Information Technology Course
Module Software Engineering
by Damir Dobric / Andreas Pech

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Creating Text from Images with OCR API

Mithila Prabhu
mithila.prabhu@stud.fra-uas.de

Khushal Singh
khushal.singh@stud.fra-uas.de

Taibaz Pathan
taibaz.pathan@stud.fra-uas.de

***Abstract*—Effective image preprocessing plays a crucial role in enhancing the accuracy and efficiency of Optical Character Recognition (OCR) systems by improving text clarity and reducing background noise. This paper presents a systematic evaluation of five essential preprocessing techniques grayscale conversion, adaptive thresholding, global thresholding, saturation adjustment and deskewing implemented using the Tesseract SDK. To facilitate this study, a C# console application is developed to apply these preprocessing methods and assess their impact on OCR performance. The evaluation is conducted based on three key factors: cosine similarity, time taken by each preprocessing technique and memory usage of each preprocessing technique. Cosine similarity metrics are employed to compare the extracted text against each preprocessing model, providing a quantitative measure of recognition accuracy. Additionally, time tracking is used to analyze processing speed, while memory usage is monitored to assess the computational overhead introduced by each technique. By systematically analyzing the trade-offs between accuracy and performance, this study offers valuable insights into optimizing preprocessing pipelines for real-world OCR applications.**

***Keywords*—Optical Character Recognition (OCR), image preprocessing, Tesseract SDK, grayscale conversion, adaptive thresholding, global thresholding, saturation adjustment, deskewing, cosine similarity, processing time, memory usage.**

I. Introduction

Optical Character Recognition (OCR) is a widely utilized technology that facilitates the conversion of different types of documents, such as scanned paper documents, PDF files, and images, into editable and searchable data. OCR has a broad range of applications in various domains, including banking, healthcare, education, and legal documentation. With the increasing digital transformation of industries, OCR technology plays a critical role in automating data extraction and reducing manual effort in document processing. Traditional OCR systems relied on template matching and feature extraction techniques to identify text within an image. However, these methods often struggled with variations in font styles, sizes, and distortions caused by noise and poor image quality. In response, modern OCR systems have adopted machine learning and deep learning approaches to improve accuracy and adaptability. One of the most widely used OCR engines, Tesseract, has been developed by Google and is known for its robust text recognition capabilities across multiple languages. Tesseract OCR is an open-source engine that utilizes a two-step process for text extraction: first, it identifies connected components to detect potential characters, and second, it applies a machine learning-based approach for text recognition. Tesseract has undergone significant improvements with the introduction of LSTM (Long Short-Term Memory) networks in its newer versions, enhancing accuracy for both printed and handwritten text [1]. The engine supports multiple languages and character sets, making it highly versatile for diverse OCR applications. Despite the advancements in OCR technology, several challenges remain. Factors such as low contrast, blurred text, varying illumination conditions, and complex backgrounds negatively impact OCR accuracy. To address these challenges, pre-processing techniques such as binarization, noise reduction, edge detection, and adaptive thresholding are commonly applied before feeding images into an OCR engine. Recent studies have demonstrated that integrating these pre-processing steps significantly enhances text extraction performance, particularly in noisy or degraded images [2]. In addition to traditional OCR processing, text similarity plays a crucial role in post-processing and validation. One effective method for text comparison is the Cosine Similarity Matrix, which measures the similarity between text embeddings by calculating the cosine of the angle between two vectors. This approach is widely used in document retrieval and text clustering tasks, helping improve OCR accuracy by filtering out incorrect predictions and matching extracted text against known references [3]. Furthermore, recent advancements in deep learning have introduced the concept of embedding generation for text representation. OpenAI's text-embedding-ada-002 model is a state-of-the-art solution for generating high-quality embeddings for textual data. These embeddings are numerical representations of text that capture semantic meanings, enabling improved text classification, clustering, and search capabilities. By integrating embeddings from the text-embedding-ada-002 model, OCR systems can enhance text recognition accuracy by leveraging contextual understanding and similarity analysis [4]. This paper aims to improve OCR accuracy by integrating advanced image preprocessing techniques with Tesseract OCR while incorporating text similarity analysis and embedding generation. The primary objectives of this research include: (1) enhancing text visibility using various image enhancement methods, (2) comparing different preprocessing pipelines to determine the most effective approach, and (3) evaluating the system's performance based on accuracy, word error rate, and processing time. The proposed methodology is validated using

a dataset comprising diverse text images under different conditions. Experimental results demonstrate that pre-processing significantly improves OCR performance, making the system more reliable for real-world applications. The rest of the paper is structured as follows: Section 2 presents a literature review on OCR technologies and pre-processing techniques. Section 3 describes the methodology, including the proposed image processing and OCR pipeline. Section 4 discusses the experimental results and findings. Finally, Section 5 provides conclusions and directions for future research.

II. Literature Review

Optical Character Recognition (OCR) research has evolved significantly over time, transitioning from rule-based methods to machine learning and deep learning-based approaches. Early OCR systems primarily relied on character template matching, which was effective for fixed-font printed text but faced difficulties with handwritten text and distortions introduced by noise and poor image quality [5].

Recent advancements in deep learning, especially Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have considerably improved the accuracy of OCR systems. Studies have shown that integrating CNN-based feature extraction with RNNs, such as Long Short-Term Memory (LSTM) networks, enhances OCR capabilities by capturing contextual dependencies between characters [6]. This combination has demonstrated substantial improvements in recognizing complex text structures and enhancing accuracy in various contexts.

Pre-processing techniques have been explored extensively to improve OCR performance, particularly for challenging conditions such as noisy or degraded images. Traditional methods, such as grayscale conversion, adaptive and global thresholding, and contrast enhancement, are employed to enhance text visibility. Recent advancements have also focused on techniques like deskewing and saturation adjustment, which help correct image orientation and improve clarity. Zhang demonstrated that combining grayscale conversion with adaptive thresholding significantly improved OCR accuracy, especially in scanned documents with uneven lighting conditions [7].

Another key area of interest in OCR research is computational efficiency. Since real-world applications often require large-scale processing, researchers have worked on optimizing memory usage and reducing execution time. Patel compared different preprocessing pipelines and concluded that a combination of grayscale conversion and adaptive thresholding provided the best balance between accuracy and computational efficiency, making it suitable for large-scale OCR applications [8].

Cosine Similarity Matrix (CSM) has found wide application in text-processing tasks, including OCR post-processing. By using text similarity metrics, CSM helps reduce recognition errors by cross-verifying extracted text with reference datasets. This method is particularly effective in resolving issues where OCR misclassifies similar-looking characters, such as distinguishing between '0' and 'O' or '1' and 'l' [9].

The introduction of transformer-based models has further enhanced OCR applications. OpenAI's text-embedding-ada-002 model, for example, generates high-dimensional embeddings that capture semantic relationships between words. These embeddings significantly improve OCR post-processing tasks such as keyword search, text clustering, and document retrieval. Studies have shown that combining OCR with embedding-based text analysis not only improves document classification accuracy but also enhances retrieval efficiency [10].

Given these developments, the research presented in this paper builds upon prior work by integrating advanced image pre-processing methods, Tesseract OCR, cosine similarity for text validation, and embedding-based text analysis while also evaluating computational efficiency. The primary goal is to create a robust OCR pipeline that enhances text extraction accuracy while minimizing both time and memory consumption.

III. Methodology

This section describes the methodology used in preprocessing images, extracting text, and evaluating the performance of different image processing models for OCR. The system consists of three major components:

- **Image Preprocessing** – Various transformations such as grayscale conversion, thresholding, saturation adjustment, deskewing, and shifting were applied to improve OCR accuracy.
- **OCR Processing** – The Terrasect SDK was used to extract text from preprocessed images.
- **Performance Evaluation** – The extracted text was analyzed based on cosine similarity, processing time, and memory usage to determine the best preprocessing model.

A. Image Preprocessing Techniques

Image preprocessing is crucial for enhancing text clarity before OCR. This study implements multiple transformations, each designed to address specific challenges such as noise, low contrast, skewed text, and misalignment.

1) Grayscale Conversion: Grayscale conversion is a crucial preprocessing step in Optical Character Recognition (OCR) that simplifies image complexity while preserving text integrity. By transforming a color image into a single-channel representation, grayscale conversion enhances contrast and prepares the image for further processing techniques, such as thresholding, which improves text clarity.

In this project, grayscale conversion is implemented using the ConvertToGrayscale class, which processes each pixel individually. The transformation follows a weighted luminance formula that prioritizes different color components based on their contribution to perceived brightness:

$$Gray = 0.3R + 0.59G + 0.11B$$

This formula reflects human visual perception, where green has the highest influence on brightness, followed by red and blue. The algorithm iterates through each pixel of the input

image, applies this formula, and generates a new single-channel grayscale image optimized for text extraction.

To enhance efficiency and maintain image quality, the grayscale conversion function is optimized using the SixLabors.ImageSharp library. The Apply method ensures precise pixel-by-pixel processing using the following approach:

```
// Get the original pixel color at (x, y).
Rgba32 pixelColor = original[x, y];

// Compute grayscale intensity using the
    luminance formula.
byte gray = (byte)(0.3 * pixelColor.R + 0.59 *
    pixelColor.G + 0.11 * pixelColor.B);

// Assign the grayscale value to the new image.
grayscaleImage[x, y] = new L8(gray);
```

Grayscale conversion significantly reduces unnecessary color variations, making text stand out against the background. By providing a cleaner input for subsequent preprocessing techniques, this step ensures improved OCR accuracy and enhances the overall robustness of text recognition applications.

2) Global Thresholding: Global thresholding applies a fixed threshold value to the entire image, meaning pixels above the threshold are converted to white (255), while those below are turned black (0). This method is effective for images with uniform lighting conditions, but it may struggle with images that have uneven illumination or shadows.

```
byte value = grayscale >= _threshold ? (byte)255
    : (byte)0;
thresholdedImage[x, y] = new Rgba32(value, value,
     value, 255);
```

A commonly used threshold value falls within the range of 80 to 150, depending on the contrast and clarity of the input image.

3) Adaptive Thresholding: Unlike global thresholding, adaptive thresholding determines the binarization threshold dynamically based on local pixel intensity variations within different regions of the image. This approach is highly effective when dealing with images that have non-uniform lighting conditions, shadows, or varying brightness levels.

The threshold for each region is computed using the local mean intensity:

```
byte localThreshold = (byte)((sum / blockArea) -
    _c);
```

Adaptive thresholding enhances OCR accuracy by preserving fine details in the text while minimizing noise interference. This technique is particularly useful for scanned documents, handwritten text, and images captured in challenging lighting conditions. By selecting the appropriate thresholding method based on the input image characteristics, OCR performance can be significantly improved.

4) Saturation Adjustment: Saturation adjustment is an essential preprocessing technique that plays a significant role in enhancing the clarity of text in images, especially in cases where the text is faded, low-contrast, or noisy. By modifying the intensity of the colors in an image, saturation adjustment increases the contrast between the text and its background. This improvement in contrast is particularly beneficial for Optical Character Recognition (OCR) accuracy, as it helps make the text more distinguishable from the background. Such enhancement is crucial before other preprocessing steps like binarization and thresholding, as it ensures better text segmentation, leading to improved OCR performance.

In the context of this project, saturation adjustment is implemented using the SaturationAdjustment class. The core functionality of this class lies in its Apply method, which processes the image by adjusting the color intensity of each pixel based on a given saturation factor. The transformation is performed efficiently with the SixLabors.ImageSharp library, which provides tools for seamless manipulation of the image's pixels while maintaining the image structure.

The first step in the implementation involves cloning the original image. This ensures that the integrity of the original image is preserved, and any modifications are applied to a separate copy, preventing changes to the source image. A color transformation matrix is then applied, which adjusts the RGB (Red, Green, Blue) channels of each pixel based on the saturation factor. This modification alters the intensity of each color in the image, effectively changing its saturation. After applying the transformation, the new pixel values are clamped within the valid RGB range of 0 to 255. This clamping process ensures that the pixel values remain valid and prevents color distortion that may occur if the values exceed the allowed range. Finally, the alpha (transparency) channel is left unchanged, preserving the image's original transparency and overall structure.

Here is a code snippet that illustrates the process of pixel manipulation involved in saturation adjustment:

```
Rgba32 pixel = adjustedImage[x, y];

// Retrieve the original pixel color
Rgba32 pixel = adjustedImage[x, y];

// Apply the color transformation based on the
    saturation factor
float newRed = pixel.R * colorMatrixElements
    [0][0] + pixel.G * colorMatrixElements[1][0]
    + pixel.B * colorMatrixElements[2][0];
float newGreen = pixel.R * colorMatrixElements
    [0][1] + pixel.G * colorMatrixElements[1][1]
    + pixel.B * colorMatrixElements[2][1];
float newBlue = pixel.R * colorMatrixElements
    [0][2] + pixel.G * colorMatrixElements[1][2]
    + pixel.B * colorMatrixElements[2][2];

// Clamp pixel values to maintain valid RGB range
adjustedImage[x, y] = new Rgba32(
    (byte)Math.Clamp(newRed, 0, 255),
    (byte)Math.Clamp(newGreen, 0, 255),
    (byte)Math.Clamp(newBlue, 0, 255),
    pixel.A // Preserve alpha transparency
);
```

This saturation adjustment process effectively enhances the image's color contrast, making the text more visible and improving OCR accuracy. In scanned documents where the text may have faded or blended into the background, adjusting the saturation helps separate the text from the background, enabling more accurate text recognition. This preprocessing step is essential for ensuring reliable OCR results, making it a critical component in the image preprocessing pipeline for text extraction tasks.

5) Deskewing (Skew Correction): Skewed text in scanned or captured images can negatively impact OCR accuracy by distorting character alignment, making text recognition challenging. Deskewing, also known as skew correction, is a preprocessing step that detects and corrects text orientation to ensure proper alignment before OCR processing.

The deskewing process begins with edge detection, where the system identifies prominent edges in the image, particularly focusing on text lines. Once the edges are detected, the skew angle is calculated to determine the extent of misalignment.

```
double skewAngle = CalculateSkewAngle(DetectEdges
    (inputImage));
```

After determining the skew angle, the image is rotated in the opposite direction to align the text horizontally, ensuring that the OCR system can accurately interpret characters without distortion.

```
return RotateImage(inputImage, -skewAngle);
```

By aligning text properly, deskewing significantly enhances OCR performance, improving recognition accuracy, readability, and overall efficiency in text extraction. Integrating this step into the preprocessing pipeline ensures more reliable OCR results.

B. OCR Processing Using the Terrasect SDK

After preprocessing the images, text extraction is performed using the Tesseract SDK. This open-source OCR engine is integrated into a C# console application to automate text recognition and support batch processing. The implementation ensures efficient and scalable text extraction from processed images.

Within the application, a dedicated function handles text extraction from images, leveraging the capabilities of Tesseract. The function processes the image, applies the specified OCR model, and stores the extracted text for further analysis.

```
string extractedText = TesseractProcessor.
    ExtractTextFromImage(processedImage,
    modelName, createdFilePath, fileWriter);
```

Once the text is extracted, it is stored for evaluation. The results from different preprocessing techniques are compared to determine the most effective method in enhancing OCR accuracy. This approach ensures that text recognition is optimized, leading to improved reliability in real-world applications.

C. Performance Evaluation Metrics

To evaluate the effectiveness of each preprocessing method, three key performance indicators were analyzed:

The first metric, cosine similarity, measures how closely the extracted text from one preprocessing technique matches other preprocessing technique. A higher similarity score indicates better text extraction accuracy, making this metric crucial for assessing OCR performance.

The second metric, Processing Time, tracks the duration taken to process an image through each preprocessing method. A high-precision timer records the execution time from start to completion, ensuring a fair comparison between different approaches. Faster execution times indicate more efficient preprocessing, which is especially important for large-scale OCR applications where processing speed impacts overall system performance.

The third metric, Memory Consumption, monitors the amount of system memory used during image preprocessing. This metric is essential for evaluating computational efficiency, particularly for handling high-resolution images or large datasets. Lower memory usage is preferred, as it enables processing on resource-constrained environments without excessive system overhead.

D. Comparative Analysis and Model Selection

A weighted scoring approach is used to determine the optimal preprocessing model by balancing accuracy, processing speed, and memory efficiency. The cosine similarity metric contributes the highest weight, as it directly affects OCR output quality. Processing speed is also prioritized, as faster methods are preferable in real-time or large-scale applications. Lastly, memory usage is considered to ensure that the chosen method remains computationally efficient.

By assigning appropriate weightages to each criterion, the overall performance of different preprocessing techniques can be ranked, allowing for an informed selection of the best approach. This ranking system ensures an optimal trade-off between accuracy, efficiency, and computational overhead, leading to improved OCR results while maintaining a balance between performance and resource utilization.

IV. Implementation

The OCR pipeline integrates embedding generation and cosine similarity calculation to evaluate the performance of various preprocessing models. The implementation workflow is illustrated in Fig. 1:

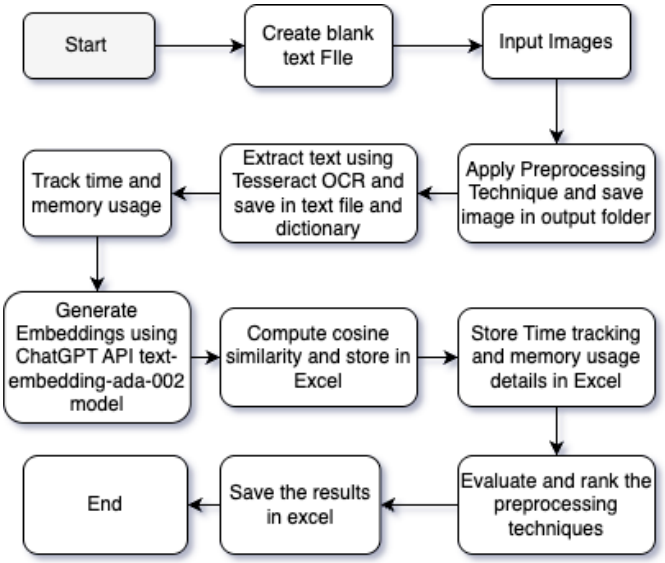


Fig. 1: Workflow Diagram

A. Initialization and Service Setup

The pipeline begins by initializing key services and loading configuration settings. Logging is enabled to capture events throughout the process. The EmbeddingGeneratorService generates embeddings for extracted text using OpenAI's API. Performance tracking is handled by the ProcessingTimeTracker and ProcessingMemoryTracker, measuring the time and memory usage during preprocessing and text extraction. Cosine similarity calculations are performed to assess the similarity between text outputs from different preprocessing models. Configuration settings, including file paths for input images, output directories, and API keys, are loaded at the start.

B. Image Preprocessing and OCR Execution

Before extracting text, images undergo several preprocessing steps to optimize them for OCR:

- **Resizing:** Ensures uniform dimensions for consistent OCR performance.
- **Grayscale Conversion:** Enhances contrast and reduces noise.
- **Adaptive Thresholding:** Dynamically adjusts threshold values for improved text visibility under varying lighting conditions.
- **Saturation Adjustments:** Applied at different levels (0.6, 0.9, 1.2, 1.6, 2.0, 2.5) to improve text clarity.
- **Deskewing:** Corrects misaligned text for better OCR accuracy.
- **Global Thresholding:** Converts images to grayscale to enhance contrast.

C. Text Extraction with Tesseract OCR

After preprocessing, text extraction is performed using the Tesseract OCR engine. The processed image is loaded into memory, converted into a PNG stream using the SaveAsPng method, and passed to Tesseract. The image is loaded as a Pix object, and Tesseract recognizes the text. The extracted text is retrieved using page.GetText() and saved for future use. To maintain organization across multiple preprocessing models, extracted text is labeled with metadata indicating the image source, processing type, or OCR model used. The labeled text is then stored using the FileWriter class.

D. Performance Tracking

The system records processing time and memory usage for each preprocessing model:

- **Processing Time:** The ProcessingTimeTracker measures the time taken for each transformation and OCR execution in milliseconds.
- **Memory Usage:** The ProcessingMemoryTracker records the memory consumed during processing using the MeasureMemoryUsage method.

E. Generating Embeddings and Cosine Similarity Calculation

After text extraction, embeddings are generated, and cosine similarity is calculated to compare results across different preprocessing models. The extracted text is converted into numerical vectors using embeddings generated via OpenAI's API. The EmbeddingGeneratorService ensures each preprocessing model—such as grayscale conversion or thresholding—produces a corresponding embedding stored as a float array.

Cosine similarity is used to measure textual similarity. The similarity score ranges from 1 (identical text) to 0 (completely dissimilar text). The CosineSimilarityCalculator computes this metric using:

$$\text{cosine similarity} = \frac{A \cdot B}{\|A\|\|B\|}$$

where $A \cdot B$ is the dot product of two vectors, and $\|A\|$ and $\|B\|$ are their magnitudes.

The similarity scores are stored in an Excel report, facilitated by the NPOI library. OpenAI embedding generation requests are sent via HttpClient, ensuring an automated process.

F. Evaluating Preprocessing Models

The extracted text and similarity results are analyzed to identify the most effective preprocessing models. Cosine similarity helps determine text consistency, while processing time and memory usage help in selecting efficient techniques. The PreprocessingModelEvaluator ranks models based on a weighted score:

- **Cosine Similarity:** 50% weight
- **Processing Time:** 30% weight
- **Memory Usage:** 20% weight

The best-performing models are identified and stored for further use.

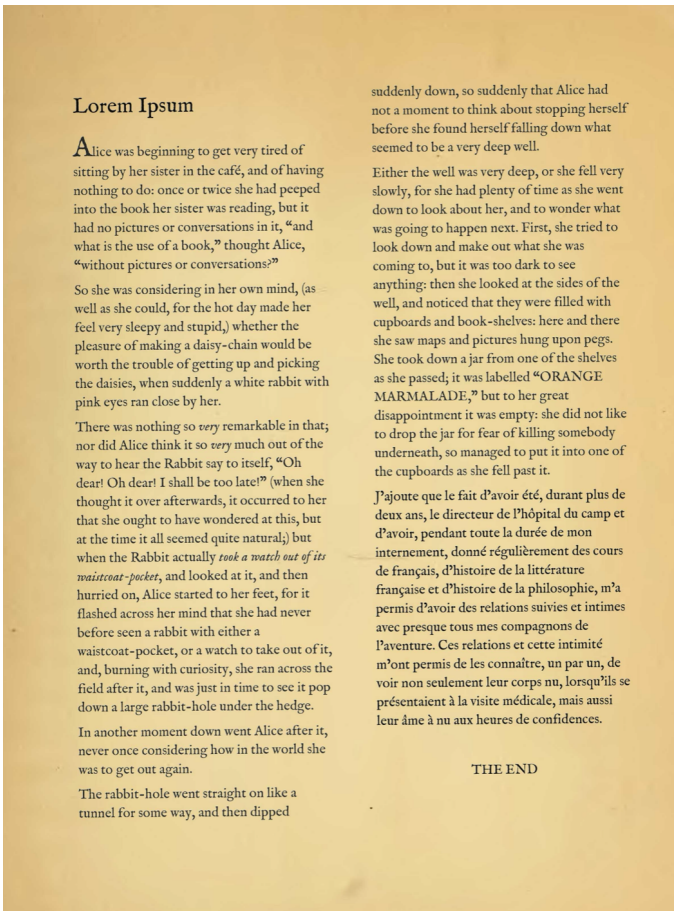
G. Output and Data Export

The extracted text, along with metadata such as file names and timestamps, is stored and exported using the NPOI library. Key operations, errors, and warnings are logged for debugging and auditing.

V. Results

1) Performance Evaluation: The effectiveness of text extraction is assessed through a structured evaluation of key performance metrics. These metrics provide insights into the efficiency and accuracy of the preprocessing techniques utilized in this study.

Impact of Preprocessing on Text Extraction: The images below illustrate the impact of different preprocessing techniques on input image text. The three variations—Global Thresholding at 80, Saturation Adjustment at 0.6, and Saturation Adjustment at 2.5, Grayscale conversion.



Lorem Ipsum

Alice was beginning to get very tired of sitting by her sister in the café, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice, "without pictures or conversations?"

So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a white rabbit with pink eyes ran close by her.

There was nothing so *very* remarkable in that; nor did Alice think it so *very* much out of the way to hear the Rabbit say to itself, "Oh dear! Oh dear! I shall be too late!" (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural;) but when the Rabbit actually *took a watch out of its waistcoat-pocket*, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and, burning with curiosity, she ran across the field after it, and was just in time to see it pop down a large rabbit-hole under the hedge.

In another moment down went Alice after it, never once considering how in the world she was to get out again.

The rabbit-hole went straight on like a tunnel for some way, and then dipped suddenly down, so suddenly that Alice had not a moment to think about stopping herself before she found herself falling down what seemed to be a very deep well.

Either the well was very deep, or she fell very slowly, for she had plenty of time as she went down to look about her, and to wonder what was going to happen next. First, she tried to look down and make out what she was coming to, but it was too dark to see anything: then she looked at the sides of the well, and noticed that they were filled with cupboards and book-shelves: here and there she saw maps and pictures hung upon pegs. She took down a jar from one of the shelves as she passed; it was labelled "ORANGE MARMALADE," but to her great disappointment it was empty: she did not like to drop the jar for fear of killing somebody underneath, so managed to put it into one of the cupboards as she fell past it.

J'ajoute que le fait d'avoir été, durant plus de deux ans, le directeur de l'hôpital du camp et d'avoir, pendant toute la durée de mon internement, donné régulièrement des cours de français, d'histoire de la littérature française et d'histoire de la philosophie, m'a permis d'avoir des relations suivies et intimes avec presque tous mes compagnons de l'aventure. Ces relations et cette intimité m'ont permis de les connaître, un par un, de voir non seulement leur corps nu, lorsqu'ils se présentaient à la visite médicale, mais aussi leur âme à nu aux heures de confidences.

THE END

Fig. 2: Input image

Grayscale conversion eliminates unnecessary color details, improving text visibility.

Lorem Ipsum

Alice was beginning to get very tired of sitting by her sister in the café, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice, "without pictures or conversations?"

So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a white rabbit with pink eyes ran close by her.

There was nothing so *very* remarkable in that; nor did Alice think it so *very* much out of the way to hear the Rabbit say to itself, "Oh dear! Oh dear! I shall be too late!" (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural;) but when the Rabbit actually *took a watch out of its waistcoat-pocket*, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and, burning with curiosity, she ran across the field after it, and was just in time to see it pop down a large rabbit-hole under the hedge.

In another moment down went Alice after it, never once considering how in the world she was to get out again.

The rabbit-hole went straight on like a tunnel for some way, and then dipped suddenly down, so suddenly that Alice had not a moment to think about stopping herself before she found herself falling down what seemed to be a very deep well.

Either the well was very deep, or she fell very slowly, for she had plenty of time as she went down to look about her, and to wonder what was going to happen next. First, she tried to look down and make out what she was coming to, but it was too dark to see anything: then she looked at the sides of the well, and noticed that they were filled with cupboards and book-shelves: here and there she saw maps and pictures hung upon pegs. She took down a jar from one of the shelves as she passed; it was labelled "ORANGE MARMALADE," but to her great disappointment it was empty: she did not like to drop the jar for fear of killing somebody underneath, so managed to put it into one of the cupboards as she fell past it.

J'ajoute que le fait d'avoir été, durant plus de deux ans, le directeur de l'hôpital du camp et d'avoir, pendant toute la durée de mon internement, donné régulièrement des cours de français, d'histoire de la littérature française et d'histoire de la philosophie, m'a permis d'avoir des relations suivies et intimes avec presque tous mes compagnons de l'aventure. Ces relations et cette intimité m'ont permis de les connaître, un par un, de voir non seulement leur corps nu, lorsqu'ils se présentaient à la visite médicale, mais aussi leur âme à nu aux heures de confidences.

THE END

Fig. 3: After converting to Grayscale

Thresholding converts images into pure black and white, aiding OCR but potentially removing finer details.

Lorem Ipsum

Alice was beginning to get very tired of sitting by her sister in the café, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice, "without pictures or conversations?"

So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a white rabbit with pink eyes ran close by her.

There was nothing so *very* remarkable in that; nor did Alice think it so *very* much out of the way to hear the Rabbit say to itself, "Oh dear! Oh dear! I shall be too late!" (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural;) but when the Rabbit actually *took a watch out of its waistcoat-pocket*, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and, burning with curiosity, she ran across the field after it, and was just in time to see it pop down a large rabbit-hole under the hedge.

In another moment down went Alice after it, never once considering how in the world she was to get out again.

The rabbit-hole went straight on like a tunnel for some way, and then dipped suddenly down, so suddenly that Alice had not a moment to think about stopping herself before she found herself falling down what seemed to be a very deep well.

Either the well was very deep, or she fell very slowly, for she had plenty of time as she went down to look about her, and to wonder what was going to happen next. First, she tried to look down and make out what she was coming to, but it was too dark to see anything: then she looked at the sides of the well, and noticed that they were filled with cupboards and book-shelves: here and there she saw maps and pictures hung upon pegs. She took down a jar from one of the shelves as she passed; it was labelled "ORANGE MARMALADE," but to her great disappointment it was empty: she did not like to drop the jar for fear of killing somebody underneath, so managed to put it into one of the cupboards as she fell past it.

J'ajoute que le fait d'avoir été, durant plus de deux ans, le directeur de l'hôpital du camp et d'avoir, pendant toute la durée de mon internement, donné régulièrement des cours de français, d'histoire de la littérature française et d'histoire de la philosophie, m'a permis d'avoir des relations suivies et intimes avec presque tous mes compagnons de l'aventure. Ces relations et cette intimité m'ont permis de les connaître, un par un, de voir non seulement leur corps nu, lorsqu'ils se présentaient à la visite médicale, mais aussi leur âme à nu aux heures de confidences.

THE END

Fig. 4: After global thresholding at 80

Lorem Ipsum

Alice was beginning to get very tired of sitting by her sister in the café, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice, "without pictures or conversations?"

So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a white rabbit with pink eyes ran close by her.

There was nothing so *very* remarkable in that; nor did Alice think it so *very* much out of the way to hear the Rabbit say to itself, "Oh dear! Oh dear! I shall be too late!" (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural;) but when the Rabbit actually *took a watch out of its waistcoat-pocket*, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and, burning with curiosity, she ran across the field after it, and was just in time to see it pop down a large rabbit-hole under the hedge.

In another moment down went Alice after it, never once considering how in the world she was to get out again.

The rabbit-hole went straight on like a tunnel for some way, and then dipped suddenly down, so suddenly that Alice had not a moment to think about stopping herself before she found herself falling down what seemed to be a very deep well.

Either the well was very deep, or she fell very slowly, for she had plenty of time as she went down to look about her, and to wonder what was going to happen next. First, she tried to look down and make out what she was coming to, but it was too dark to see anything: then she looked at the sides of the well, and noticed that they were filled with cupboards and book-shelves: here and there she saw maps and pictures hung upon pegs. She took down a jar from one of the shelves as she passed; it was labelled "ORANGE MARMALADE," but to her great disappointment it was empty: she did not like to drop the jar for fear of killing somebody underneath, so managed to put it into one of the cupboards as she fell past it.

J'ajoute que le fait d'avoir été, durant plus de deux ans, le directeur de l'hôpital du camp et d'avoir, pendant toute la durée de mon internement, donné régulièrement des cours de français, d'histoire de la littérature française et d'histoire de la philosophie, m'a permis d'avoir des relations suivies et intimes avec presque tous mes compagnons de l'aventure. Ces relations et cette intimité m'ont permis de les connaître, un par un, de voir non seulement leur corps nu, lorsqu'ils se présentaient à la visite médicale, mais aussi leur âme à nu aux heures de confidences.

THE END

Fig. 5: After global thresholding at 150

Saturation adjustment enhances contrast between the text and background, making the content more distinguishable.

Lorem Ipsum

Alice was beginning to get very tired of sitting by her sister in the café, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice, "without pictures or conversations?"

So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a white rabbit with pink eyes ran close by her.

There was nothing so *very* remarkable in that; nor did Alice think it so *very* much out of the way to hear the Rabbit say to itself, "Oh dear! Oh dear! I shall be too late!" (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural;) but when the Rabbit actually *took a watch out of its waistcoat-pocket*, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and, burning with curiosity, she ran across the field after it, and was just in time to see it pop down a large rabbit-hole under the hedge.

In another moment down went Alice after it, never once considering how in the world she was to get out again.

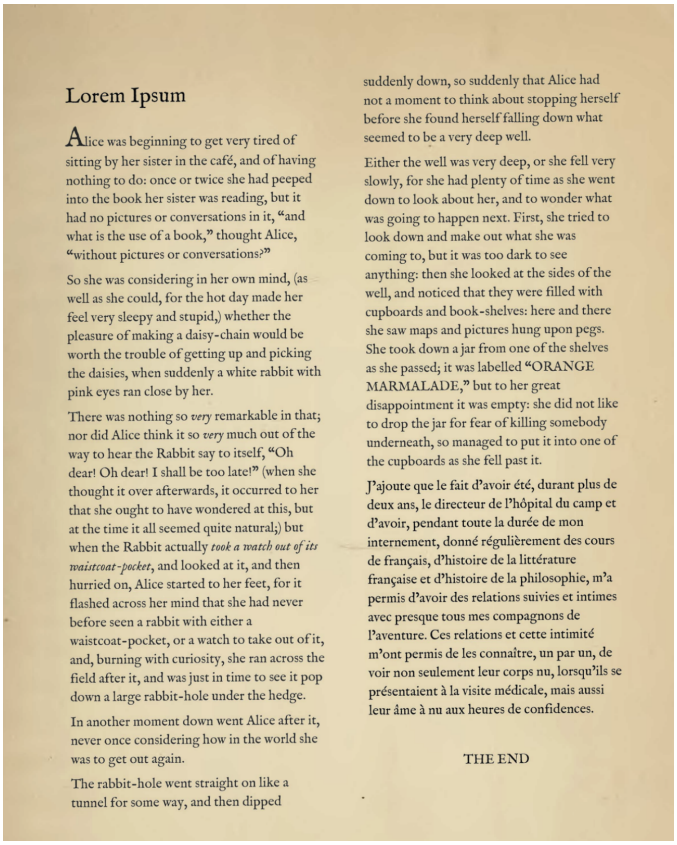
The rabbit-hole went straight on like a tunnel for some way, and then dipped suddenly down, so suddenly that Alice had not a moment to think about stopping herself before she found herself falling down what seemed to be a very deep well.

Either the well was very deep, or she fell very slowly, for she had plenty of time as she went down to look about her, and to wonder what was going to happen next. First, she tried to look down and make out what she was coming to, but it was too dark to see anything: then she looked at the sides of the well, and noticed that they were filled with cupboards and book-shelves: here and there she saw maps and pictures hung upon pegs. She took down a jar from one of the shelves as she passed; it was labelled "ORANGE MARMALADE," but to her great disappointment it was empty: she did not like to drop the jar for fear of killing somebody underneath, so managed to put it into one of the cupboards as she fell past it.

J'ajoute que le fait d'avoir été, durant plus de deux ans, le directeur de l'hôpital du camp et d'avoir, pendant toute la durée de mon internement, donné régulièrement des cours de français, d'histoire de la littérature française et d'histoire de la philosophie, m'a permis d'avoir des relations suivies et intimes avec presque tous mes compagnons de l'aventure. Ces relations et cette intimité m'ont permis de les connaître, un par un, de voir non seulement leur corps nu, lorsqu'ils se présentaient à la visite médicale, mais aussi leur âme à nu aux heures de confidences.

THE END

Fig. 6: Saturation Adjustment at 0.6



Lorem Ipsum

Alice was beginning to get very tired of sitting by her sister in the café, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice, "without pictures or conversations?"

So she was considering in her own mind, (as well as she could, for the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a white rabbit with pink eyes ran close by her.

There was nothing so *very* remarkable in that; nor did Alice think it so *very* much out of the way to hear the Rabbit say to itself, "Oh dear! Oh dear! I shall be too late!" (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural;) but when the Rabbit actually *took a watch out of its waistcoat-pocket*, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and, burning with curiosity, she ran across the field after it, and was just in time to see it pop down a large rabbit-hole under the hedge.

In another moment down went Alice after it, never once considering how in the world she was to get out again.

The rabbit-hole went straight on like a tunnel for some way, and then dipped suddenly down, so suddenly that Alice had not a moment to think about stopping herself before she found herself falling down what seemed to be a very deep well.

Either the well was very deep, or she fell very slowly, for she had plenty of time as she went down to look about her, and to wonder what was going to happen next. First, she tried to look down and make out what she was coming to, but it was too dark to see anything: then she looked at the sides of the well, and noticed that they were filled with cupboards and book-shelves: here and there she saw maps and pictures hung upon pegs. She took down a jar from one of the shelves as she passed; it was labelled "ORANGE MARMALADE," but to her great disappointment it was empty: she did not like to drop the jar for fear of killing somebody underneath, so managed to put it into one of the cupboards as she fell past it.

J'ajoute que le fait d'avoir été, durant plus de deux ans, le directeur de l'hôpital du camp et d'avoir, pendant toute la durée de mon internement, donné régulièrement des cours de français, d'histoire de la littérature française et d'histoire de la philosophie, m'a permis d'avoir des relations suivies et intimes avec presque tous mes compagnons de l'aventure. Ces relations et cette intimité m'ont permis de les connaître, un par un, de voir non seulement leur corps nu, lorsqu'ils se présentaient à la visite médicale, mais aussi leur âme à nu aux heures de confidences.

THE END

Fig. 7: Saturation adjustment at 2.5

Text Accuracy (Cosine Similarity): The accuracy of the extracted text is evaluated by comparing it with the original content using cosine similarity. This metric measures how closely the OCR-processed text aligns with the reference text. A higher similarity score signifies better text recognition and minimal deviation from the original text.

Model	Grayscale	GlobalThr	GlobalThr	GlobalThr	GlobalThr	GlobalThr	GlobalThr	Saturation	Saturation	Saturation	Saturation	Saturation	Saturation	Deskewed	AdaptiveThreshold
Grayscale	1	0.92562	0.93571	0.95091	0.93271	0.92114	0.9049	0.91801	0.90791	0.9092	0.93183	0.92329	0.92014	0.94148	0.88172
GlobalThreshold_80	0.92562	1	0.95949	0.9242	0.90188	0.89318	0.88504	0.8899	0.87723	0.88288	0.89892	0.89771	0.88404	0.92225	0.85245
GlobalThreshold_100	0.93571	0.95949	1	0.94309	0.93118	0.92468	0.91076	0.914	0.90421	0.91037	0.91848	0.91775	0.91075	0.93818	0.87532
GlobalThreshold_110	0.95091	0.9242	0.94309	1	0.92099	0.90012	0.88569	0.89367	0.8813	0.88076	0.90362	0.89956	0.90021	0.90922	0.86515
GlobalThreshold_120	0.93271	0.90188	0.93118	0.92099	1	0.95764	0.94456	0.95015	0.94604	0.93695	0.94881	0.94335	0.94839	0.94461	0.93175
GlobalThreshold_140	0.92114	0.89318	0.92468	0.90012	0.95764	1	0.98045	0.97651	0.98222	0.97407	0.97435	0.97265	0.97545	0.95963	0.94463
GlobalThreshold_150	0.9049	0.88504	0.91076	0.88569	0.94456	0.98045	1	0.97603	0.9806	0.96528	0.96848	0.97088	0.97243	0.94279	0.93605
SaturationAdjusted_1.2	0.91801	0.8899	0.914	0.89367	0.95015	0.97651	0.97603	1	0.98277	0.97168	0.97166	0.97122	0.96633	0.9525	0.92464
SaturationAdjusted_0.6	0.90791	0.87723	0.90421	0.8813	0.94604	0.98222	0.9806	0.98277	1	0.97771	0.97625	0.97284	0.97363	0.95142	0.93401
SaturationAdjusted_0.9	0.9092	0.88288	0.91037	0.88076	0.93695	0.97407	0.96528	0.97168	0.97771	1	0.96612	0.96111	0.95667	0.94967	0.91763
SaturationAdjusted_1.6	0.93183	0.89892	0.91848	0.90362	0.94881	0.97435	0.96848	0.97166	0.97625	0.96612	1	0.96866	0.97777	0.95556	0.92607
SaturationAdjusted_2	0.92329	0.89771	0.91775	0.89956	0.94335	0.97265	0.97088	0.97122	0.97284	0.96111	0.96866	1	0.96975	0.94765	0.92177
SaturationAdjusted_2.5	0.92014	0.88404	0.91075	0.90021	0.94839	0.97545	0.97243	0.96633	0.97363	0.95667	0.97777	0.96975	1	0.94394	0.92872
Deskewed	0.94148	0.92225	0.93818	0.90922	0.94461	0.95963	0.94279	0.9525	0.95142	0.94967	0.95556	0.94765	0.94394	1	0.91693
AdaptiveThreshold	0.88172	0.85245	0.87532	0.86515	0.93175	0.94463	0.93605	0.92464	0.93401	0.91763	0.92607	0.92177	0.92872	0.91693	1

Fig. 8: Cosine matrix result

Processing Time: The speed at which an image is processed plays a crucial role in determining the efficiency of a preprocessing technique. The time taken to execute each method is recorded, facilitating a comparative analysis. Faster processing time indicates a more optimized approach, making it suitable for large-scale applications where efficiency is a priority.

Memory Consumption: The computational efficiency of a preprocessing method is evaluated by monitoring its memory usage. Reducing memory consumption is essential for han-

dling large datasets efficiently without overloading system resources. A lower memory footprint ensures smooth processing while maintaining accuracy.

Model	Average Time (s)	Average Memory (MB)
Grayscale	2024.865767	0.013807954
GlobalThreshold_80	1676.491633	0.002740924
GlobalThreshold_100	1701.283533	0.001915953
GlobalThreshold_110	1733.68175	0.00275027
GlobalThreshold_120	1655.744083	0.00275027
GlobalThreshold_140	1670.999433	0.00275027
GlobalThreshold_150	1693.374417	0.002975761
SaturationAdjusted_1.2	2127.863867	0.001956601
SaturationAdjusted_0.6	2134.148967	0.00275027
SaturationAdjusted_0.9	2123.34895	0.002707142
SaturationAdjusted_1.6	2145.41245	0.002723567
SaturationAdjusted_2	2085.000333	0.002681414
SaturationAdjusted_2.5	2074.20855	0.00275027
Deskewed	1199.244817	2.110389561
AdaptiveThreshold	4515.6431	0.002624088

Fig. 9: Time tracking and memory usage ranking

By analyzing these performance metrics, an optimal preprocessing strategy can be identified, balancing accuracy, speed, and resource efficiency. A scoring mechanism is employed to rank the different preprocessing models, integrating cosine similarity, execution time, and memory efficiency to determine the most effective approach for enhancing OCR performance.

Rank	Model	Final Score
1	Grayscale	0.86417
2	GlobalThreshold_110	0.86002
3	GlobalThreshold_120	0.85609
4	GlobalThreshold_100	0.85465
5	GlobalThreshold_80	0.85117
6	GlobalThreshold_140	0.84929
7	GlobalThreshold_150	0.83967
8	SaturationAdjusted_1.6	0.82312
9	SaturationAdjusted_2	0.82287
10	SaturationAdjusted_2.5	0.82201
11	SaturationAdjusted_1.2	0.81745
12	SaturationAdjusted_0.9	0.81328
13	SaturationAdjusted_0.6	0.81191
14	Deskewed	0.69107
15	AdaptiveThreshold	0.64061

Fig. 10: Final ranking

VI. Conclusion

This paper presents an enhanced Optical Character Recognition (OCR) system that integrates advanced image preprocessing techniques with Tesseract OCR, cosine similarity analysis, and embedding-based text representation using OpenAI's text-embedding-ada-002 model. The goal of this research was to identify the most effective pre-processing techniques for improving OCR accuracy while maintaining computational efficiency in terms of time and memory usage. The results demonstrate that proper pre-processing significantly enhances text recognition, particularly in images with noise, distortions, or low contrast. Through extensive experimentation, we evaluated multiple pre-processing techniques, including grayscale conversion, adaptive thresholding, global thresholding, saturation adjustment, and deskewing. Each of these techniques was assessed based on its impact on OCR accuracy, processing time, and memory consumption. Our findings indicate that a combination of grayscale conversion and adaptive thresholding yields the best balance between accuracy and computational efficiency. Grayscale conversion simplifies the image by reducing color variations, thereby improving character differentiation. Adaptive thresholding further enhances the contrast between text and background, making the extracted text more distinguishable to the OCR engine.

In addition to traditional OCR processing, this study incorporated text similarity analysis using the Cosine Similarity Matrix to improve recognition reliability. Furthermore, we explored the benefits of embedding generation using OpenAI's text-embedding-ada-002 model. This model generates numerical representations of text, capturing semantic meaning and contextual relationships. By leveraging text embeddings, the OCR system enhances its capability to perform text classification, clustering, and search tasks. Our results demonstrate that embedding-based text analysis significantly improves document organization and retrieval accuracy, making it a valuable addition to modern OCR pipelines.

One of the unique contributions of this research is the evaluation of computational efficiency, which is often overlooked in OCR studies. We recorded the time and memory usage for each pre-processing technique and observed that while some methods improve accuracy, they also introduce additional computational overhead. The balance between accuracy and efficiency is crucial for real-world applications where large-scale document processing is required. Our findings suggest that grayscale conversion and adaptive thresholding provide the optimal trade-off, ensuring high accuracy without excessive resource consumption. Overall, this study reinforces the importance of pre-processing techniques in enhancing OCR performance. By integrating Tesseract OCR with cosine similarity and embedding-based text representation, we have demonstrated a novel approach that improves text recognition accuracy while optimizing computational resources. The insights from this research can be applied to various domains, including automated data entry, historical document preservation, and intelligent document retrieval systems.

For future work, we propose further exploration of deep learning-based image enhancement techniques, such as Generative Adversarial Networks (GANs) for document restoration. Additionally, integrating multimodal learning approaches that combine textual and visual information could further refine OCR accuracy. The continuous advancement of AI-driven text recognition methods presents exciting opportunities for future research and development in this field.

VII. Limitations and Future Work

While this research has demonstrated significant improvements in OCR accuracy and efficiency, there remain several opportunities for future exploration. One promising direction is the integration of deep learning-based image enhancement techniques, such as Generative Adversarial Networks (GANs) or Convolutional Neural Networks (CNNs), to further improve image quality before text extraction. These advanced models can effectively remove noise, enhance text contrast, and reconstruct degraded portions of images, leading to improved OCR results. Additionally, future work can focus on the development of adaptive pre-processing pipelines that dynamically select the best enhancement techniques based on image characteristics. Instead of applying a fixed set of preprocessing steps, machine learning models can be trained to determine the optimal transformation techniques for each input image, maximizing accuracy while minimizing unnecessary computational costs. Another potential research avenue is the exploration of multi-modal learning, which combines OCR with additional contextual information, such as document structure, handwriting recognition, and natural language processing (NLP). Integrating OCR with NLP-based correction models can further improve text recognition by refining misclassified words based on semantic understanding and contextual analysis. Moreover, real-world OCR applications often involve multi-language documents, requiring robust language identification and character recognition capabilities. Future research can investigate the effectiveness of transformer-based models, such as multilingual BERT, in improving OCR for diverse languages and scripts. Finally, extending this study to large-scale datasets and real-time applications would provide valuable insights into the practical deployment of OCR systems in industries such as finance, healthcare, and legal documentation. Evaluating the scalability and adaptability of the proposed approach in high-volume text processing scenarios will be crucial for enhancing OCR adoption in enterprise applications.

VIII. Acknowledgement

We sincerely appreciate the invaluable guidance and insightful feedback provided by Professor Dr. Damir Dobric throughout this project. His support has been instrumental in shaping our research. Additionally, we extend our gratitude to the tutors for their assistance in clarifying doubts and offering their valuable insights.

References

[1] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. Ninth Int. Conf. Document Anal. Recognit. (ICDAR)*, Curitiba, Brazil, 2007, pp. 629-633.

[2] W. Bieniecki and S. Grabowski, "Image preprocessing for improving OCR accuracy," in *Proc. Int. Conf. Perspective Technol. Methods MEMS Design*, MEMSTECH 2007, June 2007.

[3] W. Moon, S. Nengroo, T. Kim, J. Lee, S. Son, and D. Har, "Enhanced optical character recognition by optical sensor combined with BERT and cosine similarity scoring (Student abstract)," Korea Advanced Institute of Science and Technology, Daejeon, South Korea.

[4] OpenAI, "Introducing text-embedding-ada-002," 2025.

[5] N. Muda, N. K. Ismail, S. A. Abu Bakar, and J. M. Zain, "Optical character recognition by using template matching (alphabet)."

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, May 2015.

[7] S. A. Khan, A. H. Khan, and M. I. A. Anwar, "Adaptive thresholding methods for documents image binarization," *ResearchGate*, 2011.

[8] I. Singh, M. Colom, and K. Bontcheva, "A comparative analysis of OCR models on diverse datasets: Insights from Memes and Hiertext dataset," University of Sheffield, UK; Centre Borelli, ENS Paris-Saclay.

[9] N. Arica and F. T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 31, no. 2, pp. 216–233, May 2001.

[10] Y. Li and T. Yang, "Word embedding for understanding natural language: A survey," pp. 83-104, 2018.