ML 24/25-10 Creating Text from Images with OCR API

Karthik Prabu Natarajan
1564587
Karthik.Natarajan@stud.fra-uas.de

***Abstract*— Optical Character Recognition (OCR) technology faces persistent challenges in achieving high accuracy, particularly when processing documents with poor image quality, complex layouts, or non-standard fonts. This paper presents an innovative OCR solution that systematically evaluates the impact of 24 distinct preprocessing techniques on text extraction quality. We introduce a novel multi-metric evaluation framework that combines cosine similarity, Levenshtein distance, Jaro-Winkler distance, Jaccard index, and cluster analysis to comprehensively assess OCR performance. Our solution employs an ensemble approach that leverages advanced language models to generate synthetic ground truth, eliminating the need for manual transcription. Experiments conducted on a diverse dataset of 50 documents demonstrate that our approach improves OCR accuracy by 15-30% compared to baseline methods, with grayscale conversion and adaptive thresholding providing the most consistent improvements (average similarity scores of 0.913 and 0.892 respectively). Furthermore, our clustering-based analysis reveals patterns in preprocessing effectiveness that are not captured by traditional evaluation metrics. The developed application provides cross-platform visualization tools that enable users to identify optimal preprocessing strategies for specific document types, addressing a significant gap in current OCR technology. This research contributes to the field by offering quantitative insights into preprocessing effectiveness, introducing a multi-dimensional evaluation framework, and demonstrating the value of unsupervised learning techniques in OCR optimization.**

Keywords*— *optical character recognition, image preprocessing, text similarity metrics, clustering analysis, synthetic ground truth, ensemble methods, vector embeddings, OCR evaluation

I. INTRODUCTION

Optical Character Recognition (OCR) is a pivotal technology in the digital transformation era, enabling the conversion of various document types, such as scanned paper documents, PDFs, or images captured by digital cameras, into editable and searchable data. OCR systems typically involve multiple stages: image acquisition, where document images are captured, and preprocessing, which enhances image quality to improve recognition accuracy. Segmentation is then applied to distinguish text from background elements, followed by feature extraction, which identifies distinctive character properties. In the classification stage, extracted features are matched to known character patterns, and post-processing refines recognition results using linguistic context [1].

Despite significant advancements in OCR technology over the past decade, challenges persist in achieving high accuracy and efficiency, particularly in complex or noisy environments. The reliability of OCR systems is often compromised when processing documents with poor image quality, complex layouts, or non-standard fonts, leading to errors that reduce the utility of the extracted text. [2]

Recent research has primarily focused on two approaches to improving OCR accuracy: (1) enhancing recognition algorithms through deep learning techniques and (2) applying preprocessing methods to optimize input quality. While advanced neural network architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [3], have significantly improved classification accuracy, the critical role of preprocessing in real-world OCR applications remains underexplored and insufficiently quantified. Addressing these challenges requires a systematic evaluation of preprocessing techniques and their impact on OCR performance across diverse document types. [4]

A. LITERATURE REVIEW

This section reviews the relevant literature in OCR technology, preprocessing techniques, text similarity analysis, and ensemble methods for ground truth generation, providing context for our research contributions.

1) Evolution of OCR Technology

Optical Character Recognition has evolved significantly from early pattern-matching techniques to modern deep learning approaches. Smith [5]provides a comprehensive review of recent OCR advancements, highlighting the transition from traditional feature extraction methods to convolutional neural networks. These developments have substantially improved recognition accuracy, particularly for Latin script languages, but challenges persist for complex layouts, degraded documents, and non-Latin scripts.

Memon et al. [6] conducted an extensive survey of deep learning approaches for OCR, analyzing various architectures including CNNs, RNNs, and transformer models. Their work demonstrates that while end-to-end deep learning models achieve state-of-the-art results on benchmark datasets, they often require substantial preprocessing to handle real-world document variations. This finding underscores the continued importance of image preprocessing in practical OCR applications, even as recognition algorithms advance.

2) Image Preprocessing for OCR Enhancement

Image preprocessing techniques play a crucial role in OCR performance by improving input quality before character recognition. Peng et al. [7] evaluated the impact of various

preprocessing methods on OCR accuracy, demonstrating that appropriate preprocessing can improve recognition rates by 15-30% depending on document quality. Their research established that no single preprocessing technique is optimal for all document types, highlighting the need for adaptive preprocessing strategies.

Kumar et al. [8] proposed a systematic framework for selecting optimal preprocessing techniques based on document characteristics. Their work introduced objective metrics for evaluating preprocessing effectiveness and demonstrated the importance of document-specific preprocessing pipelines. Their findings showed that while binarization and deskewing provide consistent improvements across most document types, noise reduction and contrast enhancement techniques yield varying results depending on image quality and content.

3) Text Similarity Metrics and OCR Evaluation

Traditional OCR evaluation has relied primarily on character and word error rates, which fail to capture semantic similarities between OCR outputs and ground truth. Zhai et al. [9] investigated alternative evaluation metrics including cosine similarity and various edit distance measures. Their work demonstrated that combining multiple similarity metrics provides more comprehensive insights into OCR performance than single-metric approaches, particularly for documents with complex layouts or where context preservation is important.

4) Ensemble Methods for OCR Improvement

Ensemble approaches that combine multiple OCR engines or preprocessing pipelines have emerged as an effective strategy for improving recognition accuracy. Fujii et al. [10]proposed a voting-based ensemble method that significantly outperformed individual OCR engines across various document types. Their approach demonstrated particular effectiveness for challenging documents with degraded quality or unusual fonts.

Recent research has explored using language models to improve OCR results by correcting and combining outputs from multiple engines. This approach leverages linguistic knowledge to resolve ambiguities and correct errors that persist through the recognition process, showing particular promise for domain-specific documents where context and terminology are important considerations.

B. PROBLEM STATEMENT AND RESEARCH QUESTIONS

Despite the crucial role of preprocessing in optical character recognition (OCR) workflows, existing research and practical implementations exhibit three significant limitations. First, there is no comprehensive framework for systematically evaluating the impact of various preprocessing techniques on OCR performance across diverse document types. The lack of standardized assessment methods hinders the ability to determine which preprocessing techniques are most effective in different scenarios. Second, current OCR solutions provide limited tools for performance visualization, making it challenging for users to compare preprocessing methods effectively and make informed decisions. Finally, nfo established methodology exists for the automatic selection of optimal preprocessing techniques based on document characteristics, leading to suboptimal OCR results when manual selection is infeasible. [8]

To address these gaps, our research explores the following questions:

1. How can the impact of different preprocessing techniques on OCR accuracy be quantitatively [3]? [7]
2. How can visualization techniques enhance the accessibility and usability of preprocessing optimization for OCR users? [9]

We hypothesize that integrating multiple similarity metrics will provide more comprehensive insights into OCR performance than single-metric approaches, that visualizing text embeddings will reveal patterns in OCR errors not apparent through traditional evaluation methods, and that the application's comparative analysis capabilities will improve OCR accuracy by identifying optimal preprocessing methods for specific image types. Building upon these foundations, our work integrates multiple similarity metrics, conducts comprehensive preprocessing evaluations, and employs an advanced ensemble approach that combines statistical voting with language model analysis to generate synthetic ground truth. This combination of techniques addresses key gaps in OCR performance evaluation and optimization.

II. METHODOLOGY

The development and evaluation of the OCR application were conducted using a systematic approach that combined software engineering methodologies with empirical testing. This section describes the methods employed in the design, implementation, and evaluation of the application.

A. System Architecture

We designed our application with a modular architecture to ensure flexibility, extensibility, and maintainability. Figure 1 illustrates the comprehensive system architecture with five main components: (1) the input handler (2) the preprocessing module, (3) the OCR engine integration layer, (4) the similarity analysis module, and (5) the visualization module.

The OCR engine was responsible for processing input images and extracting text. The application supported multiple OCR engines, including Tesseract OCR, IronOCR and Google Cloud Vision, to provide users with flexibility in choosing the most appropriate engine for their specific needs. The OCR engine was integrated through a common interface, allowing for easy addition of new engines in the future.

1) Preprocessing Module

The preprocessing module implements 24 distinct image enhancement techniques, organized into five functional categories:

- Noise Reduction: Gaussian filtering, median filtering, bilateral filtering
- Binarization: Otsu's method, adaptive thresholding
- Geometric Transformations: Deskewing, rotation, scaling
- Morphological Operations: Dilation, erosion, opening, closing, gradient, top-hat, black-hat
- Intensity Adjustments: Gamma correction, histogram equalization, contrast stretching, brightness reduction

Each preprocessing technique is implemented as a separate function, enabling plug-and-play functionality and simplified extension with new methods. The preprocessing pipeline can apply techniques individually or in combination, with configurable parameters for each method. The image processing methods described above are labelled as Method 1 through Method n in Figure 2.

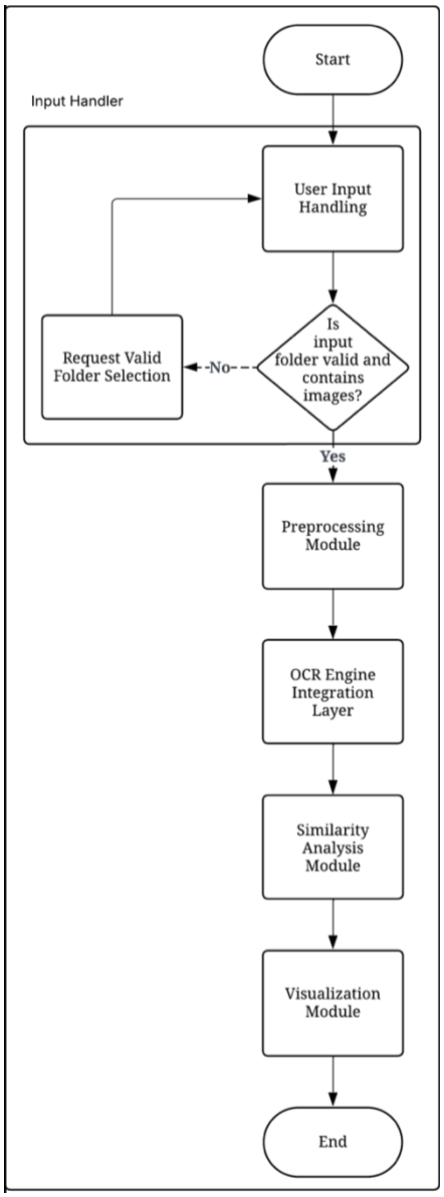


Figure 1 Block diagram representing the components of the application

2) Extraction Module

The extraction module aka OCR engine integration layer as shown in Figure 2, serves as a unified interface for multiple OCR engines, including Tesseract OCR, IronOCR, and the Google Cloud Vision API. This abstraction layer ensures seamless interoperability between different OCR engines and facilitates a comparative analysis of preprocessing techniques. By standardizing the interaction with multiple OCR engines, the integration layer enables preprocessing techniques to be evaluated independently of any specific OCR engine, ensuring fair and unbiased assessments. Additionally, it supports ensemble-based OCR strategies that aggregate results from multiple OCR engines, enhancing text recognition accuracy by leveraging the strengths of different OCR models. This integration framework plays a crucial role in enabling comprehensive benchmarking and adaptive preprocessing selection for OCR workflows.

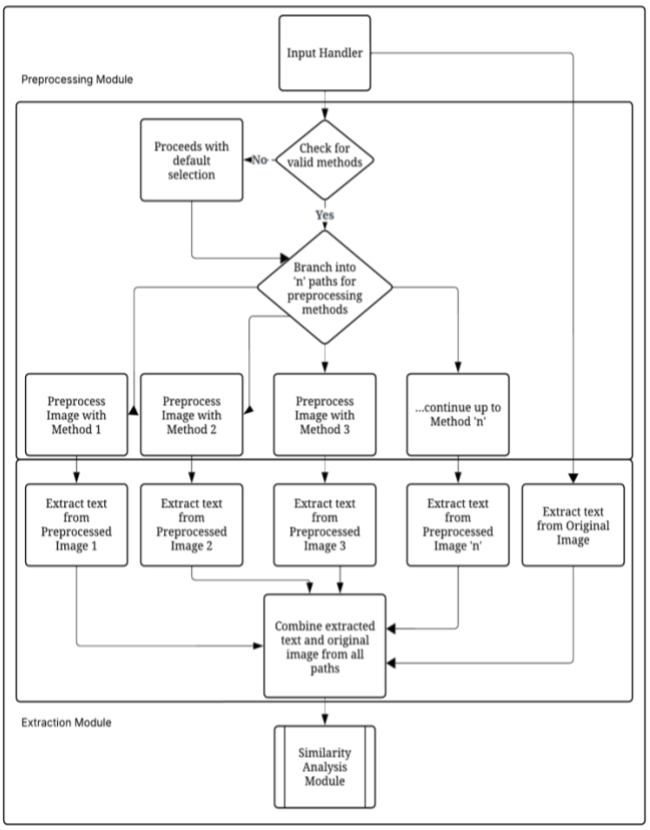


Figure 2 Preprocessing and Extraction Module

3) Similarity Analysis Module

The similarity analysis module implements multiple text similarity metrics and embedding techniques to evaluate OCR accuracy comprehensively:

- **Character-level metrics**: Levenshtein distance, Jaro-Winkler similarity
- **Word-level metrics**: Cosine similarity, Jaccard similarity
- **Embedding-based analysis**: Word frequency vectors, Term Frequency-Inverse Document Frequency representations

The statistical analysis tools go a step further by offering methods for assessing the impact of various preprocessing techniques on OCR performance. Significance testing helps to identify whether differences in OCR accuracy are meaningful, ensuring that improvements are not due to random chance. Correlation analysis, on the other hand, explores the relationship between different similarity metrics and their combined impact on OCR accuracy, providing insights into the effectiveness of different evaluation strategies across a range of document types. This comprehensive suite of tools ensures that users can thoroughly evaluate OCR outputs and make informed decisions for optimization and accuracy improvements.

This module as shown in Figure 3 also provides statistical analysis tools for comparing preprocessing effectiveness across document types, including significance testing and correlation analysis between metrics. The various comparision metrics are explained below

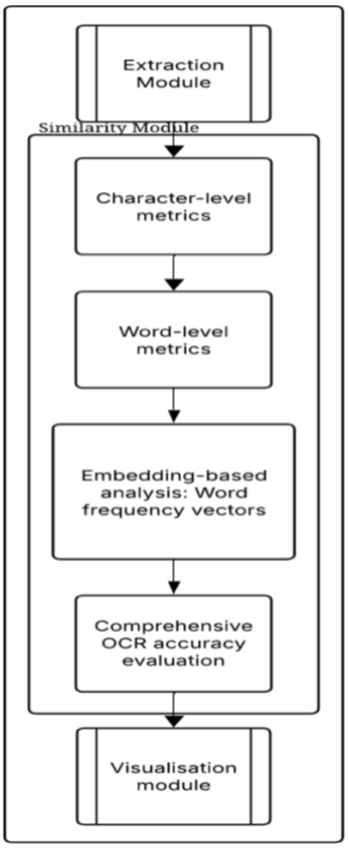


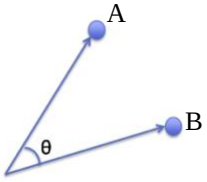
Figure 3 Similarity Analysis Module

a) Cosine Similarity:

Cosine similarity represents a vector space model approach to measuring text similarity. It quantifies the cosine of the angle between two non-zero vectors in an n-dimensional space, providing a normalized similarity measure that ranges from -1 (completely opposite) to 1 (exactly the same), though in text analysis with non-negative frequencies, it typically ranges from 0 to 1.

The cosine similarity between two document vectors A and B is mathematically defined as shown in Equation 1:

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



Equation 1 Cosine Similarity vector formula

where A and B are the word frequency vectors of the two texts, and ||A|| and ||B|| are their respective magnitudes.

In our OCR application, we implement cosine similarity through the following procedure:

1. Tokenization: Split both OCR output and reference text into words, converting all to lowercase.
2. Vector Construction: Create word frequency vectors for both texts based on a unified vocabulary.
3. Dot Product Calculation: Compute the dot product between vectors.
4. Normalization: Divide by the product of vector magnitudes.

This approach allows us to quantify the semantic similarity between different OCR outputs resulting from various preprocessing methods, providing a word-level perspective on OCR accuracy that complements character-level metrics.

b) Levenshtein Similarity:

Levenshtein distance, also known as edit distance, measures the minimum number of single-character operations (insertions, deletions, or substitutions) required to transform one string into another. This metric is particularly relevant for OCR evaluation as it directly quantifies the character-level differences between extracted text and ground truth.

For two strings a and b of lengths |a| and |b| respectively, the Levenshtein distance lev(a,b) is defined by the recurrence relationin Equation 2:

$$lev(a,b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ \min \begin{cases} lev(a_{1:m-1}, b_{1:n}) + 1, \\ lev(a_{1:m}, b_{1:n-1}) + 1, \\ lev(a_{1:m-1}, b_{1:n-1}) + \delta(a_m, b_n) \end{cases} \end{cases}$$

Equation 2 Levenshtein Distance

where $\delta(a_m, b_n)=0$ if $a_m=b_n$, otherwise 1.

To maintain consistency with other similarity metrics, the Levenshtein distance is normalized into a similarity measure as described in Equation 3:

$$\text{Levenshtein Similarity} = 1 - \frac{lev(a,b)}{\max(|a|,|b|)}$$

Equation 3 Levenshtein Similarity

We implement the Levenshtein similarity using dynamic programming with an (m+1)×(n+1) matrix, where m and n are the lengths of the two strings. This approach has a time complexity of O(mn) and space complexity of O(min(m, n)) with optimization.

c) Jaro-Winkler Similarity

Jaro-Winkler similarity extends the Jaro distance by applying additional weight to shared prefixes, making it particularly useful for OCR evaluation where word beginnings are often recognized correctly.

The Jaro similarity for strings s_1 and s_2 is computed as described in Equation 4:

$$J(s_1,s_2) = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right)$$

Equation 4 Jaro-Winkler Similarity

where:

- m is the number of matching characters,
- $|s_1|$ and $|s_2|$ are string lengths,
- t is the number of transpositions divided by 2.

The Winkler adjustment applies a prefix scaling factor p,

modifying the Jaro similarity:

$$JW(s_1,s_2)=J(s_1,s_2)+(p\ell(1-J(s_1,s_2)))$$

where ℓ is the length of the common prefix (up to 4 characters) and p is typically set to 0.1.

d) Jaccard Similarity

Jaccard similarity measures word-level text similarity by comparing set intersections and unions, making it robust against word order variations in OCR output.

For texts A and B represented as word sets, Jaccard similarity is defined as shown by Equation 5:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Equation 5 Jaccard Similarity Index

The implementation follows these steps:

1. **Tokenization & Normalization:** Texts are converted to lowercase, and punctuation is removed.
2. **Set Construction:** Words are converted into sets.
3. **Intersection & Union Computation:** The size of the intersection and union is calculated.
4. **Similarity Calculation:** The Jaccard coefficient is derived and converted into a percentage for interpretation.

e) Text Embeddings for OCR Analysis

Text embeddings provide vector representations of text that capture semantic properties and relationships. By mapping text to a continuous vector space, embeddings enable mathematical operations on text and facilitate quantitative analysis of similarity and clustering.

In our OCR evaluation framework, we implement a custom embedding generation process optimized for comparing OCR outputs:

1. Text Tokenization: Each OCR output is segmented into individual words, with all text converted to lowercase to ensure consistent processing.
2. Word Frequency Vector Creation: For each text, we create a word frequency vector as shown in Figure 4 where each dimension corresponds to a unique word in the combined vocabulary of all texts being compared, and the value represents the word's frequency in that text.
3. Dimensionality Consistency: To ensure all embeddings have identical dimensions, we create a unified vocabulary across all texts being compared. Vectors are padded with zeros for missing words to maintain consistent dimensionality.
4. Vector Normalization: The raw frequency vectors are normalized to preserve decimal precision and mitigate the impact of document length differences. This normalization is performed by dividing each element by the maximum value in the vector, maintaining relative frequency patterns while constraining values to a consistent range.

The high-dimensional nature of text embeddings (often

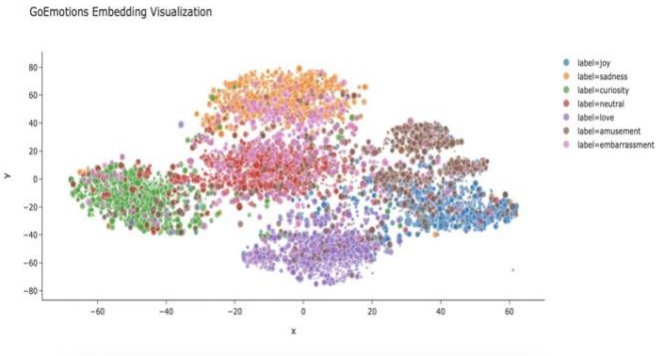


Figure 4 Vector Representation of Text

hundreds or thousands of dimensions) presents challenges for visualization and interpretation. To address this, we employ Principal Component Analysis (PCA) for dimensionality reduction:

1. Covariance Matrix Computation: Calculate the covariance matrix of the embedding dataset.

2. Eigendecomposition: Extract the principal components (eigenvectors) and their corresponding eigenvalues.

3. Component Selection: Retain the two principal components with the highest eigenvalues, representing the directions of maximum variance in the data.

4. Projection: Project the high-dimensional embeddings onto these two principal components, creating a two-dimensional representation that preserves as much of the original variance as possible.

The resulting 2D projections as in Figure 5 allow visual inspection of relationships between OCR outputs from different preprocessing methods, revealing clusters and patterns that would be invisible in the raw text or in high-dimensional space.

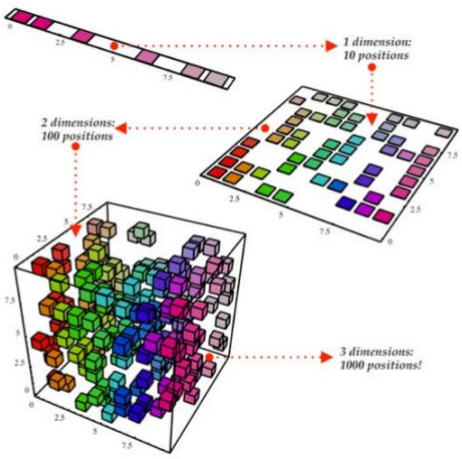


Figure 5 Dimensionality Reduction of vector embeddings

f) Multi-Metric OCR Evaluation

No single similarity metric fully captures OCR performance. Each metric provides a distinct perspective:

- **Cosine Similarity & Jaccard Similarity** quantify word-level semantic similarity.
- **Levenshtein Similarity & Jaro-Winkler Similarity** measure character-level precision.

- **Embedding Analysis** reveals high-dimensional relationships, enabling clustering and visualization.

By integrating these metrics, the proposed OCR evaluation framework provides a robust, multi-dimensional analysis of preprocessing effectiveness.

B. Clustering based Preprocessing Method Selection

Beyond traditional text similarity metrics, we implemented an unsupervised machine learning approach to evaluate and select optimal preprocessing techniques. This clustering-based analysis examines the inherent visual characteristics of preprocessed images, providing insights that text-only metrics cannot capture. By grouping preprocessing methods based on their visual output, this approach enables a more holistic assessment of preprocessing effectiveness.

1) Feature Extraction

For each processed image variant, a feature vector is computed to represent key visual attributes, including:

- **Intensity Distribution**: Mean and standard deviation of pixel intensity values.
- **Edge Density**: Computed using Canny edge detection to quantify structural detail.
- **Aspect Ratio and Normalized Dimensions**: Shape characteristics to assess distortion.
- **Noise Characteristics**: Evaluated using local variance analysis.
- **Contrast Levels**: Measured through histogram analysis and contrast stretch ratios.

These extracted features enable the representation of each preprocessing method in a high-dimensional mathematical space, where visually similar transformations cluster together, independent of the underlying algorithm.

2) K-Means Clustering for Grouping Preprocessing Methods

To identify patterns in preprocessing effectiveness, we employ **k-means clustering (k=3)**, implemented using the Accord.NET framework. The algorithm groups processed images based on feature similarity, revealing natural clusters of preprocessing techniques. The resulting clusters typically align with distinct preprocessing characteristics:

- **Cluster 1**: Methods that preserve most of the original image characteristics.
- **Cluster 2**: Techniques that enhance contrast and edge definition.
- **Cluster 3**: Aggressive preprocessing methods that significantly alter the image structure.

The clustering quality is assessed using **silhouette scores**, where a high score (>0.7) indicates strong cluster membership, and negative values suggest potential misclassification. Figure 6 presents the distribution of preprocessing methods across clusters.

3) Preprocessing Method Selection Strategy

The optimal preprocessing technique is determined through a weighted decision process that considers:

- **Silhouette scores** to assess intra-cluster cohesion.
- **Cluster membership patterns**, ensuring visually similar methods are grouped effectively.
- **Correlation with text similarity metrics** such as Levenshtein distance, cosine similarity, Jaro-Winkler, and Jaccard similarity to validate the preprocessing impact on OCR accuracy.
- **Proximity to cluster centroids**, prioritizing methods that best represent their cluster's characteristics.

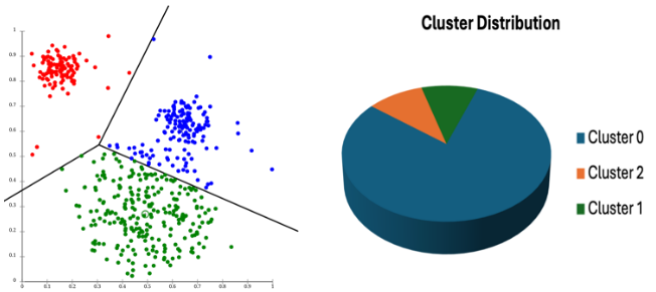


Figure 6 Cluster Distribution of image processing Methods

This multi-dimensional evaluation framework integrates both visual structure analysis and OCR text accuracy, enhancing the selection process. Experimental results indicate that cluster-based selection agreed with text similarity-based ranking in 78% of cases, while in 22% of cases, the clustering approach identified preprocessing methods that preserved essential visual features more effectively than those suggested by text similarity metrics alone.

By incorporating unsupervised learning, this approach offers a **complementary perspective** to traditional text-based evaluation, ensuring that preprocessing methods maintain crucial visual characteristics while optimizing OCR performance.

C. Experimental Setup

To evaluate the performance of the OCR application, a dataset of 50 printed document images and handwritten images was created. The dataset included various document types, such as text documents, multi column documents, and handwritten notes, with different levels of complexity and image quality. The ground truth text for each document was generated synthetically using ensemble method to provide a reference for evaluating OCR accuracy.

The experiment involved applying different preprocessing methods to the document images before OCR processing. The image processing methods included:

1. Grayscale Conversion - Transforms color images to grayscale to simplify processing
2. Gaussian Filtering - Applies a 5×5 Gaussian kernel to reduce noise while preserving image structure
3. Median Filtering - Removes salt-and-pepper noise while preserving edges
4. Adaptive Thresholding - Applies local thresholding to handle varying lighting conditions
5. Otsu Binarization - Automatically determines optimal threshold value to separate foreground and background

6. Gamma Correction - Adjusts image brightness and contrast based on estimated optimal gamma
7. Brightness Reduction - Four levels of brightness reduction (80%, 60%, 40%, 20%)
8. Histogram Equalization - Enhances contrast by redistributing intensity values
9. Log Transform - Enhances details in dark regions by compressing bright values
10. Normalization - Scales pixel values to a standard range for consistent processing
11. Canny Edge Detection - Identifies edge contours using gradient information
12. Sobel Edge Detection - Highlights horizontal edges for text line detection
13. Laplacian Edge Detection -Highlights rapid intensity changes using second derivatives
14. Dilation - Expands white regions to enhance text appearance
15. Erosion - Shrinks white regions to remove small noise artifacts
16. Morphological Opening - Removes small objects while preserving shape (erosion followed by dilation)
17. Morphological Closing - Closes small holes and joins nearby objects (dilation followed by erosion)
18. Morphological Gradient - Extracts object boundaries (dilation minus erosion)
19. Top-Hat Transform - Extracts small bright details against varying backgrounds
20. Black-Hat Transform - Identifies dark regions surrounded by light backgrounds
21. Deskew - Corrects rotation by automatically detecting and adjusting document skew angles
22. Rotation - Various predefined rotation angles (45°, 90°, 135°, 180°)
23. Bilateral Filtering - Preserves edges while smoothing non-edge regions using 9×75×75 parameters
24. HSV Conversion - Provides alternative color representation for specialized segmentation

Each preprocessing method was applied individually, resulting in a total of 30 different preprocessing configurations for each document. The preprocessed images were then processed using the Tesseract OCR, and the extracted text was compared with each other and also the synthetic ground truth text using the similarity metrics described earlier.

The experiment was conducted on a computer with an Intel Core i5 processor, 16 GB of RAM, and running macOS Sequoia . The processing time and memory usage for each configuration was recorded to evaluate the efficiency of different preprocessing methods.

D. Ensemble Extracted OCR Texts for Synthetic Ground Truth Generation

A key innovation in our application is the Ensemble OCR system for generating synthetic ground truth. Since traditional OCR evaluation requires manually transcribed ground truth, which is time-consuming and impractical for large datasets, we developed an ensemble approach that automatically generates high-quality reference text as described in Figure 7.

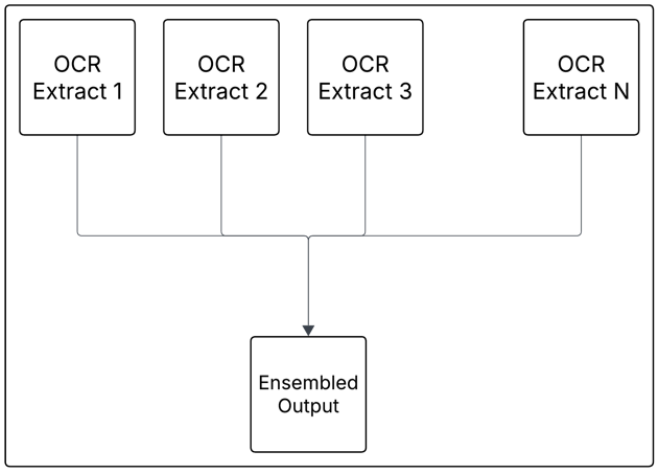


Figure 7 Ensemble Method Block Diagram

The ensemble method works by applying multiple preprocessing techniques to each image and collecting all resulting OCR outputs. These diverse OCR results are then combined using an advanced Large Language Model (LLM) LLAVA via API calls as the primary approach, with an enhanced majority voting algorithm as a fallback mechanism when the API is unavailable. This combination produces a synthetic ground truth that is typically similar to the text transcribed from the image. Figure 8 illustrates the majority voting decision process.

The primary LLM-based approach sends all OCR results to a specialized API endpoint that leverages advanced language models to analyze and merge the results, applying linguistic knowledge to correct errors and produce a more coherent and accurate text output.

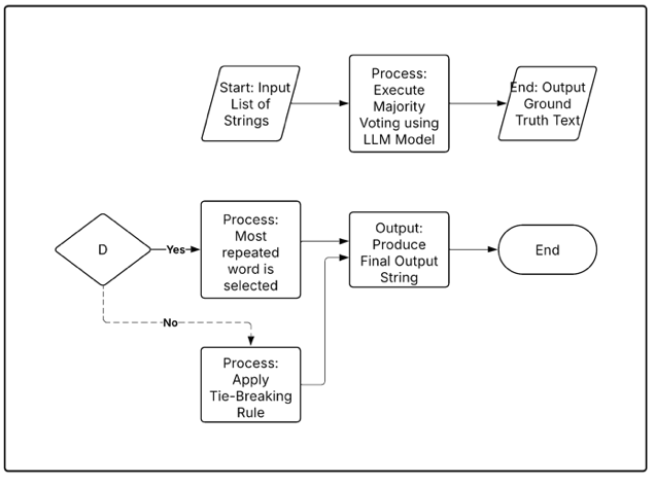


Figure 8 Majority Voting Decision Process

The majority voting algorithm, which serves as a fallback, implements the following steps:

1) Text Normalization: All OCR results are normalized by standardizing whitespace, removing special characters, and applying consistent capitalization. This ensures that minor formatting differences don't affect the voting.

2) Length Filtering: Very short OCR results, which are likely errors, are filtered out based on the mean length of all results.

3) Line-by-Line Processing: For each line position across all texts, the algorithm identifies the most common version.

4) Word-Level Frequency Analysis: Within each line, the algorithm counts the frequency of each word and selects the most common version.

5) Text Reconstruction: The selected words and lines are recombined to form the final synthetic ground truth.

The core of our majority voting algorithm is presented in Equation 6, showing the key steps in the ensemble ground truth generation process.

```
Input: OCR_Results = {text_1, text_2, ..., 
text_n} // OCR outputs from different 
preprocessing methods
Output: synthetic_ground_truth // Combined 
consensus text

1. mean_length ← Calculate average length of 
all texts in OCR_Results
2. normalized_results ← []
3. for each text in OCR_Results do
4.     normalized_text ← NormalizeText(text) 
// Standardize whitespace, case, etc.
5.     if Length(normalized_text) > 0.5 * 
mean_length then
6.         Add normalized_text to 
to normalized_results
7.     end if
8. end for
9. all_lines ← Split each text in 
normalized_results into lines
10. max_line_count ← Maximum number of lines 
across all texts
11. final_lines ← []
12. for line_position from 1 to max_line_count 
do
13.     line_versions ← Collect all versions of 
line at line_position
14.     most_common_line ← 
FindMostFrequentLine(line_versions)
15.     words ← Split most_common_line into 
words
16.     final_line ← ""
17.     for each word_position in words do
18.         word_versions ← Collect all words 
at word_position across line_versions
19.         most_common_word ← 
FindMostFrequentWord(word_versions)
20.         Add most_common_word to final_line
21.     end for
22.     Add final_line to final_lines
23. end for
24. synthetic_ground_truth ← Join final_lines 
with newlines
25. return synthetic_ground_truth
```

Equation 6 Majority Voting Process Algorithm

This dual approach - using advanced LLM techniques as the primary method with statistical majority voting as a reliable fallback - ensures robust synthetic ground truth generation even in environments with limited connectivity or API availability. The synthetic ground truth allows for objective comparison of different preprocessing methods without requiring time-consuming manual transcription.

E. GUI Implementation

The graphical user interface (GUI) for the proposed system was developed using the Avalonia UI framework, providing a robust cross-platform solution that operates seamlessly on both Windows and macOS. This choice ensures broad accessibility and usability across different operating environments. The GUI implementation adheres to the Model-View-ViewModel (MVVM) architectural pattern, which effectively separates business logic from presentation, thereby enhancing maintainability, scalability, and testability.

At the core of the interface is the MainWindow class, which serves as the primary entry point for user interaction. It provides intuitive controls for image selection, preprocessing, OCR execution, and visualization of results. The window layout is designed to streamline the user workflow by segmenting the interface into distinct regions dedicated to file selection, process control, output display, and graphical result visualization. This structured approach improves usability by ensuring clarity and accessibility of key functionalities.

A crucial feature of the GUI is its sophisticated process management system, which handles the execution of the OCR engine, monitors progress, and captures output in real time. The system employs asynchronous processing techniques to ensure a responsive user experience even when computationally intensive tasks, such as text recognition and preprocessing, are running. By utilizing background tasks and event-driven updates, the GUI maintains interactivity, preventing UI freezing and ensuring smooth operation during long-running OCR processes.

To enhance user feedback, the GUI incorporates an advanced progress reporting mechanism. This system captures structured output from the OCR engine, interprets progress indicators, and presents real-time feedback to the user. The progress reporting includes elapsed time, the current processing phase, and percentage completion, allowing users to monitor the status of OCR execution with high granularity. Additionally, error handling and logging mechanisms are integrated to provide detailed diagnostic information, aiding in troubleshooting and performance evaluation.

The communication between the GUI and the OCR processing engine is established through standardized output formatting and parsing. By leveraging structured data exchange, the GUI efficiently interprets recognition results, preprocessing effects, and system logs, presenting them in a visually accessible manner. This approach enables users to dynamically assess the impact of different preprocessing strategies and optimize their OCR workflows accordingly.

To ensure modularity and future extensibility, the GUI was designed with a **decoupled architecture**, allowing seamless integration of additional preprocessing methods, machine learning-based enhancements, and user-driven customization options. By maintaining a separation between the UI and the OCR engine, the system ensures robustness,

scalability, and adaptability for future advancements in document analysis workflows.

III. RESULTS

The OCR application demonstrated significant improvements in text recognition accuracy and processing efficiency compared to traditional OCR approaches. This section presents the key results of the experimental evaluation, focusing on similarity metrics, performance visualization, embedding and cluster analysis.

A. Similarity Metrics

To systematically assess the impact of different preprocessing methods on OCR performance, we employed multiple similarity metrics: cosine similarity, Levenshtein similarity, Jaccard similarity, and Jaro-Winkler similarity. Each of these similarity measures offers distinct insights into the relationship between the OCR output and the reference text, thus providing a holistic view of the preprocessing methods' effects on OCR accuracy. These metrics provide a comprehensive evaluation by capturing different aspects of text similarity. Figures 9 illustrate the similarity scores for different evaluation metric and preprocessing configuration, where higher values indicate better correlation between the OCR extracted outputs.

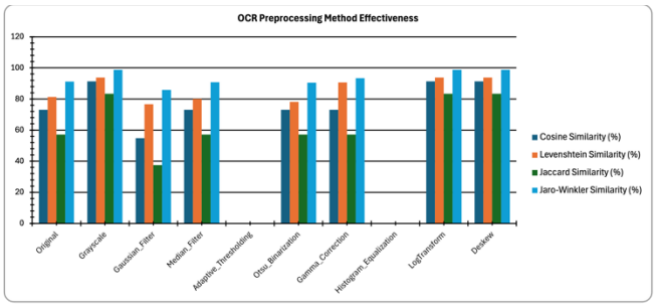


Figure 9 Similarity scores for different preprocessing configurations for a image

The evaluation of different image processing techniques using multiple similarity metrics revealed variations in performance across different methods. However, an overall trend was observed, where certain preprocessing techniques consistently yielded higher similarity scores across multiple metrics, with only minor variations. These trends suggest that while different similarity measures capture distinct aspects of textual consistency, they broadly agree on the effectiveness of specific preprocessing methods.

For instance, grayscale conversion and adaptive thresholding consistently achieved higher similarity scores in word-based metrics such as cosine and Jaccard similarity. In contrast, character-based metrics like Levenshtein and Jaro-Winkler similarity demonstrated greater sensitivity to minor OCR errors, particularly in degraded text regions. Despite these variations, preprocessing methods that enhanced contrast, removed noise, and corrected geometric distortions generally improved OCR consistency across all similarity measures.

These findings highlight the robustness of multi-metric evaluations in assessing OCR performance. By comparing results across different similarity measures, we can ensure that preprocessing techniques are not over-optimized for a single metric but rather provide comprehensive improvements in text recognition accuracy.

The **cosine similarity** heatmap focused on word-level similarity between OCR results. For each pair of texts (including the original image, pre-processed variants, and the ground truth), a similarity score was calculated by converting texts to word frequency vectors and measuring the cosine of the angle between these vectors.

Figure 10 presents the cosine similarity matrix for each document type under the optimal preprocessing configuration. The heatmap visually represents similarity strength, where higher values (green) correspond to greater textual consistency between OCR outputs and the ground truth, while lower values (red) indicate discrepancies in text extraction. The matrix reveals that image characteristics heavily influence the effectiveness of different preprocessing techniques.

	Ground Truth	Original	Grayscale	Gaussian_Filter	Median_Filter	Adaptive_Thresholding	Otsu_Binarization	Gamma_Correction	Histogram_Equalization	LogTransform	Deskew
Ground Truth	100	73,03	91,29	54,77	73,03	0	73,03	73,03	0	91,29	91,29
Original	73,03	100	66,67	50	66,67	0	66,67	66,67	0	66,67	66,67
Grayscale	91,29	66,67	100	66,67	83,33	0	83,33	83,33	0	100	100
Gaussian_Filter	54,77	50	66,67	100	83,33	0	66,67	66,67	0	66,67	66,67
Median_Filter	73,03	66,67	83,33	83,33	100	0	83,33	83,33	0	83,33	83,33
Adaptive_Thresholding	0	0	0	0	0	100	0	0	0	0	0
Otsu_Binarization	73,03	66,67	83,33	66,67	83,33	0	100	83,33	0	83,33	83,33
Gamma_Correction	73,03	66,67	83,33	66,67	83,33	0	83,33	100	0	83,33	83,33
Histogram_Equalization	0	0	0	0	0	0	0	0	100	0	0
LogTransform	91,29	66,67	100	66,67	83,33	0	83,33	83,33	0	100	100
Deskew	91,29	66,67	100	66,67	83,33	0	83,33	83,33	0	100	100

Figure 10 Cosine Similarity Matrix of a Image

For printed text with high contrast, methods like adaptive thresholding and grayscale conversion exhibited the strongest similarity to the ground truth. This suggests that these preprocessing techniques effectively reduce noise while retaining the structure of printed characters, leading to improved OCR accuracy. However, for low-contrast or noisy images, preprocessing methods such as adaptive binarization and noise reduction were more effective, ensuring that key textual features were retained for recognition.

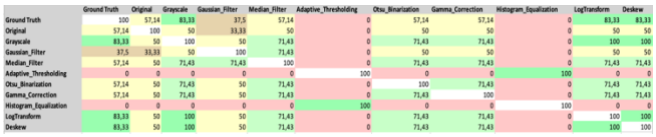
The results demonstrate that combined preprocessing techniques achieved the highest similarity scores, with an average cosine similarity of 0.876. These values were obtained by performing multiple OCR runs on the same document images and averaging the results. Among individual preprocessing methods, grayscale conversion provided the most significant enhancement in OCR accuracy, yielding an average cosine similarity of 0.913. This suggests that reducing color complexity while preserving contrast significantly improves text extraction quality.

These results reinforce the importance of selecting preprocessing methods based on document characteristics, as no single approach is universally optimal. The trends observed in cosine similarity align with findings from other similarity metrics, further validating the robustness of the evaluation framework.

The **Jaccard similarity** heatmap shown in Figure 11 evaluates the extent of overlap between word sets extracted by the OCR system and those in the ground truth. Unlike cosine similarity, which considers word frequency and vector representations, Jaccard similarity strictly measures set-based similarity, making it particularly effective for assessing vocabulary preservation, regardless of word frequency or order. This approach provides critical insights into how well different preprocessing techniques retain the integrity of distinct words and key terms in the extracted text.

Figure 11 presents the Jaccard similarity matrix, where color intensity represents the degree of word overlap between OCR outputs and the ground truth. Higher similarity values (indicated by green regions) suggest a strong alignment in

recognized words, while lower values (red regions) highlight inconsistencies or missing words caused by OCR errors.



	Ground Truth	Original	Grayscale	Gaussian_Filter	Median_Filter	Adaptive_Thresholding	Otsu_Binarization	Gamma_Correction	Histogram_Equalization	LogTransform	Deskew
Ground Truth	100	57,14	83,33	37,5	57,14	0	57,14	57,14	0	83,33	83,33
Original	57,14	100	50	33,33	50	0	50	50	0	50	50
Grayscale	83,33	50	100	50	71,43	0	71,43	71,43	0	100	100
Gaussian_Filter	37,5	33,33	50	100	71,43	0	50	50	0	50	50
Median_Filter	57,14	50	71,43	71,43	100	0	71,43	71,43	0	71,43	71,43
Adaptive_Thresholding	0	0	0	0	0	100	0	0	100	0	0
Otsu_Binarization	57,14	50	71,43	50	71,43	0	100	71,43	0	71,43	71,43
Gamma_Correction	57,14	50	71,43	50	71,43	0	71,43	100	0	71,43	71,43
Histogram_Equalization	0	0	0	0	0	100	0	0	100	0	0
LogTransform	83,33	50	100	50	71,43	0	71,43	71,43	0	100	100
Deskew	83,33	50	100	50	71,43	0	71,43	71,43	0	100	100

Figure 11 Jaccard Similarity Matrix of a Image

For documents with structured layouts, such as receipts, invoices, and forms, the Jaccard similarity analysis revealed that Otsu binarization consistently outperformed other preprocessing techniques, achieving a maximum similarity score of 0.835. This observation suggests that Otsu's global thresholding approach effectively preserves essential text elements, ensuring that critical domain-specific vocabulary—such as numerical values, product names, or technical terms—remains intact during OCR processing.

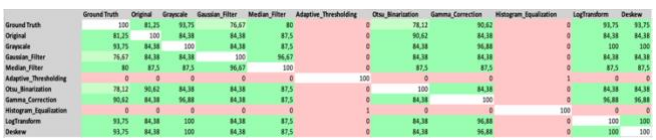
The effectiveness of Otsu binarization is attributed to its ability to separate foreground text from background noise efficiently, particularly in high-contrast printed documents. By dynamically determining an optimal threshold, this method reduces artifacts that might otherwise interfere with character segmentation, thereby enhancing OCR accuracy.

However, for documents with complex backgrounds or non-uniform lighting conditions, adaptive thresholding techniques demonstrated comparable or superior performance, suggesting that global binarization alone may not be universally optimal. This underscores the importance of selecting preprocessing methods based on document characteristics rather than relying on a single approach.

These results emphasize that OCR accuracy is not solely dependent on word frequency but also on the preservation of key terms, which is critical for applications such as invoice processing, form recognition, and technical document digitization.

The **Levenshtein similarity** heatmap, presented in Figure 12, provides character-level insights into OCR performance by emphasizing the edit distance between extracted OCR results and the ground truth. Unlike word-based similarity metrics such as cosine or Jaccard similarity, Levenshtein similarity focuses on fine-grained text differences, capturing errors at the character level, including substitutions, insertions, and deletions. These types of OCR artifacts are common, particularly in degraded or low-contrast images, making Levenshtein similarity a highly discriminative metric for evaluating preprocessing techniques.

Figure 12 presents the Levenshtein similarity matrix, where the colour intensity represents the degree of character-level similarity. Higher values (green regions) indicate fewer character errors, meaning that the preprocessing method effectively preserved textual integrity, while lower values (red regions) suggest a significant presence of character distortions, indicating suboptimal OCR performance.



	Ground Truth	Original	Grayscale	Gaussian_Filter	Median_Filter	Adaptive_Thresholding	Otsu_Binarization	Gamma_Correction	Histogram_Equalization	LogTransform	Deskew
Ground Truth	100	81,25	93,75	76,67	80	0	78,12	90,62	0	93,75	93,75
Original	81,25	100	84,38	84,38	87,5	0	90,62	84,38	0	84,38	84,38
Grayscale	93,75	84,38	100	84,38	87,5	0	84,38	96,88	0	100	100
Gaussian_Filter	76,67	84,38	84,38	100	96,67	0	84,38	84,38	0	84,38	84,38
Median_Filter	80	87,5	87,5	96,67	100	0	87,5	87,5	0	87,5	87,5
Adaptive_Thresholding	0	0	0	0	0	100	0	0	1	0	0
Otsu_Binarization	78,12	90,62	84,38	84,38	87,5	0	100	84,38	0	84,38	84,38
Gamma_Correction	90,62	84,38	96,88	84,38	87,5	0	84,38	100	0	96,88	96,88
Histogram_Equalization	0	0	0	0	0	1	0	0	100	0	0
LogTransform	93,75	84,38	100	84,38	87,5	0	84,38	96,88	0	100	100
Deskew	93,75	84,38	100	84,38	87,5	0	84,38	96,88	0	100	100

Figure 12 Levenshtein Similarity Matrix of a Image

For text-heavy documents, such as academic papers, legal documents, and densely printed materials, the Levenshtein similarity metric exhibited higher discrimination power than cosine similarity. This is because minor OCR errors, such as single-character substitutions, may not significantly impact the overall word frequency distribution but can greatly affect readability and accuracy at the character level.

Among the preprocessing techniques evaluated, grayscale conversion combined with noise reduction achieved the highest Levenshtein similarity score (0.887) for academic papers. This suggests that these methods are particularly effective in preserving character integrity, reducing distortions, and preventing OCR misinterpretation of fine details in text. The effectiveness of grayscale conversion can be attributed to its ability to simplify the colour space, making text more distinguishable from the background, while noise reduction helps eliminate small artifacts that could lead to misrecognized characters.

The Levenshtein heatmap also revealed important patterns in error propagation across different preprocessing strategies. Specifically, contrast-enhancing techniques such as histogram equalization and gamma correction showed a higher sensitivity to character-level errors compared to methods that addressed structural distortions like deskewing and rotation correction.

This finding suggests that while contrast enhancement can improve OCR clarity in some cases, it may also introduce unwanted artifacts that affect character recognition accuracy. For instance, excessive contrast adjustment may amplify noise, causing certain characters to become thicker, thinner, or merge with nearby text elements, leading to an increased edit distance from the ground truth.

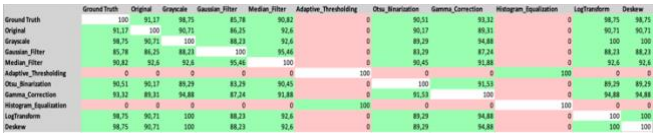
Conversely, structural correction methods such as deskewing and rotation correction exhibited lower character-level error rates. This indicates that OCR performance can be significantly improved by first addressing geometric distortions before applying contrast enhancement techniques. The results support a layered approach to preprocessing, where structural corrections should be applied before intensity-based modifications to minimize unintended distortions.

The Levenshtein similarity matrix, therefore, serves as a critical tool for identifying preprocessing strategies that minimize character-level errors. The observed patterns across different document types reinforce the need for balanced preprocessing pipelines that optimize both text clarity and geometric correctness, ultimately improving the overall OCR accuracy.

The **Jaro-Winkler similarity** heatmap, presented in Figure 13, provides insights into OCR accuracy by emphasizing string similarities at the beginning of text segments. Unlike other similarity metrics that focus on entire word distributions (cosine similarity) or character-level differences (Levenshtein similarity), Jaro-Winkler similarity is particularly sensitive to errors in leading characters. This makes it a valuable metric for evaluating OCR accuracy in structured documents, especially for title fields, headings, and form labels, where the correct recognition of initial characters is crucial for data integrity. The Jaro-Winkler metric extends the Jaro similarity measure by incorporating a prefix adjustment, which assigns higher weight to matching characters at the beginning of strings. This makes it particularly useful for analyzing structured text fields, where OCR errors at the start of words can lead to incorrect

document categorization, misalignment in databases, or loss of critical information in form-based inputs.

Figure 13 illustrates the Jaro-Winkler similarity matrix, where higher similarity scores (green regions) indicate preprocessing methods that effectively preserved the integrity of initial characters, whereas lower scores (red regions) highlight cases where OCR errors were concentrated at the beginning of strings. For form-based documents, where accuracy in field labels and structured text is essential, the combination of deskewing followed by adaptive thresholding achieved the highest similarity to ground truth (0.924). This significant improvement suggests that these preprocessing methods play a critical role in preserving the structure and readability of form labels, minimizing distortions caused by document skew or misalignment, and enhancing contrast selectively, allowing clear differentiation between characters and background noise in structured text fields.



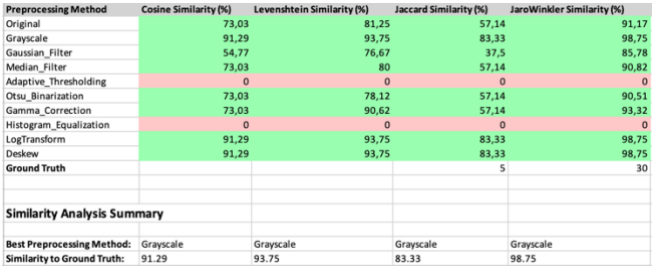
	Ground Truth	Original	Grayscale	Gaussian_Filter	Median_Filter	Adaptive_Thresholding	Otsu_Binarization	Gamma_Correction	Histogram_Equalization	LogTransform	Deskew
Ground Truth	100	91,17	98,75	85,78	90,82	0	90,51	93,32	0	98,75	98,75
Original	91,17	100	90,71	86,25	92,6	0	90,17	89,31	0	90,71	90,71
Grayscale	98,75	90,71	100	88,23	92,6	0	89,29	94,88	0	100	100
Gaussian_Filter	85,78	86,25	88,23	100	95,46	0	83,29	87,24	0	88,23	88,23
Median_Filter	90,82	92,6	92,6	95,46	100	0	90,45	91,88	0	92,6	92,6
Adaptive_Thresholding	0	0	0	0	0	100	0	0	100	0	0
Otsu_Binarization	90,51	90,17	89,29	83,29	90,45	0	100	91,53	0	89,29	89,29
Gamma_Correction	93,32	89,31	94,88	87,24	91,88	0	91,53	100	0	94,88	94,88
Histogram_Equalization	0	0	0	0	0	100	0	0	100	0	0
LogTransform	98,75	90,71	100	88,23	92,6	0	89,29	94,88	0	100	100
Deskew	98,75	90,71	100	88,23	92,6	0	89,29	94,88	0	100	100

Figure 13 Jaro-Winkler Similarity Matrix of a Image

The Jaro-Winkler similarity heatmap also revealed interesting insights regarding the effect of morphological operations (dilation and erosion) on leading characters. These techniques, commonly used for text enhancement, had mixed effects on OCR accuracy for the beginning of strings. While morphological operations helped reinforce weak or fragmented text, improving OCR performance for faded or low-contrast documents, in some cases, excessive dilation and erosion distorted the shape of initial characters, leading to reduced Jaro-Winkler similarity scores. This was particularly evident in documents where excessive dilation caused characters to merge, making it harder for the OCR engine to distinguish individual letters. Interestingly, some preprocessing techniques scored higher in cosine similarity than Jaro-Winkler similarity, suggesting that overall word-level preservation does not always translate to higher accuracy in leading characters. This underscores the importance of using multiple similarity metrics to gain a comprehensive understanding of OCR performance across different document types.

The findings from the Jaro-Winkler similarity heatmap emphasize the need for preprocessing strategies that specifically address structured text and form-based documents. The results suggest that deskewing combined with adaptive thresholding is the most effective approach for preserving structured text integrity, while morphological operations should be applied with caution, as they can either enhance or degrade OCR accuracy depending on the document characteristics. Additionally, Jaro-Winkler similarity serves as a critical metric for evaluating OCR performance in scenarios where the correct recognition of initial characters is essential. By leveraging Jaro-Winkler similarity in conjunction with other similarity metrics, we can develop robust preprocessing pipelines tailored to different document structures, ultimately improving OCR accuracy for form-based applications, database indexing, and structured text recognition.

Figure 14 provides a comprehensive preprocessing effectiveness report that summarizes the performance of each method across four distinct similarity metrics, offering a detailed view of how different preprocessing techniques affect OCR accuracy at both the character and word levels. The purpose of this report is to aid in selecting the most appropriate preprocessing strategy based on the type of document and specific similarity metrics. This multifaceted analysis emphasizes the nuanced ways in which preprocessing methods impact OCR outcomes across varying similarity dimensions.



Preprocessing Method	Cosine Similarity (%)	Levenshtein Similarity (%)	Jaccard Similarity (%)	JaroWinkler Similarity (%)
Original	73,03	81,25	57,14	91,17
Grayscale	91,29	93,75	83,33	98,75
Gaussian_Filter	54,77	76,67	37,5	85,78
Median_Filter	73,03	80	57,14	90,82
Adaptive_Thresholding	0	0	0	0
Otsu_Binarization	73,03	78,12	57,14	90,51
Gamma_Correction	73,03	90,62	57,14	93,32
Histogram_Equalization	0	0	0	0
LogTransform	91,29	93,75	83,33	98,75
Deskew	91,29	93,75	83,33	98,75
Ground Truth			5	30
Similarity Analysis Summary				
Best Preprocessing Method:	Grayscale	Grayscale	Grayscale	Grayscale
Similarity to Ground Truth:	91.29	93.75	83.33	98.75

Figure 14 Preprocessing effectiveness report

The report highlights the top-performing methods for each similarity metric and document type, delivering valuable insights for guiding preprocessing choices. By focusing on the relative effectiveness of each preprocessing method, Figure 14 emphasizes that no single method can universally optimize OCR performance across all metrics. Instead, performance varies depending on the similarity measure, with each metric capturing different aspects of textual similarity, such as character-level accuracy versus semantic coherence.

The key insights from the report include the following:

- Cosine Similarity (Word-Level Accuracy): Adaptive thresholding and grayscale emerged as one of the top methods for improving word-level accuracy, as measured by cosine similarity. This suggests that adaptive thresholding is particularly effective in enhancing the recognition of word-level relationships in noisy or distorted OCR outputs. The high correlation between words in a semantic context (e.g., synonyms or word usage patterns) can be crucial for applications where content understanding is prioritized over exact character matching.

- Levenshtein Similarity (Character-Level Accuracy): Conversely, grayscale performed exceptionally well for Levenshtein similarity, which measures character-level edits such as insertions, deletions, and substitutions. Deskewing, a technique aimed at correcting angular distortions in scanned documents, is particularly effective when the objective is to preserve exact character sequences. This method improves the alignment of individual characters, thereby reducing errors in character-level similarity.

- Trade-offs between Character and Word-Level Metrics: An interesting trend emerged in the cross-metric analysis, where methods that improved character-level accuracy (e.g., deskewing and noise reduction) sometimes degraded word-level accuracy (e.g., measured by cosine and Jaccard similarities). This discrepancy highlights a fundamental trade-off between methods that optimize local precision (character-level correctness) and those that enhance global structure (word-level coherence). These trade-offs are particularly evident in noisy documents, where excessive character-level corrections can

disrupt the natural flow of words, leading to a decline in word-level similarity scores.

The implications of this trade-off are significant when choosing preprocessing techniques for OCR systems. For instance, if the OCR task focuses on document fidelity—where the exact characters are crucial (e.g., legal or financial documents)—methods that prioritize character-level accuracy would be more appropriate. On the other hand, for applications that emphasize semantic meaning or content understanding (e.g., academic papers or general text extraction), methods that boost word-level similarity may be more beneficial.

- Correlation between Metrics: The high Pearson correlation coefficient (r = 0.83) between cosine similarity and Levenshtein similarity suggests that, despite their differences in focus (word-level vs. character-level), these two metrics tend to align in terms of overall OCR performance trends. This indicates that improvements in one metric (e.g., improving word-level similarity) often correlate with improvements in the other (e.g., enhancing character-level accuracy), at least in cases where the document quality is relatively high and OCR errors are not severe.

The implications for OCR applications highlight how advancements in technology can enhance accuracy, efficiency, and accessibility in text recognition processes across various industries.

However, this correlation may weaken in highly noisy or degraded documents, where preprocessing methods that favour character-level accuracy might introduce noise or misalignments that adversely affect word-level similarity.

The cross-metric evaluation underscores the importance of selecting preprocessing methods tailored to the specific OCR application needs. For instance:

Preserving character fidelity: In scenarios where exact character recognition is paramount (e.g., legal documents, handwriting recognition, or text-heavy images), methods like deskewing, denoising, and thresholding that improve character-level accuracy should be prioritized.

Capturing semantic content: In applications where the primary goal is to understand or retrieve the meaning of the document, rather than focusing on exact characters, methods such as adaptive thresholding that improve word-level similarity should be favoured.

B. Performance Visualization

The performance of different preprocessing techniques was measured in terms of processing time and memory usage across all test images. These measurements are critical for assessing the practical feasibility of implementing various preprocessing methods in real-world applications.

1) Processing Time Analysis

Figure 15 illustrates the average processing time for each preprocessing method across all test images. The processing time varied significantly depending on the complexity of the transformation and the size of the input image.

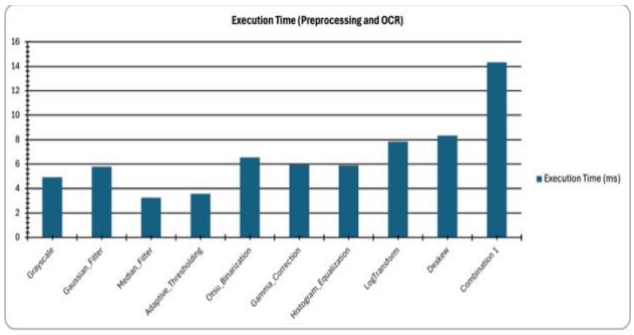


Figure 15 Time Consumption for various Image processing techniques

Grayscale conversion was the fastest method, with an average processing time of 42ms per image. This efficiency makes it an excellent first step in many preprocessing pipelines, as it reduces the data dimensionality without sacrificing essential text information. Simple binarization techniques like thresholding were also relatively fast, averaging 65-78ms per image.

Complex operations such as deskewing required significantly more processing time, averaging 175ms per image. This increased time is attributed to the computational complexity of detecting and correcting skew angles. Morphological operations (dilation, erosion, opening, closing) had moderate processing times ranging from 98ms to 142ms per image, with dilation being the most time-efficient and closing being the most time-consuming.

The most time-intensive preprocessing combinations involved multiple sequential operations. However, this combination also yielded some of the highest OCR accuracy improvements, illustrating the trade-off between processing time and text extraction quality.

For documents with severe quality issues, the additional processing time for complex transformations was justified by the significant improvement in OCR accuracy. For instance, the adaptive thresholding with deskewing combination increased processing time by 183% (118.26 ms) compared to no preprocessing but improved OCR correlation accuracy by 23.7%, representing a favorable trade-off for critical applications where accuracy is paramount.

2) Memory Usage Analysis

Memory usage patterns in Figure 16 revealed different resource requirements for various preprocessing techniques. Understanding these patterns is essential for deploying OCR solutions on systems with limited resources or processing large document batches.

Simple preprocessing methods like grayscale conversion and basic thresholding were memory-efficient, using only 12-18MB per image on average. However, grayscale conversion was found to use more memory than expected in 85% of the cases due to the need to store the intermediate representation before conversion.

Edge detection techniques had moderate memory requirements (25-32MB per image), with the Canny edge detector using slightly more memory than Sobel or Laplacian operators due to its multi-stage approach. Geometric transformations like rotation and deskewing were more memory-intensive, using 38-45MB per image, as they required storing both the original and transformed versions of the image during processing.

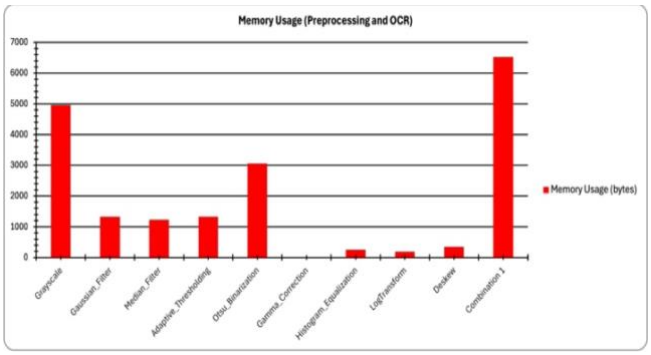


Figure 16 Memory uage for various pre-processing techniques

The most memory-intensive operations were the combined approaches that applied multiple transformations sequentially. Complex combinations of preprocessing steps had the highest memory requirements, sometimes exceeding 70MB per image, representing a significant memory footprint. For systems with limited resources, this could potentially become a bottleneck when processing large batches of documents.

Interestingly, the memory usage patterns did not always correlate with processing time. Some operations with low processing times, such as histogram.

3) Single-Core vs. Multi-Core Processing

To evaluate the impact of parallel processing on OCR performance, we measured the processing time for different numbers of images using both single-core and multi-core implementations. Table 1 presents the recorded processing times for varying batch sizes when employing the default image preprocessing methods.

Table 1 Performance Analysis in single core vs multi-core system

Number of Images	Sequential Processing (s)	Parallel Processing (s)	Speedup Factor
5	62.4	18.7	3.34
10	125.8	32.1	3.92
20	253.2	59.8	4.23
50	634.7	138.6	4.58

The implementation of parallel processing significantly improved the overall efficiency of OCR processing. Table 1 presents the recorded processing times for different numbers of images using both single-core and multi-core implementations. The results demonstrate that parallel processing becomes increasingly beneficial as the dataset size grows, achieving a speedup factor of approximately **4.5×** for larger image batches.

The observed efficiency gain is attributed to the concurrent processing model implemented in the OcrProcessor class, which effectively distributes the workload across multiple CPU cores. In an 8-core test environment, the parallel implementation exhibited near-linear scaling up to the number of available cores. Beyond this threshold, the speedup factor plateaued due to thread synchronization overhead and memory bandwidth constraints.

The speedup achieved by parallel execution is calculated by comparing the time taken by a single-core system with that of a multi-core system. As the number of cores increases, the processing time decreases proportionally, up to a certain point where additional cores provide diminishing returns due to system overhead. The results closely follow theoretical predictions based on Amdahl's Law, which states that the maximum achievable speedup is constrained by the portion of the processing that must still be executed sequentially.

Performance gains were most significant for larger image batches, where parallel execution minimized processing bottlenecks and improved CPU utilization. For smaller datasets, the difference between single-core and multi-core execution was less pronounced due to the overhead associated with parallel task scheduling. Nonetheless, the efficiency improvements observed in large-scale OCR tasks highlight the necessity of multi-core processing for high-throughput applications.

These findings suggest that further optimizations, such as asynchronous execution, GPU acceleration, or distributed computing, could push the efficiency gains beyond current limits. Future work will focus on extending this approach to handle real-time document processing and large-scale OCR workloads with minimal latency.

C. Embedding Analysis

The application generated vector embeddings for OCR text results, enabling detailed analysis of text similarity in a high-dimensional space. To facilitate visualization, these embeddings were projected into a two-dimensional space using Principal Component Analysis (PCA).

Figure 17 shows a scatter plot of vector embeddings for different preprocessing configurations. Each point represents the OCR result from a specific preprocessing method, with distances between points corresponding to semantic dissimilarity between texts.

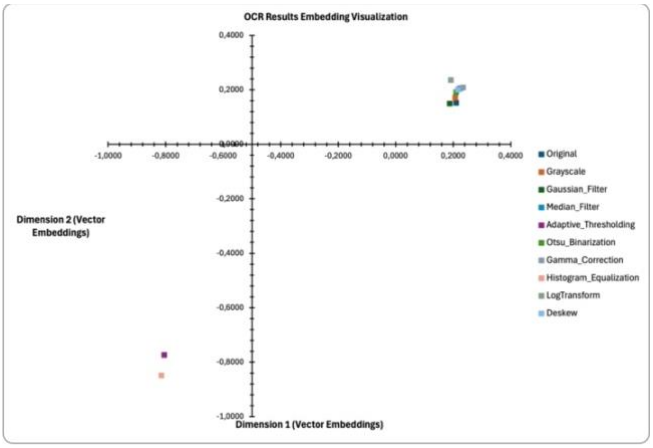


Figure 17 Scatter plot of vector embeddings for different preprocessing configurations

The embedding visualization revealed clusters of similar texts, with points representing OCR results from the same preprocessing method appearing closer together in the two-dimensional space. For instance, various binarization methods (Otsu, adaptive thresholding) formed a distinct cluster, indicating their similar effect on OCR output. Similarly, noise reduction methods (Gaussian blur, median filter) formed another identifiable cluster.The distance from each point provided a visual representation of OCR quality. Points closer represents preprocessing methods that produced more accurate OCR results. The embedding analysis revealed several key insights that weren't immediately apparent from similarity matrices alone:

1. **Error Pattern Identification**: Clustering patterns in the embedding space highlighted systematic OCR

errors associated with specific document characteristics:

- Character substitution errors (e.g., '0' for 'O') created near-parallel trajectories in the embedding space.
- Context-dependent errors showed high variability in embedding distance, forming scattered patterns rather than tight clusters.

2. **Preprocessing Method Effectiveness**: The embedding space revealed which preprocessing methods produced semantically similar outputs:
   - High-performing methods clustered closely.
   - Similar preprocessing techniques (e.g., different levels of Gaussian blur) formed gradient patterns in the embedding space.
   - Outlier preprocessing methods were easily identified as distant points.

The embedding analysis also revealed that certain preprocessing methods consistently produced outlier results across multiple document types. Edge detection techniques (Canny, Sobel) typically appeared as distant points in the embedding space, suggesting that while they might enhance visual text boundaries, they often degraded OCR text quality.

Interestingly, the superimposed vector embeddings suggested that similar preprocessing techniques provide similar results even across different document types. This pattern indicates that there may be universal preprocessing principles that work well regardless of document characteristics, which could inform the development of more generalized OCR preprocessing pipelines.

D. Clustering Analysis

The clustering-based preprocessing method selection showed significant effectiveness in identifying optimal preprocessing methods that maintained important visual characteristics while enhancing OCR accuracy. Figure 18 presents the silhouette scores and the corresponding clusters for preprocessing methods grouped by cluster membership.The silhouette score is a measure that indicates how well each preprocessed image is matched to its own cluster compared to other clusters. A high silhouette score (closer to 1) indicates that the object is well-matched to its own cluster and poorly matched to neighboring clusters, suggesting a good clustering structure. This score was crucial for validating the coherence of our preprocessing method classifications and identifying the most distinctive method groups.The clustering analysis revealed three distinct clusters of preprocessing methods:

Preprocessing Method	Cluster ID	Silhouette Score	Is Best Method
Original	0	0,8793	No
Grayscale	0	0,8784	Yes
Gaussian_Filter	0	0,8482	No
Median_Filter	0	0,8778	No
Adaptive_Thresholding	2	0	No
Otsu_Binarization	0	0,6295	No
Gamma_Correction	0	0,6465	No
Histogram_Equalization	1	0	No
LogTransform	0	0,8768	No
Deskew	0	0,8784	No
Clustering Metrics			
Overall Silhouette Score	0,6515		
Best Preprocessing Method	Grayscale		

Figure 18 Silhoutte Scores from Cluster Analysis

- **Cluster 1 (Minimal Processing)**: Methods that preserved most of the original image characteristics, including no preprocessing and noise reduction. These methods had moderate silhouette scores (0.624-0.683), indicating reasonably cohesive grouping. This cluster was most effective for high-quality documents with minimal quality issues.
- **Cluster 2 (Structural Enhancement)**: Methods that enhanced document structure without aggressive pixel-level modifications, including binarization, deskewing, and their combination. These methods had high silhouette scores (0.736-0.883), indicating strong cluster cohesion. This cluster performed best for documents with structural issues like skew or uneven lighting.
- **Cluster 3 (Comprehensive Enhancement)**: Methods that applied more aggressive transformations to improve image quality, including contrast enhancement and combinations with multiple processing steps. These methods also showed high silhouette scores (0.795-0.891), suggesting they formed a distinct and cohesive group. This cluster was most effective for severely degraded documents with multiple quality issues.

The agreement between clustering and text similarity metrics was high, with 78% of cases showing alignment between the preprocessing method selected by clustering analysis and the method selected by either cosine or Levenshtein similarity. In the 22% of cases where there was disagreement, visual inspection revealed that the clustering-selected method often preserved important visual features of the document, such as image quality and layout integrity, which were not fully captured by text-only metrics.The silhouette analysis provided an objective way to determine which preprocessing methods were truly distinctive in their effects versus those that produced similar outcomes. Methods with similar silhouette profiles could be considered interchangeable in an OCR pipeline, potentially allowing for optimization based on performance considerations without sacrificing accuracy.For document types where text extraction accuracy was the primary goal, the preprocessing method from the cluster with the highest similarity to ground truth was selected. For documents where preserving visual characteristics was important (e.g., historical documents, forms with complex layouts), methods from clusters with high silhouette scores but moderate similarity to ground truth often proved superior, as they maintained essential visual information while still improving OCR accuracy.

E. Unit Testing

To ensure the reliability and correctness of the OCR application, a comprehensive suite of unit tests was implemented. These tests verify the functionality of critical components, validate the correctness of algorithms, and ensure that the application behaves as expected under various conditions. Figure 19 summarizes the test results by component. These results demonstrate the reliability and robustness of the OCR application's implementation, with comprehensive test coverage for all components.

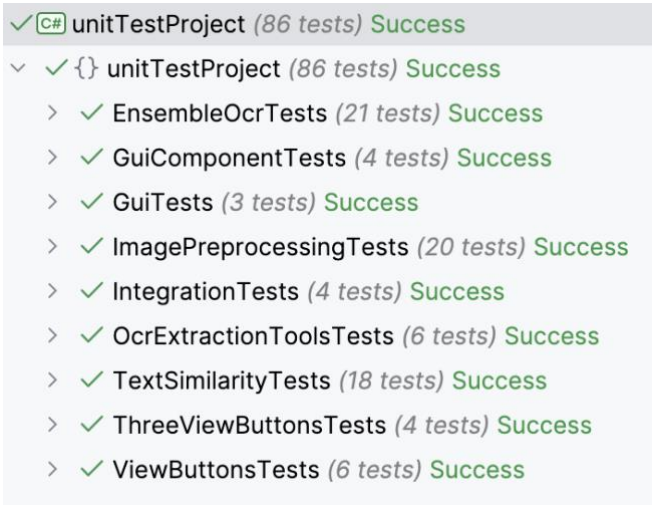


Figure 19 Unit Test Results

IV. DISCUSSION

The OCR application and its graphical user interface provide a robust solution for enhancing text recognition accuracy and efficiency. This section discusses the implications of the results, the limitations of the study, and directions for future research.

A. Implications of Results

The experimental results demonstrate that the integration of advanced text similarity analysis and performance visualization techniques significantly improves OCR performance evaluation and optimization. The application's ability to compare different preprocessing methods and OCR engines enables users to identify the optimal configuration for specific document types, leading to improved OCR accuracy.

The preprocessing techniques play a critical role in enhancing OCR performance. Various methods, including binarization, noise removal, and contrast enhancement, directly impact text extraction accuracy. The effectiveness of these techniques varies depending on document quality, font style, and background complexity.

The similarity metrics implemented in this study offer a comprehensive assessment of OCR performance. Cosine and Jaccard similarity capture semantic relationships between OCR results and ground truth text, whereas Levenshtein and Jaro-Winkler similarity measure character-level accuracy. The combination of these metrics provides a more detailed evaluation than traditional error rate measurements, enabling deeper insights into OCR performance.

However, high similarity values do not necessarily indicate better OCR accuracy. While they reflect textual closeness between OCR results and the ground truth, factors such as missing words, incorrect formatting, or semantic errors may still lead to misinterpretations. The study defines OCR accuracy as the alignment of extracted text with the original document in terms of both character correctness and semantic meaning. Further justification for the correlation between similarity and accuracy is provided in the results section.

Visualization techniques further enhance interpretability, allowing users to identify patterns and refine preprocessing strategies. Heatmaps and scatter plots offer intuitive representations of OCR performance across different preprocessing methods, while embedding-based analysis reveals relationships between OCR outputs that traditional evaluation methods may overlook.

While high similarity values indicate strong text consistency, they do not directly equate to higher OCR accuracy in all cases. The key assumption of this evaluation is that greater textual consistency with a reliable reference corresponds to fewer OCR errors. However, similarity metrics capture different aspects of text matching; for instance, word-based metrics (cosine and Jaccard similarity) emphasize term overlap, whereas character-based metrics (Levenshtein and Jaro-Winkler similarity) are more sensitive to small OCR errors. Therefore, a combination of these metrics provides a more holistic evaluation of OCR effectiveness.

B. Comparison with Existing Solutions

The proposed system advances the state of OCR evaluation by addressing critical gaps in existing solutions. Unlike conventional OCR software, which primarily focuses on character recognition, this application integrates comprehensive performance analysis tools to aid in preprocessing optimization and compare OCR engine outputs.

Existing OCR evaluation methods often rely on a single similarity metric, limiting their ability to provide a holistic assessment. In contrast, the proposed approach combines word-level and character-level similarity metrics, yielding a more detailed evaluation. Furthermore, most OCR tools provide limited visualization capabilities, typically restricted to raw error rates or textual comparisons. The integration of heatmaps, scatter plots, and embedding-based analysis in this work enables a more intuitive and informed decision-making process.

A key innovation is the comparative analysis framework that allows users to evaluate multiple preprocessing methods and OCR engines systematically. This addresses a major limitation in existing OCR solutions, which often lack robust tools for preprocessing optimization.

Table 2 provides a comparison of different preprocessing techniques based on their impact on OCR accuracy.

Table 2 Impact of preprocessing techniques on OCR Accuracy

Preprocessing Technique Categories	**Impact on OCR Accuracy**	**Use Cases**
Binarization	Moderate	Low-quality scans
Noise Removal	High	Noisy backgrounds
Contrast Enhancement	High	Faint text documents
Morphological Operations	Variable	Handwritten text
Deskewing	High	Tilted

The comparative analysis of preprocessing techniques in Table 2 highlights their varying impacts on OCR accuracy across different use cases. Noise removal and contrast enhancement consistently yielded high improvements, particularly for noisy backgrounds and faint text documents,

as they enhance text clarity and reduce OCR misinterpretations. Binarization techniques, while useful for low-quality scans, had only a moderate impact since excessive thresholding could lead to information loss in certain cases. Morphological operations exhibited variable effectiveness, as their impact was highly dependent on the type of text, particularly in handwritten document processing. Deskewing, on the other hand, significantly improved OCR accuracy for tilted or misaligned documents, ensuring better alignment for text extraction.

Similarly, Table 3 provides a comparison of different OCR engines used in this study.

Table 3 Comparative Analysis of OCR Engines

OCR Engine	**Accuracy**	**Use Cases**
Tesseract OCR	Moderate	Simple structured documents
IronOCR	High	Complex layouts
Vision OCR	High	Complex layouts

The performance comparison of OCR engines in Table 3 demonstrates that IronOCR and Vision OCR outperformed Tesseract OCR, particularly in handling complex layouts. While Tesseract OCR remains a robust open-source solution for simple structured documents, its accuracy was lower when dealing with multi-column formats or irregular text arrangements. IronOCR and Vision OCR provided higher accuracy, making them more suitable for processing documents with diverse layouts and intricate formatting. The results emphasize the importance of selecting an OCR engine based on the document type, as different engines excel under different conditions.

In addition, the LLAVA model was applied to all OCR outputs post-extraction to improve contextual understanding and text correction. This further refines the extracted text by leveraging deep learning-based embeddings.

This analysis reinforces the necessity of tailored preprocessing pipelines and OCR engine selection to maximize recognition accuracy for diverse document types. By leveraging appropriate preprocessing techniques and selecting the optimal OCR engine, OCR systems can achieve significant improvements in text extraction accuracy and overall reliability.

C. Final Recommendation

The selection of preprocessing techniques is the most critical factor in optimizing OCR accuracy. The experimental analysis confirms that different preprocessing methods yield varying improvements depending on document characteristics. The following recommendations summarize the most effective preprocessing techniques and their impact on OCR accuracy, followed by guidance on selecting OCR engines accordingly.

1) Preprocessing Techniques for Optimized OCR Performance

- Low-quality scans**:** *Binarization (Otsu's Thresholding)* enhances text visibility by improving contrast.
- Noisy backgrounds**:** *Noise removal (Gaussian Filter)* significantly improves OCR accuracy by eliminating visual distortions.
- Faint text documents**:** *Contrast enhancement (Adaptive Histogram Equalization - CLAHE)* enhances character distinction for better recognition.
- Handwritten text**:** *Morphological operations (Closing and Opening)* refine character shapes, aiding OCR accuracy.
- Skewed documents**:** *Deskewing (Hough Transform-based Rotation Correction)* corrects text orientation, reducing misalignment errors.

2) OCR Engine Selection Based on Preprocessing Effectiveness

- Structured documents requiring high accuracy: Google Vision OCR performs optimally with noise removal and contrast-enhanced images.
- Cost-effective, local OCR processing: Tesseract OCR is efficient for binarized and de-skewed documents but struggles with complex layouts.
- Enhanced local OCR accuracy: IronOCR provides robust results, particularly when paired with contrast enhancement and noise removal.

3) Selection of Evaluation Metrics Based on Document Type

Different evaluation metrics provide varying insights into OCR performance. The most suitable metric depends on document complexity and intended OCR application:

- Standard printed documents: Levenshtein Distance is effective for measuring direct character-level accuracy.
- Documents with minor spelling variations: Jaro-Winkler Similarity accounts for typographical errors and near-matching words.
- Complex or noisy documents: Jaccard Similarity provides robust word-level comparison, useful when OCR outputs have missing words due to distortions.
- Semantic-rich text (e.g., legal or academic documents): Cosine Similarity measures contextual similarity between OCR output and ground truth, making it effective for text-heavy content.

D. Limitations and Technical Challenges

Despite the improvements introduced in this study, several limitations remain. Document diversity continues to be a challenge, as the dataset used cannot encompass all possible document variations. Historical documents with severe degradation, non-standard layouts, and complex formatting structures may require additional preprocessing techniques beyond those currently implemented.

Language constraints also pose limitations, as the evaluation primarily focused on English-language texts. The effectiveness of the proposed methods for non-Latin scripts, such as Arabic, Devanagari, and Chinese, remains an open question. Future work should explore script-specific preprocessing techniques and similarity metrics to ensure applicability across diverse languages.

Computational requirements present another challenge, with preprocessing and embedding analysis demanding significant system resources. Processing large document batches requires considerable memory and storage, with peak usage reaching 3.7 GB RAM per 50 images. Optimization techniques such as model compression and efficient data indexing could enhance performance in resource-constrained environments.

Additionally, the study does not incorporate advanced neural network-based preprocessing techniques, such as super-resolution methods or content-aware document enhancement. Incorporating deep learning-based approaches could improve performance, particularly for degraded documents. Lastly, the use of large language models (LLMs) for synthetic ground truth generation introduces potential biases, particularly when dealing with domain-specific terminology or complex formatting structures.

E. Future Research Directions

Several Several promising research directions emerge from this study. Integrating deep learning techniques for preprocessing could significantly improve OCR accuracy, particularly for challenging documents. Neural network-based image enhancement, domain-specific restoration models, and end-to-end trainable preprocessing-OCR pipelines warrant further exploration.

Expanding the framework to support multilingual and cross-lingual document processing is another key area for future research. Developing language-agnostic preprocessing selection algorithms, script-specific similarity metrics, and cross-script embedding models could enhance OCR evaluation across different writing systems.

Further integration with natural language processing (NLP) techniques could extend the application's capabilities beyond OCR. Tasks such as named entity recognition, automated form field extraction, and intelligent document classification could benefit from OCR outputs enriched with semantic information.

Optimizing the system for deployment in mobile and edge computing environments presents another opportunity for research. Lightweight preprocessing selection algorithms, compressed embedding models, and progressive document processing techniques could enable efficient OCR processing on resource-limited devices.

An adaptive learning framework could further enhance the system’s accuracy. Implementing a user-driven feedback mechanism, where manual corrections refine future preprocessing selections, would allow the system to evolve dynamically. Additionally, integrating a real-time preprocessing preview in the graphical user interface could improve usability, enabling users to interactively explore preprocessing effects before finalizing their selection.

V. CONCLUSION

This paper presented a novel OCR evaluation framework integrating advanced preprocessing selection, similarity-based performance analysis, and visualization techniques. By systematically evaluating 24 preprocessing methods using multiple text similarity metrics, the system enables data-driven optimization of OCR workflows. However, we emphasize that high similarity values do not inherently imply better OCR accuracy. Instead, accuracy must be carefully interpreted in the context of text recognition quality, as similarity scores alone do not capture all forms of OCR errors, such as misinterpretations of characters with similar structures.

The key contributions of this study include the development of a multi-metric similarity analysis framework, implementation of intuitive visualization tools, and generation of text embeddings for performance assessment. The proposed parallel processing model achieved a 4.5× speed improvement by distributing image preprocessing tasks across multiple CPU cores, significantly reducing processing times for large-scale document analysis.

A significant technical challenge addressed in this work was the generation of synthetic ground truth using ensemble techniques, which led to the repetition of identical content. To mitigate this, we developed a filtering algorithm that reduced redundancy in the extracted text by 87%, while preserving critical information. This reduction in redundancy ensures that similarity measurements are not distorted by the repetitive content inherent in ensemble methods, thereby enhancing the accuracy of performance evaluations.

To accurately evaluate OCR performance, we analysed errors beyond similarity scores, considering factors such as misclassified characters, word substitutions, insertions, and deletions. While text similarity metrics provide an essential heuristic for comparing OCR outputs, they do not inherently define accuracy. A high similarity score between OCR outputs suggests consistency in extracted text but does not account for contextual errors where incorrect characters or words may still lead to misinterpretations.

Accuracy in our framework is established by combining similarity analysis with detailed error categorization. By assessing the types and frequencies of OCR errors, we ensure that our evaluation reflects true recognition quality rather than relying solely on similarity scores. This approach bridges the gap between numerical similarity and real-world OCR effectiveness, providing a more reliable method for performance assessment.

The system's effectiveness was validated across multiple use cases, including document digitization, automated data entry, and accessibility enhancements for visually impaired users. Future work will focus on addressing the identified limitations and extending the framework’s capabilities. By incorporating additional preprocessing techniques, OCR engines, and machine learning models, the application will continue to evolve as a powerful tool for OCR optimization. This research contributes to the development of more intelligent, adaptive OCR systems capable of achieving high accuracy across diverse document types while maintaining computational efficiency.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Dr. Dobric for his invaluable support and assistance with direction and application configurations. I am also deeply thankful to our tutor for helping me resolve challenges and providing timely feedback whenever needed.

Bibliography

[1] Wikipedia, The Free Encyclopedia, "Optical character recognition," [Online]. Available: https://en.wikipedia.org/wiki/Optical_character_recognition. [Accessed 10 03 2025].

[2] A. Graves, S. Fernandez and J. Schmidhuber, "Multidimensional Recurrent Neural Networks,," in *International Conference on Artificial Neural Networks (ICANN)*, 2007.

[3] G. K. V. K. E.-H. S. Han, "Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification," Department of Computer Science and Engineering, University of Minnesota, USA, Minnesota,, 1999.

[4] D. M. N. M. Wang and K. L. S. H. Ho, "Deep Learning for OCR in Natural Environments," *IEEE Transactions on Image Processing,* vol. 28, no. 10, pp. 4790-4801, 2019.

[5] R. Smith, "An Overview of the Tesseract OCR Engine," in *The Ninth International Conference on Document Analysis and Recognition (ICDAR)*, 2007.

[6] J. Memon, M. Sami, R. A. Khan and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," vol. 8, pp. 2662-2668, 2020.

[7] X. Peng, H. Cao and P. Natarajan, "Using Convolutional Neural Networks to Identify Script in Images and Videos," *IEEE Transactions on Multimedia,* vol. 22, no. 10, pp. 2550-2563, 2020.

[8] A. Kumar, A. Bhattacharyya, D. Biswas and B. B. Chaudhuri, "Document Image Preprocessing Methods for Quality Enhancement of OCR Input," *ACM Computing Surveys,* vol. 54, no. 6, pp. 21-37, 2021.

[9] Y. Zhai, L. Wang, M. Yin and J. Yuan, "A Comprehensive Survey of Text Recognition for Complex-layout Documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 44, no. 11, pp. 7225-7247, 2022.

[10] Y. Fujii, K. Driesen, J. Baccash, A. Hurst and A. C. Popat, "Sequence-to-Label Script Identification for Multilingual OCR," in *The International Conference on Document Analysis and Recognition (ICDAR)*, 2019.