/jads.v7i4.1548
This is an open access article under the CC-BY license (https://creativecommons.org/licenses/by/4.0/).
© Authors retain all copyrights

cursor, text with lighting issues, or blurring caused by unstable camera movement. These conditions may affect the text segmentation process, leading to recognition errors in OCR. Various methods have been applied to extract text from images, including the use of regions of interest (ROI) and specialized segmentation methods, such as EAST [6], [11], [12] and CRAFT [13], [14]. However, the segmentation process in the OCR pipeline is still prone to over-segmentation, which causes the OCR output to contain tokens fragmented into multiple invalid parts, thus requiring a post-processing step to correct these errors.

Researchers have studied OCR post-processing, mainly concentrating on text correction, either by correcting words in isolation [8], [15], [16] or by using context [9]. An isolated word-based correction approach using Tesseract OCR and Levenshtein Distance has been implemented to enhance Indonesian road sign text recognition [15]. A Region of Interest (ROI)-based segmentation was applied before the OCR process. String similarity was measured using the Levenshtein distance on a pre-inputted database of street names. However, the study still has limitations in terms of using ROI-based segmentation, as the segmentation process must be accurate. The accuracy of the ROI segmentation results influences the validity of the character recognition results. Similarly, during the post-processing stage, characters with fewer than 4 letters are eliminated, which affects the accuracy of the text correction results, as indicated by an increase in the Character Error Rate (CER) when applied to general text.

Another approach based on context correction can also be used to improve the quality of OCR-generated text using n-gram features and Google Spelling Suggestions [9]. The advantage of this system is that it can reduce correction errors by matching words based on similarities. Better word candidates are provided by the system by exploiting word relationships in the lexicon. However, the quality of the corpus and the amount of data greatly affect the correction results, especially when the amount of data used is limited.

The development of the OCR method with lexicon-based pre- and post-processing, using Tesseract OCR and the Levenshtein distance algorithm, can evidently increase the recognition of low-quality English-language documents [8]. The results of the study showed that the quality of OCR text improved after correction based on string similarity. These studies generally focus only on resolving character-level invalidity issues, such as deletions, substitutions, and insertions, without addressing the specific challenges posed by token fragmentation errors, which frequently occur in documents with dynamic text.

This paper proposes the Word Segmentation Recovery Lexicon Approach (WSR-LA) to address the problem of fragmented tokens. The WSR-LA method is implemented by sequentially scanning tokens and generating hypotheses for merging multiple tokens, which are then deterministically validated using a lexicon and the Levenshtein distance algorithm until a complete and valid token is found. The main contribution of this research is the development of a complete text correction framework. This framework consists of the recovery of word segmentation as a pre-correction step and lexicon-based spelling validation afterwards.

2. Related Work

2.1. OCR Error and Postprocessing

The main approaches to improve the quality of OCR are modifying the input image, improving the OCR system, and processing the OCR output [17]. The post-OCR text correction process includes two sequential tasks: text detection and text correction. Generally, OCR errors can be classified into three categories, namely character insertion, deletion, and substitution errors, which cause the recognized OCR text to deviate from the original (ground truth) and become invalid [18]. In addition to character-level errors, other types of errors can also occur. OCR can cause word boundary issues, called over- and under-segmentation [19], and hallucinations [20].

These various character errors may cause invalid tokens. Some researchers classify OCR token errors into two main types: real-word errors and non-word errors [21]. A non-word error is an error that produces a token that does not exist in the lexicon or dictionary and is therefore an invalid word. A real-word error is an error that produces a real word that is in the lexicon but is inappropriate in the context of the sentence, producing an incorrect meaning. For example, the tokens “PARK IR” and “EVAKUASb”, which originate from Indonesian text, are considered non-word errors because these OCR outputs are not found in the Indonesian lexicon (dictionary).

However, the Indonesian word “KUMPULAN” (“collection” in English) is mistaken for the token “KUMPUL” (“a gathering of people” in English) which is called a real-word error because both words are valid and defined in the lexicon but have different meanings in the context of the sentence. OCR errors are dominated by real-word errors (59.21%) rather than non-word errors, and over-segmentation has a higher error rate compared to under-segmentation [21]. Of the average 15% of word boundary errors generated by OCR, 2% are forms of over-segmentation, also known as incorrect word segmentation errors [22].

2.2. Lexicon-based OCR Correction

Error correction systems for OCR results can be classified into two major approaches: isolated word-based correction and context-based correction. The isolated word-based correction compares OCR tokens with valid dictionary data using string similarity methods. The context-based correction approach compares OCR tokens with valid dictionary data while considering sentence context based on bigrams or trigrams of words [9].

The isolated word-based correction approach has been used to improve the text recognition results of road signs in the Indonesian language using Tesseract OCR and Levenshtein Distance [15]. In that study, ROI-based segmentation was performed before the OCR process. The combination of methods was successful in improving OCR results by using a pre-input database of street names based on string similarity using Levenshtein distance. During the post-processing phase, the irrelevant characters are first removed, and subsequently, the OCR results are compared with the lexicon of street names stored in a database. The purpose of this matching process is to identify the best candidate according to the minimum edit distance calculation.

This isolated-word-based correction system has also been applied to improve text recognition results on low-quality historical documents [8]. The combination of preprocessing techniques and the Levenshtein Distance method was shown to outperform other methods (Hamming Distance, Jaro-Winkler Distance, and Sørensen-Dice coefficient) based on a Word Error Rate (WER) of 8.645%, as evaluated on 40 sample documents. Furthermore, lexicon-based correction using dictionary lookup and Levenshtein Distance has been shown to improve OCR results for National Identification Card. The proposed combination of pre-processing and post-processing methods reduced the Character Error Rate (CER) to 13.94% [16].

On the other hand, a context-based correction approach was also employed to develop OCR correction techniques based on dictionary lookup and n-grams using Google Spelling Suggestion. Utilizing context-based correction has been shown to improve the quality of OCR-generated text more effectively than word-similarity-based correction methods that do not consider sentence context [9].

2.3. Word Segmentation Recovery for OCR Text

Although many post-processing correction methods have been proposed, most research still focuses on character error correction or single-word error correction. The problem of over-segmentation in OCR recognition results, e.g., the splitting of tokens into multiple invalid word fragments, has not been extensively addressed, even with classical or statistical approaches [23]. Deep learning methods for addressing OCR error cases that result in fragmented words are beginning to be investigated. For example, the Transformer-based sequence-to-sequence architecture (BART) has been used to correct noisy OCR output from old newspapers by formulating OCR correction as a translation task and showed a 29% improvement in character accuracy [20]. Although the pretrained language models are highly robust, this approach suffers from high computational cost and dependence on large-scale training datasets. Hence, this study proposes a lighter and deterministic solution through the Word Segmentation Recovery Lexicon Approach (WSR-LA) to reconstruct fragmented tokens, which are then corrected through a string similarity approach, such as Levenshtein Distance, against a lexicon to generate valid tokens.

 ISSN 2723-6471

3. Methodology

3.1. Data Preprocessing

3.1.1. Video Data

Seventy videos were independently collected at the Herman Yohanes Building, School of Applied Sciences, Universitas Gadjah Mada, Yogyakarta, during the period from March to April 2026 and used for testing purposes. Videos were recorded with a SONY ILCE7M4 camera with XAVC S HD (MPEG-4 AVC/H.264) specifications (a resolution of 1920 x 1080 pixels, a frame rate of 50 fps, and an average video duration of 13 seconds). Lighting, movement effects, and blurring, among other factors, affected the characteristics of the video data. The videos were chosen from unconstrained footage taken around the campus, containing indoor and outdoor text in different styles (room/lab name signs, announcements, text on the walls, etc.). The textual information for each frame of the testing videos was extracted from a different video and based on specific codes. The selected frames were subsequently used as inputs for detection and segmentation, followed by the process of text recognition and correction.

3.1.2. Lexicon Information

The lexicon was constructed from several corpus sources in Indonesian and English, including the digital textbooks *Analisis dan Desain Perangkat Lunak*/ Software Analysis and Design (2022, 392 pages) [24] and *Analisis dan Perancangan Sistem Berorientasi Objek dengan UML*/ Object-Oriented System Analysis and Des ign with UML (2021, 288 pages) [25]. Another corpus source was obtained from the Indonesian thesaurus, entitled Indonesian Language Center Thesaurus (2008, 560 pages) [26]. The source of the English corpus was obtained from Roget's 21st Century Thesaurus in Dictionary Form, Third Edition, as a reference source for English vocabulary (2005, 962 pages) [27]. The reference vocabulary comprises 46,572 English and 30,621 Indonesian word entries, for a total of 77, 193 word entries.

3.1.3. Lexicon Construction

The lexicon was constructed through four main stages: corpus extraction and normalization, tokenization and frequency filtering, linguistic annotation and language identification, and lexicon generation. HTML files were parsed using *BeautifulSoup* to extract textual content, followed by lowercasing, removal of non-alphabetic characters, and whitespace normalization. The resulting corpus was tokenized into words, and tokens shorter than two characters or occurring below the predefined frequency threshold were excluded. The remaining tokens were matched against the thesaurus to obtain simplified POS tags and identify Indonesian words. Tokens not found in the thesaurus were checked against the English dictionary provided by the NLTK corpus, while unmatched tokens were retained as Indonesian. Each retained token was stored with its language, POS tag, and frequency information, and the resulting entries were saved in *lexicon.json* for subsequent candidate validation and OCR correction.

3.2. Pre-processing

This stage prepares image data for character recognition. The process begins by extracting frames from a video source into static images. The images are then processed using text detection and segmentation, visual enhancement operations such as brightness and contrast adjustment, and processing filters such as morphological filters and blurring.

3.3. OCR

The characters in the segmented text area are extracted by the Tesseract OCR engine to convert image patterns into digital text strings. Several studies have shown that OCR capabilities could effectively extract text from documents. However, OCR is still prone to errors due to low image quality, complex layouts, typographic variations, and imaging defects [21]. The effect of these different conditions can cause invalid text recognition due to the addition, deletion and modification of characters as well as over-segmentation [20]. These errors are corrected through post-processing, as shown in figure 1.

ISSN 2723-6471

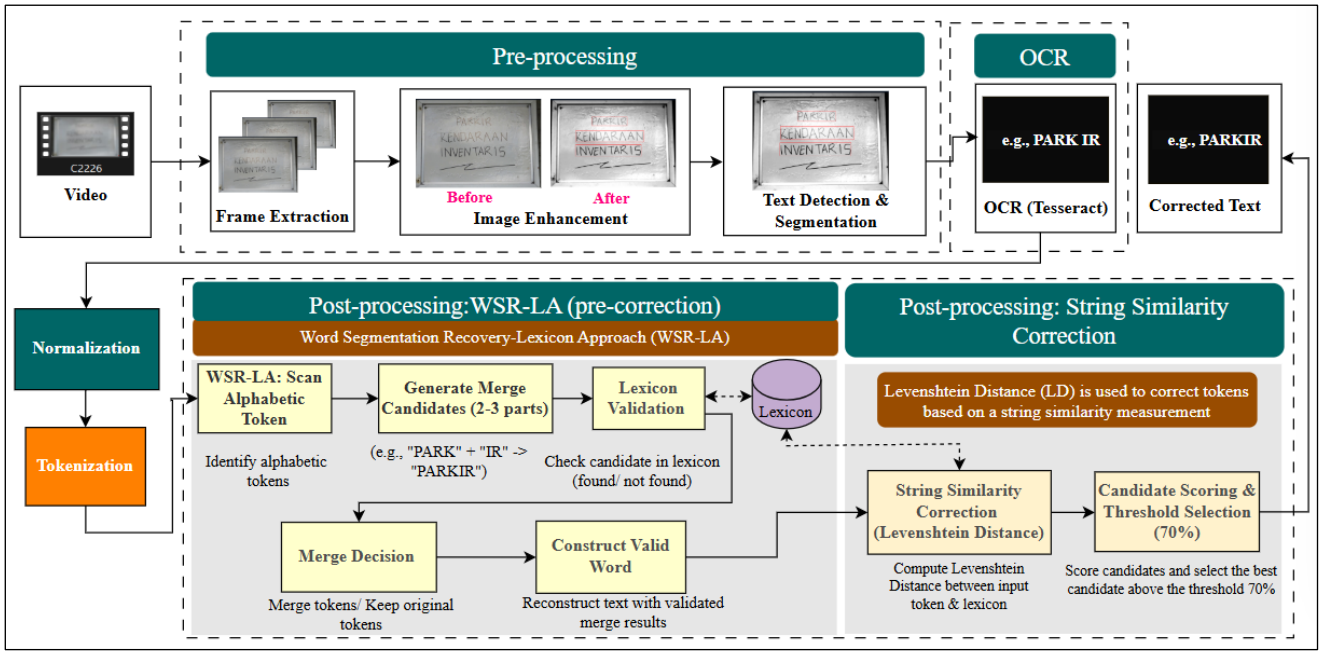


Figure 1. Character recognition and text correction steps using word segmentation recovery and a lexicon

3.4. Post-processing

This study proposes a two-stage rule-based post-processing system consisting of error detection and error correction to improve the quality of OCR output.

3.4.1. Error Detection

Error detection is conducted through lexicon-based validation, where each word is compared with entries in the lexicon to identify potential recognition errors. If the token is found in the lexicon, it is considered valid. Otherwise, it proceeds to the correction process.

3.4.2. Error Correction

The error correction process in this study consists of two approaches. First with Pre-correction using WSR-LA. The Word Segmentation Recovery-Lexicon Approach (WSR-LA) is designed to address over-segmentation errors in OCR output, where a word is split into several fragments or tokens. The WSR-LA method scans tokens sequentially to generate candidate words by merging multiple tokens and then validating these candidates against a lexicon to identify a valid token.

Second with String-based Correction. Any token not found in the lexicon is identified as a potential OCR error. The system calculates the similarity score between the token and all entries in the lexicon, using similarity calculation methods such as Levenshtein Distance (LD), Jaro (Jr), Jaro-Winkler (Jrw), Jaccard (Jcd) and Damerau-Levenshtein (DL). The correction result is the candidate with the highest similarity score that exceeds the threshold. If no candidate exceeds the threshold, the token is returned unchanged. This string similarity is calculated to correct minor character errors by assuming that the string is similar to all entries in the lexicon.

3.5. Experimental Methodology

The experiment was designed in two stages. First, an ablation study was performed to find the best setup of text correction methods and thresholds. In this setup stage, 45 video frames were tested by comparing five text similarity algorithms: Levenshtein Distance [15], Damerau-Levenshtein (DL) [28], Jaro (Jr) [28], Jaro-Winkler (Jrw) [8], and Jaccard (Jcd)[29], at different threshold levels (50%, 60%, 70%, 80%, and 90%). In the second stage, the optimal configuration obtained from the ablation study was used for the evaluation stage on an additional of 25 video frames. The performance of the proposed WSR-LA using Levenshtein Distance was compared with four OCR correction methods used as baselines: Levenshtein-distance-based correction [8], [15], [16]; and a combination of Levenshtein

 ISSN 2723-6471

Distance and n-grams [9]. The four similarity-based correction methods were re-implemented and evaluated using the same 25 video frame evaluation dataset, preprocessing procedures, OCR engine, lexicon, and evaluation metrics.

Levenshtein Distance measures the difference between an OCR word and its corresponding lexicon entry by computing their edit distance using Equation 1.

$$Lev_{a,b}(m,n) = \begin{cases} \max(m,n), & \min(m,n) = 0, \\ \min \begin{cases} Lev_{a,b}(m-1,n)+1, \\ Lev_{a,b}(m,n-1)+1, \\ Lev_{a,b}(m-1,n-1)+1_{(a_m \neq b_n)} \end{cases} & \text{otherwise.} \end{cases} \quad (1)$$

Meanwhile, the level of similarity between tokens is calculated using the Levenshtein Similarity Ratio, which is a normalization of the Levenshtein Distance value. The similarity ratio is calculated using Equation 2.

$$Levenshtein\ Similarity\ Ratio = \frac{(|a|+|b|)-Lev_{a,b}(m,n)}{|a|+|b|}. \quad (2)$$

In this equation, a is an OCR token and b is a defined reference token. |a| denotes the length of token a (number of characters), and |b| denotes the length of token b. $Lev_{a,b}(m, n)$ denotes the Levenshtein Distance value between token a and token b, which is calculated up to the m-th character in token a and the n-th character in token b. A candidate word is accepted if its Levenshtein Similarity Ratio exceeds the predefined threshold of 70%. Furthermore, performance is evaluated using the Character Error Rate (CER), defined in Equation 3, and the Word Error Rate (WER), defined in Equation 4, where lower values denote better correction performance [30].

$$CER = \frac{S_c + I_c + D_c}{N_c} \text{ x } 100\ \%\ , \quad (3)$$

$$WER = \frac{S_w + I_w + D_w}{N_w} \text{ x } 100\ \%. \quad (4)$$

N_c and N_w denote the total numbers of characters and words in the ground-truth. Likewise, S_c, I_c, and D_c denote the numbers of substituted, inserted, and deleted characters. Whereas, S_w, I_w, and D_w denote the numbers of substituted, inserted, and deleted words.

In this study, space characters are included in the character sequence when calculating CER. Character case is preserved, meaning that uppercase and lowercase letters are treated as distinct characters. Special symbols are also preserved and counted as individual characters when they appear in the text being evaluated. These rules are applied consistently throughout the experiments.

3.6. WSR-LA

Word Segmentation Recovery-Lexicon Approach (WSR-LA) is a lexicon validation-based pre-correction method used to recover fragmented OCR tokens resulting from failed OCR segmentation or recognition. The goal of WSR-LA is to recover fragmented words due to segmentation or OCR recognition errors before the correction process. Generally, segmentation errors occur not only at the character level but also at the word level. Word-level segmentation errors are classified into two types: over-segmentation and under-segmentation [31]. The processing stages of WSR-LA are summarized in table 1, with the OCR output "*PARK IR*" serving as an illustrative example. The corresponding ground-truth word is "*PARKIR*," an Indonesian word meaning "parking" in English.

Table 1. Processing Stages of WSR-LA for the OCR Output "*PARK IR*"

No	Stage	State of the token sequence
1	Tokenization	{t_1="*PARK*" t_2="*IR*"}
2	Fragment Detection	("*PARK*", "*IR*") flagged as candidate
3	Candidate Reconstruction	"*PARKIR*" (hypothesis, unvalidated)
4	Lexicon Validation	"*PARKIR*" ∈ L (valid)
5	Merge Decision	{"*PARK*", "*IR*"} →{"*PARKIR*"}
6	Output to LD Correction	"*PARKIR*"

3.7. Mechanism of WSR-LA

WSR-LA operates as a five-stage pipeline that identifies fragmented OCR tokens and reconstructs them into complete lexical units through sequential, lexicon-guided validation. This approach presents the reconstruction mechanism as a continuous workflow spanning from tokenization to the merge decision.

3.7.1. Tokenization

The pipeline begins by decomposing a raw OCR string into a sequence of basic units, including alphabetic tokens, space characters, and punctuation, while preserving their original order, as formulated in Equation 5:

$$T = \{t_1, t_2, .. t_n\} \tag{5}$$

T denotes the sequence of OCR tokens, t_i represents the *i*-th token in the sequence, and *n* denotes the total number of tokens. Space characters are intentionally retained and not removed at this stage, because the position of the word boundary itself is the information that WSR-LA needs to evaluate. For the OCR output "*PARK IR*", tokenization yields T= $\{t_1,t_2\}$, where t_1 = "*PARK*" and t_2 = "*IR*" (table 4, row 5).

3.7.2. Fragment Detection

Fragment detection is the process of identifying sequences of adjacent alphabetic tokens that are likely to originate from a single word fragmented by OCR segmentation errors. The objective of this stage is to locate potential word fragments before candidate reconstruction and lexicon-based validation are performed.

The system scans T from left to right, looking for adjacent alphabetic tokens that are plausible fragments of a single word. Non-alphabetic tokens (punctuation, isolated symbols) are skipped and never treated as merge candidates, though they remain in the sequence to preserve the sentence structure.

For "*PARK IR*", the scan identifies t_1 and t_2 as adjacent alphabetic tokens and flags the pair (t_1, t_2) as a fragment candidate. A hypothesis that these two tokens may in fact be pieces of one word broken apart by OCR segmentation error, rather than two independent words.

3.7.3. Candidate Reconstruction

Each fragment candidate is concatenated into a single hypothesis string, following Equation 6:

$$C_{i,j} = t_i \oplus t_{i+1} \oplus \ldots \oplus t_j. \tag{6}$$

$C_{i,j}$ denotes the reconstruction candidates formed from tokens t_i through t_j and the symbol $\oplus$ is used as a merge operation for the two tokens. In the current implementation, candidates are limited to chains of two or three consecutive tokens, a bound chosen to keep the search space small while still covering the great majority of over-segmentation patterns in the video-frame dataset.

Based on the 45 video frames used in the ablation study, three tokens are fragmented: two are split into two words (e.g., "*PARK IR*" (ground truth: *PARKIR*) and "*INVEN TAKS*" (ground truth: *INVENTARIS*)), and one is split into three words (e.g., "*INVEN TAR IS*"). Of these two cases, WSR-LA successfully reconstructed one correctly ("*PARK IR*" becoming "*PARKIR*"), yielding a merge recall of 1/2 (50%) and a merge precision of 1/2 (50%) across the two merge attempts. The second case was not resolved correctly: WSR-LA merged the fragments into "*INVEN TAKSA*", a valid lexicon entry, but not the intended word, "*INVENTARIS*". Notably, across these 45 test data frames, we found no instances where WSR-LA erroneously merged two adjacent tokens that were already valid, standalone words (i.e., no false-positive over-merging occurred on text that was not actually fragmented). Instead, the observed error was a partial-correction failure in a genuine fragmentation case; this was caused by the OCR fragment itself ("*TAKS*") having low string similarity to the correct word. As a result, the longest-candidate selection process chooses an incorrect lexicon entry ("*TAKSA*") instead. To illustrate the reconstruction process, consider the two-token fragmentation case "*PARK IR*". For the token pair (t_1, t_2)= ("*PARK*", "*IR*"), concatenation produces the candidate $C_{1,2}$ = "*PARKIR*". At this point, the candidate remains a hypothesis. The system has not yet decided whether it is correct.

3.7.4. Lexicon Validation

The hypothesis is then checked against the reference lexicon *L* (Section 3.1.2), following Equation 7:

$$V(C_{i,j}) = \begin{cases} 1, & if\ C_{i,j} \in L, \\ 0, & otherwise,\ if\ C_{i,j} \notin L. \end{cases} \tag{7}$$

 ISSN 2723-6471

For example, "PARKIR" is a valid entry in the Indonesian lexicon, so $V(C_{1,2})$=1, and the candidate is accepted as a valid lexical unit.

3.7.5. Merge Decision

If validation succeeds, the fragments producing the candidate are replaced by a single merged token, and the token sequence is updated accordingly. If validation fails, the original tokens are left untouched and the scan continues. When more than one candidate length (two tokens vs. three tokens) is available at a given position, the longest valid candidate is preferred, since a longer accepted merge resolves more fragments in a single decision and reduces the chance of a partial, incorrect merge downstream.

For "*PARK IR*", the merge decision accepts $C_{1,2}$ = "*PARKIR*", and the fragmented pair is replaced: the token sequence "*PARK*", "*IR*" becomes a single token "*PARKIR*". The scan then resumes from the current position, since a successful merge can occasionally expose a further merge opportunity with the next adjacent token (e.g., three-way fragmentation resolved in two passes).

3.7.6. Integration with Levenshtein-based Correction

The WSR-LA output serves as an input to the Levenshtein-based correction stage. The reconstructed sequence with "*PARKIR*" restored as a single token is passed forward as input to the string-similarity correction stage described in Section 3.4.2.2. This ordering is deliberate; Levenshtein-based correction operates most reliably on whole-word candidates, and a fragment such as "*IR*" in isolation is both too short and too ambiguous for meaningful similarity scoring against the lexicon. By resolving segmentation errors first, WSR-LA ensures that the correction module receives valid word-level candidates rather than isolated, uninterpretable fragments, which is the central motivation for treating segmentation recovery as a distinct pre-correction stage rather than folding it into the similarity-based correction step.

3.7.7. WSR-LA Algorithm

The WSR-LA procedure is summarized in algorithm 1. The algorithm takes the OCR token sequence and the lexicon as inputs and reconstructs potentially fragmented tokens by generating candidate combinations, validating them against the lexicon, and selecting the longest valid candidate.

Algorithm 1. Pseudocode of the WSR-LA procedure.

```
Input:
  – OCR token sequence T
  – Lexicon L
  – Similarity threshold τ
Output:
  – Reconstructed token sequence R
R ←T
i ←1
while i ≤ length (R) do
    if R[i] is not alphabetic token then
     i←i+1
     continue
end if
Generate two-token and three-token candidate from R[i]
valid_candidates ←∅
for each candidate C do
    w ← concatenate tokens in C
   similarity_score ←similarity (w, L)
   if similarity_score=100% then
        mark w as valid
        add w to valid_candidates
    else if w ∈ L then
        mark w as valid
        add w to valid_candidates
    else if similarity_score ≥ τ then
        mark the best lexicon match for w as valid
```

```
            add the best lexicon match to valid_candidates
        end if
    end for
    if valid candidates is not empty then
        select the longest valid candidate
        merge the corresponding tokens
        replace the original tokens with the selected candidate
    else
        retain the original tokens
        mark them as unresolved or potential OOV tokens
    end if
    move to the next unprocessed token
    end while
    return R
```

4. Results and Discussion

The results and discussion section is grouped into three stages. First, an ablation study aimed at identifying the correction methods and thresholds achieving the lowest average error in the OCR post-processing stage is presented. Second, the text correction results obtained using the optimal configuration are presented to illustrate the effectiveness of the proposed framework. Finally, a comparative study is presented to evaluate the proposed correction method against the other OCR correction methods.

4.1. Ablation Study

To evaluate and analyze the contribution of different components used in the proposed method, an ablation study was conducted. The ablation results are presented in table 2.

Table 2. OCR correction performance using CER and WER

Method	CER		WER	
	Mean	***Std***	***Mean***	***Std***
Levenshtein Distance (LD)	**1.90%**	4.13%	**8.20%**	16.55%
Jaro (Jr)	2.95%	5.77%	8.63%	16.47%
Jaccard (Jcd)	2.55%	4.12%	13.93%	21.29%
Jaro-Winkler (Jrw)	3.02%	5.76%	8.37%	15.96%
Damerau-Levenshtein (DL)	1.99%	4.28%	8.41%	17.77%

*The best performance is shown in bold. CER = Character Error Rate; WER = Word Error Rate.

As shown in table 2, among the text similarity (string similarity) measurement methods on 45 video frames, the best average Character Error Rate (CER) of 1.90% was achieved using Levenshtein Distance method (LD). On the other hand, the secondbest result was obtained by the Damerau-Levenshtein (DL) method with a value of 1.99%. The Jaro (Jr) and Jaro-Winkler (Jrw) methods, however, had the highest average CER values at 2.95% and 3.02%, respectively, indicating that text correction performance was less effective with these two methods.

The standard deviation of the Jaccard (Jcd) method is 4.12% in terms of CER consistency. It is relatively stable compared with other measurement methods. The Jaro (Jr) and Jaro-Winkler (Jrw) methods have standard deviations of 5.77% and 5.76%, respectively, which means that they have high error variability. Table 2 presents the Word Error Rate (WER) result for the similarity measurement methods. The average value of the test results is 8.20% using the Levenshtein Distance (LD) method. In addition, the Jaro-Winkler (Jrw) method had the lowest standard deviation score of 15.96%. Overall, the Levenshtein Distance (LD) method is the best for recognition error correction and provides the lowest average CER and WER values. Among the other methods, Jaccard (Jcd) is the most stable in correcting character-level errors, and Jaro-Winkler (Jrw) is the most stable at the word level. The results of the comparison of similarity measurement thresholds are presented in table 3.

 ISSN 2723-6471

Table 3. CER and WER-based evaluation of threshold selection using Levenshtein Distance

Threshold	CER		WER	
	Mean	***Std***	***Mean***	***Std***
50%	2.70%	6.23%	7.80%	18.55%
60%	1.80%	4.69%	5.50%	14.51%
70%	**0.95%**	2.93%	**3.89%**	11.87%
80%	1.14%	2.75%	5.43%	12.97%
90%	2.93%	4.03%	18.38%	24.84%

* The threshold value of 70% achieved the lowest average CER and WER among the evaluated thresholds on the development set and was therefore selected for the subsequent evaluation.

From table 3, the best performance is achieved by the Levenshtein Distance (LD) method with a threshold value of 70%, which is indicated by the lowest average error correction rates of 0.95% at the character level and 3.89% at the word level. This indicates that the Levenshtein Distance method with a 70% threshold can achieve the lowest error correction rates among the methods. Conversely, the 80% threshold has the lowest standard deviation of 2.75%, but a higher average error correction rate of 1.14%; therefore, its performance is still lower than that of the 70% threshold. The average error rates of 2.70% and 2.93% show that the 50% and 90% thresholds are less effective for OCR character correction than the 70% threshold. Thus, the best threshold for the Levenshtein Distance method is, in general, 70%.

4.2. Text Correction Result

The correction results presented in table 4 show that the proposed correction method, which validates OCR tokens against the lexicon, improves OCR output, especially for minor spelling errors. The lexicon-based correction method successfully matched invalid OCR tokens with valid entries in the lexicon. Consequently, words such as "*KOMPUTER-*", "*mandir*", "*terhadan*", and "*EVAKUASb*" were corrected to "*KOMPUTER*", "*mandiri*", *"terhadap*", and "*EVAKUASI*", respectively. The correction was successful because each invalid OCR token was matched with a valid lexicon entry, as indicated by a high Levenshtein Similarity Ratio.

Table 4. Example of OCR correction using WSR-LA

No	OCR Output	WSR-LA Result	GroundTruth	English Translation	Baseline OCR		WSR-LA	
					CER	***WER***	***CER***	***WER***
1	*JALUR* ***EVAKUASIb***	*JALUR* ***EVAKUASI***	*JALUR EVAKUASI*	EVACUATION ROUTE	7.14 **%**	50.00%	**0.00%**	**0.00%**
2	*LABORATORIUM* ***KOMPUTER-***	*LABORATORIUM* ***KOMPUTER***	*LABORATORIUM KOMPUTER*	COMPUTER LABORATORY	4.76%	50.00%	**0.00%**	**0.00%**
3	*masyarakat yang* ***mandir***	*masyarakat yang* ***mandiri***	*masyarakat yang mandiri*	an independent society	4.35%	33.33%	**0.00%**	**0.00%**
4	*persentase klasifikasi yang bernilai valid* ***terhadan*** *total klasifikasi yang dilakukan*	*persentase klasifikasi yang bernilai valid* ***terhadap*** *total klasifikasi yang dilakukan*	*persentase klasifikasi yang bernilai valid terhadap total klasifikasi yang dilakukan*	the percentage of valid classifications relative to the total number of classifications performed	1.19%	10.00%	**0.00%**	**0.00%**
5	***PARK IR*** *KENDARAAN* ***INVEN TAKS***	***PARKIR*** *KENDARAAN* ***INVENSI TAKSA***	*PARKIR KENDARAAN INVENTARIS*	COMPANY VEHICLE PARKING	14.81%	133.33%	22.22%	66.67%

*Incorrect OCR tokens are shown in bold. The best correction results are also highlighted in bold.

This improved accuracy was also reflected in the reduced CER and WER values. The text "*LABORATORIUM KOMPUTER-*" was corrected to "*LABORATORIUM KOMPUTER*", with a baseline OCR error rate of 4.76% at the character level and 50.00% at the word level. After correction, the values were reduced to 0.00%, both for the character and word levels. Similarly, the text "*JALUR EVAKUASb*" had a character-level error of 7.14% and a word-level error of 50.00%. After correction to "*JALUR EVAKUASI*", both error rates were reduced to 0.00%. This success was attributable to the candidate words used for the correction process, which were available in the lexicon.

 ISSN 2723-6471

Based on Equation 3, in the case of "*JALUR EVAKUASb*" the OCR output, there is only one character substitution (S_c = = 1). The character "I" in the ground-truth text is recognized as "b" by OCR. Meanwhile, there are no insertion or deletion operations (I_c = 0 and D_c = 0). With the number of characters in the ground-truth text as many as 14 characters (N_c = 14, including spaces), the result of CER is

$$CER = \frac{S_c + I_c + D_c}{N_c} \text{ x } 100\ \% = \frac{1 + 0 + 0}{14} \text{ x } 100\% = 7.14\%.$$

At the word level, the OCR output consists of two words: "*JALUR*" and "*EVAKUASb*." Although the second word differs from the ground-truth word "*EVAKUASI*" by only one character, it is still treated as an incorrect word. Therefore, S_w = 1, I_w= 0, D_w = 0, and N_w = 2. Based on Equation 4, the WER value is calculated as follows:

$$WER = \frac{S_w + I_w + D_w}{N_w} \text{ x } 100\ \% = \frac{1 + 0 + 0}{2} \text{ x } 100\% = 50.00\%.$$

The experimental results indicate that character substitution, insertion, and deletion are the dominant OCR error types corrected by the proposed method. For example, "*I*" is substituted by "*b*" in "*EVAKUASI*", "-" is inserted in "*KOMPUTER*", and the final character "*i*" is deleted from "*mandiri*".

In the context of OCR text recognition, errors occur not only at the character level but also at the word level [19]. A word can be split (fragmented) into several parts, as shown in the OCR output "*PARK IR*" (table 4). If fragmentation errors are not addressed before the main correction process, dictionary-based correction modules often produce inaccurate results. Therefore, the WSR-LA must be understood as a pre-correction stage that determines the quality of the final OCR output. WSR-LA is a dictionary-based approach that concatenates several individual (fragmented) tokens into a valid token, which is then matched against the lexicon. Apart from its simplicity and speed, its advantages also include seamless integration with existing applications. However, the completeness of the lexicon is of utmost importance for performing this normalization process.

WSR-LA successfully reconstructed the over-segmented token "*PARK IR*" into the valid word "*PARKIR*". This correction reduced the WER from 133.33% to 66.67%. In the example "*PARKIR KENDARAAN INVENTARIS*", the WER of 133.33% was obtained because the OCR output contains two substitutions and two insertions (table 4), while the number of words in the ground-truth text contains only three words (table 5).

Based on the minimum edit distance algorithm, the word "*PARKIR*" was substituted with "*PARK*", while "*INVENTARIS*" was substituted with "*INVEN*". In addition, the tokens "*IR*" and "*TAKS*" were counted as insertions. Thus, the resulting values are S_w = 2, I_w = 2, D_w = 0, and N_w= 3. Therefore, the WER is calculated using Equation 4 as follows:

$$WER = \frac{S_w + I_w + D_w}{N_w} \text{ x } 100\ \% = \frac{2 + 2 + 0}{3} \text{ x } 100\% = 133.33\%.$$

Table 5. Word-level edit operations baseline OCR (WER = 133.33%)

No	Ground Truth	OCR Output	Edit Operation
1	*PARKIR*	*PARK*	Substitution
2	-	*IR*	Insertion
3	*KENDARAAN*	*KENDARAAN*	Correct
4	*INVENTARIS*	*INVEN*	Substitution
5	-	*TAKS*	Insertion

Table 6 summarizes the remaining word-level edit operations using WSR-LA. After WSR-LA processing, only one word substitution and one word insertion remained. The word "*INVENTARIS*" in the ground-truth text was corrected to "*INVENSI*", resulting in one word substitution (S_w = 1). Additionally, the token "*TAKSA*" was not present in the ground-truth text, resulting in one word insertion (I_w = 1). As the ground-truth text contains three words (N_w = 3), the resulting values are S_w = 1, I_w = 1, D_w = 0, and N_w = 3. The WER is calculated using Equation 4 as follows:

$$WER = \frac{S_w + I_w + D_w}{N_w} \text{ x } 100\ \% = \frac{1 + 1 + 0}{3} \text{ x } 100\% = 66.67\%.$$

 ISSN 2723-6471

Table 6. Word-level edit operations using WSR-LA (WER = 66.67%)

No	Ground Truth	WSR-LA Output	Edit Operation
1	*PARKIR*	*PARKIR*	Correct
2	*KENDARAAN*	*KENDARAAN*	Correct
3	*INVENTARIS*	*INVENSI*	Substitution
4	-	*TAKSA*	Insertion

These remaining errors indicate that WSR-LA still depends on the quality of the OCR output. For example, the OCR token "*INVEN TAKS*" (table 4) was reconstructed as "*INVENSI TAKSA*" (table 6) but was not corrected to "*INVENTARIS*". This occurred because the OCR output contained multiple character recognition and segmentation errors, reducing the similarity between the reconstructed token and the correct lexicon entry, which ultimately affected the candidate selection process.

Overall, WSR-LA is effective in recovering over-segmentation errors by reconstructing fragmented tokens into valid words. However, its performance decreases when the OCR output contains multiple character recognition errors, out-of-vocabulary words, or OCR artifacts.

4.3. Comparative Study

The proposed WSR-LA method using Levenshtein Distance, was compared with four existing lexicon-based OCR post-correction methods, as summarized in table 7. The comparison includes preprocessing techniques, OCR inputs, OCR engines, OCR post-correction approaches, and domains used in previous studies. The reviewed methods employed different correction strategies, such as Levenshtein Distance for traffic sign recognition [15], contextual correction with stemming and n-gram models for Indonesian historical newspapers [9], lookup-table matching with Levenshtein Distance for national identification cards [16], and a comparative evaluation of four string similarity algorithms, namely Levenshtein Distance, Hamming Distance, Jaro-Winkler Distance, and Sorensen-Dice coefficient, for Old English documents [8].

Table 7. Overview of existing lexicon-based OCR post-correction methods

No	Author	Preprocessing	OCR Engine	OCR post-correction	Domain
1	Priambada & Widyantoro [15]	HSV Green Mask + Otsu + Canny + Bounding Box ROI	Tesseract	Road name database, Levenshtein Distance, Rule-based filtering	Traffic Signs
2	Purwantoro et al. [9]	Brightness and Contrast, Denoise, Deskew, Binarize	Tesseract	Dictionary, n-gram, Levenshtein Distance, Google Spelling	Historical Newspapers
3	Christian & Kusuma [16]	Shadow Removal, Gaussian Blur, Custom Grayscale	Tesseract	Lookup Table, Levenshtein Distance	ID Cards
4	Enrico & Kusuma [8]	Background Removal, Sharpening, Blur, Binarization, Non-Local Means (NLM) Denoise	Tesseract	Levenshtein Distance, Dictionary, Optional (Hamming Distance, Jaro-Winkler, Sorensen–Dice coefficient)	Old Documents
5	Proposed Method	Brightness & Contrast Adjustment, Binarization, Denoising, Morphology & Blurring filter, CRAFT	Tesseract	Lexicon, WSR-LA, Levenshtein Distance, Supports Jaro, Jaro-Winkler, Jaccard, Damerau-Levenshtein	Campus Scene Text

The comparison was conducted using the evaluation metrics of CER and WER, and the results were summarized using the mean and standard deviation. The performance of the proposed WSR-LA method in comparison with the four OCR post-correction methods is presented in table 8. The proposed WSR-LA framework using Levenshtein Distance (LD) at a 70% similarity threshold, achieved the lowest average CER of 0.63% and WER of 1.96% among the four rule-based and lexicon-based baseline correction methods evaluated on the same 25 video frames.

ISSN 2723-6471

Table 8. Statistical comparison of OCR post-correction with existing methods

No	Method	Combination	CER		WER	
			Mean	***Std***	***Mean***	***Std***
1	[15]	Tesseract + Dictionary + LD + Rule-based filtering	6.33%	8.16%	11.37%	13.57%
2	[9]	Tesseract + Dictionary + LD + n-gram correction	5.14%	8.41%	7.56%	10.60%
3	[16]	Tesseract + Lookup Table + LD	3.56%	7.35%	7.65%	12.92%
4	[8]	Tesseract + LD + Dictionary	3.46%	5.64%	7.68%	12.31%
5	**Proposed**	Tesseract + Lexicon + WSR-LA	**0.63%**	1.52%	**1.96%**	4.11%

The 0.63% character-level error rate was due to the WSR-LA method's inability to correct some words with affixes. Furthermore, some errors were caused by character substitutions, which were categorized as capitalization errors, such as changing the token "*DI*" to "*Di*". Although both words are lexically valid (real-word errors), the character differences were still counted as errors in the CER evaluation. Meanwhile, the word-level error rate was 1.96% because some words were unavailable in the lexicon, causing the system to replace the correction candidate with a word different from the reference word.

Meanwhile, method [15] produced character-level errors of 6.33% and word-level errors of 11.37%. The character error rate of 6.33% was due to the correction method failing to eliminate noise artifacts in the OCR results because they were recognized as valid words in the lexicon (e.g., "AGEE" (a valid English word), "*UMAT*" (a valid Indonesian word meaning "the believers of a religion")). As a result, the lexicon-based method retained these tokens even though they were contextually meaningless. Meanwhile, the 11.37% word-level error rate was due to the OCR text containing abbreviations or short words of fewer than four characters, such as "*DI*" and "*INI*" (valid Indonesian words meaning "in" and "this" respectively) and "NO" (a short token), as shown in table 9. These tokens could be eliminated during the correction process, causing the corrected text to differ from the reference text.

Table 9. Qualitative comparison of OCR post-correction results

Image	Method	OCR Output	Corrected Text
 (EVACUATION ROUTE)	Tesseract + Levenshtein Distance [15]	*a=JALUR; EVAKUAS*	*JALUR EVAKUASI*
	Tesseract + LD + n-gram correction [9]	*La — . 3 JALUR Ja EVAKUASI*	*JALUR EVAKUASI*
	Tesseract + Lookup table + LD [16]	*JALUR EVAKUASI*	*JALUR EVAKUASI*
	Tesseract + LD + Dictionary [8]	*JALUR EVAKUASI*	*JALUR EVAKUASI*
	Tesseract + Lexicon + WSR-LA (Proposed)	*JALUR EVAKUASI*	*JALUR EVAKUASI*
 (Social Media Communication Etiquette)	Tesseract + Levenshtein Distance [15]	*ON - Sa NG Ia an mau Etiket Berkomunikasi di Media Sosial*	*ETIKET BERKOMUNIKASI MEDIA SOSIAL*
	Tesseract + LD + n-gram correction [9]	*TIM... Etiket Berkomunikasi di Media Sosial*	*TIM Etiket Berkomunikasi di Media Sosial*
	Tesseract + Lookup table + LD [16]	*Etiket Berkomunikasi di Media Sosial*	*Etiket Berkomunikasi di Media Sosial*
	Tesseract + LD + Dictionary [8]	*Etiket Berkomunikasi di Media Sosial*	*Etiket Berkomunikasi di Media Sosial*
	Tesseract + Lexicon + WSR-LA (Proposed)	*Etiket Berkomunikasi di Media Sosial*	*Etiket Berkomunikasi di Media Sosial*
	Tesseract + Levenshtein Distance [15]	*O) SNPMB Kuesioner Peserta UTBK 2026 Pra Pir ans Ngee eed" an oa: Pe Pe l Oo Pe Soee Pye AA AN a KI*	*SNPMB KUESIONER PESERTA UTBK AGEE*
	Tesseract + LD + n-gram correction [9]	*SNPMB Kuesioner Peserta UTBK 2026*	*Kuesioner Peserta UTBK*
	Tesseract + Lookup table + LD [16]	*© SNPMB Kuesioner Peserta UTBK 2026*	*SNPMB Kuesioner Peserta UTBK 2026*
	Tesseract + LD + Dictionary [8]	*¿ SNPMB Kuesioner Peserta UTBK 2026 ea*	*SNPMB Kuesioner Peserta UTBK 2026 Ea*
	Tesseract + Lexicon + WSR-LA (Proposed)	*SNPMB Kuesioner Peserta UTBK 2026*	*SNPMB Kuesioner Peserta UTBK 2026*
	Tesseract + Levenshtein Distance [15]	*N Bo.*	*DILARANG MEROKOK*

ISSN 2723-6471

	Tesseract + LD + n-gram correction [9]	” .g *DILARANG MEROKOK . &* *. DI AREA INI 1.* *— NO SMOKING AREA , —* *e e ea TI* *”TA an* *d . – —* *! . ”* *dB* *DILARANG MEROKOK . G* *DI AREA INI ”5* *B NO SMOKING AREA -* *e e - ’*	*AREA* *SMOKING AREA* *an* *DILARANG MEROKOK* *DI AREA INI* *NO SMOKING AREA*
	Tesseract + Lookup table + LD [16]	*DILARANG MEROKOK* *DI AREA INI* *NO SMOKING AREA* *e “* *7* *-* *.*	*DILARANG MEROKOK* *DI AREA INI* *NO SMOKING AREA* *e* *7*
	Tesseract + LD + Dictionary [8]	*-* *DILARANG MEROKOK* *DI AREA INI* *NO SMOKING AREA* *er)* *4*	*DILARANG MEROKOK* *DI AREA INI* *NO SMOKING AREA*
	Tesseract + Lexicon + WSR-LA (Proposed)	*r* *DILARANG MEROKOK* *Di AREA INI* *NO SMOKING AREA*	*DILARANG MEROKOK* *Di AREA INI* *NO SMOKING AREA*

Character-level errors were also observed in method [9], with a character-level error rate of 5.14% and a word-level error rate of 7.56%. The error rate of 5.14% was due to noise or artifacts in the OCR results that could not be eliminated because they were recognized as valid words in the lexicon (e.g., “an” (a valid English word) and “*TIM*” (a valid Indonesian word meaning “team”)). As a result, lexicon-based approaches tended to keep these tokens even though they did not carry meaningful information in the context. At the same time, the 7.56% word-level error was due to the method’s inability to preserve special entities, such as abbreviations, codes, numerical tokens, and acronyms. As a result, in some cases, valid tokens listed in the lexicon, such as “*SNPMB*” (an Indonesian acronym for the National Selection for University Student Admission/ “*Seleksi Nasional Penerimaan Mahasiswa Baru*”) and “2026” (a numerical token), could be eliminated during the correction process. These special entities are often different and do not follow common linguistic patterns that are found in the vocabulary and thus can be considered noise by the OCR system.

Method [16], on the other hand, produced a character-level error rate of 3.56% and a word-level error rate of 7.65%. OCR artifacts during the correction process may have caused character errors, such as the addition of characters and symbols not registered in the lexicon, which can cause the corrected text to differ. Moreover, the lookup table and string similarity methods may generate correction candidates that are not meaningfully appropriate. For example, the OCR token “nam” is corrected to “ENAM” (an Indonesian word meaning “six”) because it has high string similarity, even though it is not necessarily suitable for the sentence context. However, this method is more effective on documents with a relatively fixed structure, layout, and vocabulary, such as data extracted from National Identification Card.

On the other hand, method [8] produced a character error rate of 3.46% and a word error rate of 7.68%. The character errors were primarily caused by OCR artifacts that were not handled during the correction process, thereby increasing the CER. Some OCR artifacts coincidentally matched valid entries in the lexicon and were therefore retained without correction. Examples of such artifacts are “a”, “an”, “ae”, and “er”. The word errors were mainly caused by multiple character recognition errors, particularly character deletions. For example, the Indonesian OCR token “*menggunak*”, which should have been “*menggunakan*” (an Indonesian word meaning “to use”), was incorrectly corrected to “*menggasak*” (an Indonesian word meaning “to snatch”). The absence of multiple characters reduced the similarity between the OCR token and the intended word, causing the lexicon-based candidate selection process to choose an incorrect correction. However, this correction method is more effective for structured documents than for text extracted from videos, which tend to have more noise and layout variations.

Overall, the comparative analysis shows that current methods remain limited in handling OCR artifacts, segmentation errors, and domain-specific words in OCR-extracted text. Therefore, the proposed WSR-LA framework introduces a segmentation recovery stage before lexicon validation to address these challenges.

4.1. Discussion

The proposed method differs fundamentally from previous lexicon-based OCR post-correction approach. Existing method [8], [9], [16] perform OCR directly on the entire image or document, or rely on conventional region extraction before lexical correction [15], without handling fragmented tokens. In contrast, the proposed approach integrates CRAFT-based text detection with WSR-LA, allowing fragmented OCR tokens to be reconstructed before lexical validation.

While WSR-LA effectively addresses word fragmentation caused by OCR segmentation errors, we acknowledge that its longest-valid-candidate strategy may occasionally introduce merges or false positives. This reconstruction process remains highly dependent on the quality of the OCR tokens produced in earlier stages. For example, the fragmented OCR output "*INVEN TAKS*" was reconstructed into the lexically valid phrase "*INVENSI TAKSA*" instead of the ground-truth word "*INVENTARIS*". In this case, the original OCR fragments already contained multiple character-level errors before the merge decision was applied. As a result, the reconstructed candidate matched an alternative lexicon entry rather than the intended word. This observation indicates that WSR-LA is capable of resolving word boundary errors, but its effectiveness is constrained by the character-level accuracy of the OCR output supplied as input.

The suitability of the data domain (e.g., ID cards, traffic signs, or historical newspapers) should also be considered. Performance differences observed across comparative studies are influenced by the characteristics of the images used for the test data. The image data used in this study were extracted from videos, which have a wider variety of characteristics, such as font type, layout, color, and lighting factors. This high variation resulted in more OCR errors that required correction.

Despite the significant reductions in CER and WER, the proposed WSR-LA framework has three major limitations. First, the reconstruction process is highly dependent on the quality of the OCR tokens. Although WSR-LA can recover fragmented words, it cannot correct severe character level-errors that occur before segmentation recovery. Consequently, incorrectly recognize fragment may be reconstructed into lexical valid but unintended words. Second, its heavy reliance on the lexicon makes it prone to failure when encountering out-of-vocabulary (OOV) tokens, non-standard abbreviations, or unregistered specific terms. Third, the use of pure string distance metrics, such as Levenshtein Distance, operates exclusively at the lexical (character) level and does not consider the semantic context between words. This potentially leads to real-word errors in which spellings are corrected to dictionary-valid words that are inappropriate for the context of the sentence.

5. Conclusion

This study shows that the proposed WSR-LA framework can address token fragmentation in dynamic video OCR output prior to character-level correction. As a pre-correction stage, WSR-LA reconstructs fragmented word boundaries by generating and validating candidate word combinations against the lexicon. Based on an ablation study of various text similarity measurement algorithms, Levenshtein Distance (LD) with a threshold of 70% achieved the lowest average error rates, reducing the CER to 0.95% in the testing phase. Furthermore, in the comparative evaluation phase, the integration of WSR-LA using LD achieved a CER of 0.63% and a WER of 1.96%, representing the lowest error rates among the evaluated baseline methods.

The current WSR-LA implementation has several limitations. First, it uses a global 70% Levenshtein similarity threshold for tokens of different lengths, which may not provide equal correction sensitivity for short and long tokens. Second, the reconstruction process primarily relies on the sequential order of OCR tokens and does not directly incorporate spatial relationships between detected text regions, which may cause inappropriate merging in complex or multi-line layouts. In addition, the closed lexicon may limit the correction of OOV words, and capitalization ambiguities may remain unresolved when multiple lexical candidates are valid. For example, the token "*Di*" was not corrected to "*DI*" because both forms were accepted during lexical validation. Furthermore, non-alphabetic characters,

symbols, and OCR artifacts may require more targeted handling. Future research should therefore investigate length-adaptive similarity thresholds, spatial information such as bounding-box proximity, vertical alignment, and text-line relationships, as well as lexicon expansion and contextual language models such as n-grams to improve candidate selection. Further work on classifying and filtering non-alphabetic OCR artifacts could also help remove irrelevant noise while preserving meaningful information from the image.

6. Declarations

6.1. Author Contributions

Conceptualization: M.N.; Methodology: M.N.; Software: M.N.; Validation: M.N.; Formal Analysis: M.N.; Investigation: M.N.; Resources: N.R.; Data Curation: N.R.; Writing – Original Draft Preparation: M.N.; Writing – Review and Editing: N.R., K.M.; Visualization: M.N.; All authors have read and agreed to the submitted version of the manuscript.

6.2. Data Availability Statement

The lexicon dataset used in this study is publicly available and open for access [32]. The video dataset used in this study was obtained from the School of Applied Sciences, Universitas Gadjah Mada. The terms of this agreement explicitly restrict the public dissemination of the collected records. Consequently, the video dataset is not publicly available. Researchers interested in accessing the data may contact the author to discuss potential collaboration.

6.3. Funding

The authors would like to express their sincere gratitude to the Indonesian Education Scholarship, Center for Higher Education Funding and Assessment, and Indonesian Endowment Fund for Education, Ministry of Higher Education, Science, and Technology of the Republic of Indonesia, with grant No. 00347/BPPT/BPI.06/9/2024.

6.4. Institutional Review Board Statement

This study did not involve human participants; therefore, ethical approval was not required.

6.5. Informed Consent Statement

The video dataset used in this research was obtained with authorization from the Department of Electrical Engineering and Informatics, School of Applied Sciences, Universitas Gadjah Mada. In addition, the thesaurus resources used in this study have obtained permission from the Center for Language and Literature Development and Protection, Ministry of Primary and Secondary Education, Republic of Indonesia.

6.6. Declaration of Competing Interest

The authors declare no conflict of interest.

6.7. Acknowledgements

The authors also sincerely appreciate the Department of Electrical Engineering and Informatics, School of Applied Sciences, Universitas Gadjah Mada, Yogyakarta, for providing the video dataset used in this research. The authors also express their gratitude to the Center for Language and Literature Development and Protection of the Ministry of Primary and Secondary Education of the Republic of Indonesia, and the Central Java Provincial Language Office for their valuable support and assistance throughout this research.

References

[1] M. A. Shejwal and S. D. Bharkad, "Segmentation and extraction of text from curved text lines using image processing approach," *Proc. Int. Conf. Inf., Commun., Instrumentation Control*, vol. 2017, no. 1, pp. 1–5, Dec. 2017, doi: 10.1109/ICOMICON.2017.8279138.

[2] V. Nastase and J. Hitschler, "Correction of OCR word segmentation errors in articles from the ACL collection through neural machine translation methods," *Proc. Int. Conf. Language Resources Evaluation*, vol. 2018, no. Jul., pp. 706–711, Jul. 2018.

[3] T. A. Tofiq and J. A. Hussein, "Kurdish text segmentation using projection-based approaches," *UHD J. Sci. Technol.*, vol. 5, no. 1, pp. 56–65, 2021, doi: 10.21928/uhdjst.v5n1y2021.pp56-65.

[4] X. Zhou, C. Wang, X. Wang, and W. Liu, "Video text tracking with transformer-based local search," *Neurocomputing*, vol. 609, no. 1, pp. 1–12, 2024, doi: 10.1016/j.neucom.2024.128420.

[5] Z. Raisi and J. Zelek, "Occluded text detection and recognition in the wild," *Proc. Conf. Robots Vision*, vol. 2022, no. 1, pp. 140–147, 2022, doi: 10.1109/CRV55824.2022.00026.

[6] X. Zhou et al., "EAST: An efficient and accurate scene text detector," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2017, no. Jul., pp. 5551–5560, Jul. 2017, doi: 10.1109/CVPR.2017.283.

[7] N. Mehta and J. Doshi, "Segmentation methods: A review," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 1, pp. 536–540, 2020, doi: 10.22214/ijraset.2020.31939.

[8] N. Enrico and G. P. Kusuma, "Enhancing the extraction of low-quality old English documents using optical character recognition," *JOIV: Int. J. Informatics Visualization*, vol. 10, no. 2, pp. 670–677, 2026, doi: 10.62527/joiv.10.2.3375.

[9] D. Purwantoro, H. Akbar, A. Hidayati, and Sfenrianto, "OCR correction for Indonesian historic newspapers using word repetition, stemmer and n-gram," *J. Phys.: Conf. Ser.*, vol. 1193, no. 1, pp. 1–10, 2019, doi: 10.1088/1742-6596/1193/1/012032.

[10] B. U. Kota, A. Stone, K. Davila, S. Setlur, and V. Govindaraju, "Automated whiteboard lecture video summarization by content region detection and representation," *Proc. Int. Conf. Pattern Recognit.*, vol. 2021, no. Jan., pp. 10704–10711, Jan. 2021, doi: 10.1109/ICPR48806.2021.9412386.

[11] K. Dutta, M. Mathew, P. Krishnan, and C. V. Jawahar, "Localizing and recognizing text in lecture videos," *Proc. Int. Conf. Frontiers Handwriting Recognit.*, vol. 2018, no. Aug., pp. 235–240, Aug. 2018, doi: 10.1109/ICFHR-2018.2018.00049.

[12] X. Peng, X. Chen, and C. Liu, "Real-time traffic sign text detection based on deep learning," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 768, no. 7, pp. 1–10, 2020, doi: 10.1088/1757-899X/768/7/072039.

[13] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, vol. 2019, no. Jun., pp. 9365–9374, Jun. 2019, doi: 10.1109/CVPR.2019.00959.

[14] M. R. Hasan et al., "Smart OCR for recognizing Bangla characters with CRAFT and deep learning models," *Proc. IEEE Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf.*, vol. 2022, no. Nov., pp. 573–577, Nov. 2022, doi: 10.1109/UEMCON54665.2022.9965668.

[15] S. Priambada and D. H. Widyantoro, "Levenshtein distance as a post-process to improve the performance of OCR in written road signs," *Proc. Int. Conf. Inf. Technol. Syst. Innovation*, vol. 2017, no. Nov., pp. 1–6, Nov. 2017, doi: 10.1109/ICITSI.2017.8267912.

[16] I. Christian and G. P. Kusuma, "Improving OCR performance on low-quality image using pre-processing and post-processing methods," *Int. J. Eng. Trends Technol.*, vol. 71, no. 6, pp. 396–405, 2023, doi: 10.14445/22315381/IJETT-V71I6P239.

[17] M. Volk, L. Furrer, and R. Sennrich, "Strategies for reducing and correcting OCR errors," in *Language Technology for Cultural Heritage: Selected Papers from the LaTeCH Workshop Series*, Berlin, Heidelberg: Springer, vol. 2011, no. 1, pp. 3–22, 2011, doi: 10.1007/978-3-642-20227-8_1.

[18] S. Karthikeyan, A. G. S. de Herrera, F. Doctor, and A. Mirza, "An OCR post-correction approach using deep learning for processing medical reports," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 2574–2581, May 2022, doi: 10.1109/TCSVT.2021.3074958.

[19] G. Shim, "REVISE: A framework for revising OCRed text in practical information systems with data contamination strategy," in *Proc. 63rd Annu. Meeting Assoc. Comput. Linguistics (ACL), Industry Track*, Vienna, Austria, vol. 2025, no. Jul., pp. 1423–1434, Jul. 2025, doi: 10.18653/v1/2025.acl-industry.100.

[20] E. Soper, S. Fujimoto, and Y.-Y. Yu, "BART for post-correction of OCR newspaper text," in *Proc. 7th Workshop Noisy User-Generated Text (W-NUT)*, Online, vol. 2021, no. Nov., pp. 284–290, Nov. 2021, doi: 10.18653/v1/2021.wnut-1.31.

[21] T.-T.-H. Nguyen, A. Jatowt, M. Coustaty, N.-V. Nguyen, and A. Doucet, "Deep statistical analysis of OCR errors for effective post-OCR processing," in *Proc. ACM/IEEE Joint Conf. Digital Libraries (JCDL)*, Urbana-Champaign, IL, USA, vol. 2019, no. 1, pp. 29–38, 2019, doi: 10.1109/JCDL.2019.00015.

[22] K. Kukich, "Spelling correction for the telecommunications network for the deaf," *Commun. ACM*, vol. 35, no. 5, pp. 80–90, May 1992, doi: 10.1145/129875.129882.

[23] Q.-D. Nguyen, D.-A. Le, N.-M. Phan, and I. Zelinka, "OCR error correction using correction patterns and self-organizing migrating algorithm," *Pattern Anal. Appl.*, vol. 24, no. 2, pp. 701–721, May 2021, doi: 10.1007/s10044-020-00936-y.

[24] A. S. Rosa, *Analisis dan Desain Perangkat Lunak: Rekayasa Perangkat Lunak untuk Pemrograman Terstruktur, Berorientasi Objek, dan Agile*. Bandung, Indonesia: Informatika Bandung, vol. 2022, no. 1, pp. 1–500, 2022.

[25] Munawar, *Analisis Perancangan Sistem Berorientasi Objek dengan UML (Unified Modeling Language)*, 2nd ed. Bandung, Indonesia: Informatika Bandung, vol. 2021, no. 1, pp. 1–500, 2021.

[26] Pusat Bahasa, *Tesaurus Bahasa Indonesia*. Jakarta, Indonesia: Pusat Bahasa, Departemen Pendidikan Nasional, vol. 2008, no. 1, pp. 1–1000, 2008.

[27] B. A. Kipfer, *Roget's 21st Century Thesaurus in Dictionary Form*, 3rd ed. New York, NY, USA: The Philip Lief Group, Inc., vol. 2005, no. 1, pp. 1–1000, 2005.

[28] N. Ifada, F. H. Rachman, and S. Wahyuni, "Character-based string matching similarity algorithms for Madurese spelling correction: A preliminary study," in *Proc. Int. Conf. Electr. Eng. Informatics (ICEEI)*, Denpasar, Bali, Indonesia, vol. 2023, no. 1, pp. 1–6, 2023, doi: 10.1109/ICEEI59426.2023.10346716.

[29] S. H. Precilla P., S. B. E., and R. Mohandas, "Automated plagiarism detection in handwritten documents using various computer vision algorithms," in *Proc. 4th OPJU Int. Technol. Conf. Smart Comput. Innovation Advancement Industry 5.0 (OTCON)*, Raigarh, India, vol. 2025, no. 1, pp. 1–7, 2025, doi: 10.1109/OTCON65728.2025.11070812.

[30] T. T. H. Nguyen, A. Jatowt, M. Coustaty, and A. Doucet, "Survey of post-OCR processing approaches," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 124:1–124:37, Jul. 2021, doi: 10.1145/3453476.

[31] H. Afli, Z. Q. A. W. P. S., "Using SMT for OCR error correction of historical texts," in *Proc. 10th Int. Conf. Language Resources and Evaluation (LREC)*, Portorož, Slovenia, vol. 2016, no. 1, pp. 962–966, May 2016. [Online]. Available: https://aclanthology.org/L16-1153/

[32] M. Nurkamid, "Bilingual frequency-based lexicon," *Kaggle*, Jul. 2026, doi: 10.34740/kaggle/dsv/18218660.